Termination Tools in Automated Reasoning

dissertation

by

Sarah Winkler

submitted to the Faculty of Mathematics, Computer Science and Physics of the University of Innsbruck

in partial fulfillment of the requirements
for the degree of Doktor der technischen Wissenschaften

advisor: Univ.-Prof. Dr. Aart Middeldorp

Innsbruck, 17 April 2013

dissertation

Termination Tools in Automated Reasoning

Sarah Winkler (0019379)
sarah.winkler@uibk.ac.at

17 April 2013

advisor: Univ.-Prof. Dr. Aart Middeldorp

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt durch meine eigenhändige Unterschrift, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe. Alle Stellen, die wörtlich oder inhaltlich den angegebenen Quellen entnommen wurden, sind als solche kenntlich gemacht.
Die vorliegende Arbeit wurde bisher in gleicher oder ähnlicher Form noch nicht als Magister-/Master-/Diplomarbeit/Dissertation eingereicht.

Datum	Unterschrift

Abstract

Automated reasoning gained considerable interest over the last two decades and has by now a variety of applications in formal methods of computer science, such as formal specification of software and hardware systems or verification of cryptographic protocols. Automated reasoning systems are also used in mathematical research, where mathematical software provides proof assistance using automated reasoning techniques. Computing systems in automated reasoning are often based on decision calculi which require a well-founded term ordering as input. Such a term ordering is always critical for success, but an appropriate choice is hard to determine in advance. For standard Knuth-Bendix completion, different approaches to automate this choice were proposed by exploiting recent progress in research on automatic termination proving techniques on one hand, and satisfiability checking on the other hand. Since modern termination tools employ far more sophisticated techniques than classical term orderings, this approach contributes to more flexible and powerful automated reasoning and theorem proving technology. Automation techniques relying on satisfiability checkers allow for efficient implementations. This thesis explores the use of these approaches in different deduction calculi of equational reasoning. Besides standard Knuth-Bendix completion also ordered completion, normalized completion and rewriting induction are considered.

Acknowledgments

First of all, I am grateful to Aart Middeldorp for introducing me into the fascinating world of computational logic. His inspiring teaching led me to work on term rewriting, and his knowledge, guidance, continuous support, and careful checking were invaluable when writing this thesis. I am also greatly obliged to Nao Hirokawa for giving me a glance into the magic of functional programming, and to Georg Moser for getting me acquainted with the mind-boggling world of set theory. Special thanks go to Chris, Harald, and Martin K. for producing all this neat TTT2 code. It was a real pleasure to work in the Computational Logic group. All of you CL people not only provided an inspiring working environment and a source of highly useful expertise, but also made many fun hiking events and lab parties happen.

Concerning funding, I am indebted to the Austrian Academy of Science for providing me with a docfForte grant that made my PhD studies possible, and to the University of Innsbruck for earlier financial support.

My outstanding appreciation goes to my parents—for giving me the opportunity to study, but even more for their continuous encouragement, care, and patience with a daughter spending her studies on a lot of strange symbols.

Last but not least I am most happy to have shared this time of my studies with such great people. Eva, Robert, and Wolfi—without your company everywhere between the Malaysian jungle, the Cuban sea and the coffee machine, these years are unthinkable. Eva, thanks for sharing an apartment and a climbing rope as well as so many other things, and for always lending an ear—*you* truly are the best. Extra thanks go to Andi, Matthias, and Zugi for all those exciting ski tours, bike rides, and climbs. Simon and Julia, the relaxing evenings with you were always a wonderful distraction. But certainly, without Wolfi's cheerful, supportive and simply terrific companionship in our joint adventures nothing would have been this way.

Contents

Chapter 1

Introduction

Reasoning refers to the ability to draw conclusions from certain facts. *Automated reasoning* aims to design computing systems that automate this process. This requires a precise algorithmic description of some formal deduction calculus such that reasoning can be efficiently implemented.

In automated theorem proving (ATP), one of the branches of automated reasoning which receives a lot of interest, conjectures are checked against sets of axioms and hypotheses. Automated reasoning gained considerable attention over the last two decades, in particular, developments in the area of automated theorem proving have resulted in a variety of application areas in mathematics and different branches of computer science. These include formal specification and verification of software and hardware systems, planning and scheduling problems requiring deduction involving constraints and goals, verification of cryptographic protocols, and also semantic web applications. Research as well as teaching in mathematics benefited from automated theorem provers and proof assistants, which have been used to provide evidence for conjectures in a variety of mathematical contexts. ATP techniques are used to enhance the capabilities of mathematical software like *Mathematica*.

On a theoretical level, the reasoning methods underlying theorem provers rely on *deduction calculi*. A deduction calculus is usually specified by inference rules which describe how to draw conclusions from a set of facts.

Knuth-Bendix completion [57] constitutes a classical example of a deduction calculus for equational theories. It attempts to derive a program from a given set of equations, thus extracting a decision procedure for the associated equational theory. More precisely, it aims to transform a set of equations into a term rewrite system which is terminating, confluent and has the same equational theory. If such a rewrite system can be found for an equational theory then it can be used to effectively decide the theory.

Example 1.1. For instance, Knuth-Bendix completion can transform the following three equations axiomatizing group theory:

$$\mathsf{e} \cdot x \approx x \qquad \mathsf{i}(x) \cdot x \approx \mathsf{e} \qquad x \cdot (y \cdot z) \approx (x \cdot y) \cdot z$$

into a terminating and confluent rewrite system:

$$\begin{array}{lll} \mathsf{e} \cdot x \to x & \mathsf{i}(x) \cdot x \to \mathsf{e} & x \cdot (y \cdot z) \to (x \cdot y) \cdot z \\ x \cdot \mathsf{e} \to x & x \cdot \mathsf{i}(x) \to \mathsf{e} & (x \cdot y) \cdot \mathsf{i}(y) \to x \\ \mathsf{i}(\mathsf{e}) \to \mathsf{e} & \mathsf{i}(\mathsf{i}(x)) \to x & (x \cdot \mathsf{i}(y)) \cdot y \to x \\ \mathsf{i}(x \cdot y) \to \mathsf{i}(y) \cdot \mathsf{i}(x) & & \end{array}$$

Although on an abstract level, this rewrite system constitutes a program that can decide validity of any conjecture concerning pure group theory.

Further common inference systems for equational reasoning include ordered completion [13] and different approaches for completion modulo theories, with normalized completion [75] being the most recent method. Paramodulation is widely employed in theorem provers [14,15]. Congruence closure algorithms [17] are used to obtain decision procedures for the simpler case of equational theories without variables. Calculi such as inductionless induction [58] and rewriting induction [86] are applied in the area of inductive theorem proving.

All the above-mentioned calculi are described by an inference system which requires a well-founded term ordering as input. This *reduction order* guides the search process and guarantees termination of the resulting rewrite system. Virtually all ATP tools accomplish this requirement by implementing well-known reduction orders such as LPO, KBO, or their AC-compatible counterparts. But these orders have inherent limitations.

Example 1.2. Consider the following set of equations CGE_3 describing the algebraic structure of a group with three commuting group endomorphisms:

$$\begin{array}{rlll} \mathsf{e} \cdot x \approx x & \mathsf{f}_1(x \cdot y) \approx \mathsf{f}_1(x) \cdot \mathsf{f}_1(y) & \mathsf{f}_1(x) \cdot \mathsf{f}_2(y) \approx \mathsf{f}_2(y) \cdot \mathsf{f}_1(x) \\ \mathsf{i}(x) \cdot x \approx \mathsf{e} & \mathsf{f}_2(x \cdot y) \approx \mathsf{f}_2(x) \cdot \mathsf{f}_2(y) & \mathsf{f}_1(x) \cdot \mathsf{f}_3(y) \approx \mathsf{f}_3(y) \cdot \mathsf{f}_1(x) \\ x \cdot (y \cdot z) \approx (x \cdot y) \cdot z & \mathsf{f}_3(x \cdot y) \approx \mathsf{f}_3(x) \cdot \mathsf{f}_3(y) & \mathsf{f}_2(x) \cdot \mathsf{f}_3(y) \approx \mathsf{f}_3(y) \cdot \mathsf{f}_2(x) \end{array}$$

A decision procedure for this theory has applications in SMT solvers for the theory of uninterpreted function symbols [99]. However, no Knuth-Bendix completion tool relying on LPO and KBO can succeed in this case because termination of the equations in the last column cannot be established.

For all of the above-mentioned deduction calculi the restriction to classical reduction orders is a severe limitation with respect to computational power. Moreover, the ordering typically needs to be supplied by the user. It is always critical for success, but an appropriate choice is hard to determine in advance. It is hence desirable to have *fully automatic* procedures for equational reasoning, where no reduction order is required as input.

The aim of this thesis is thus to develop powerful and fully automatic methods for equational theorem.

For standard Knuth-Bendix completion, this research direction was already tackled in different ways: (1) Kondo and Kurihara proposed multi-completion, which employs multiple reduction orders in parallel [62], (2) Wehrman, Stump, and Westbrook showed that the use of automatic termination tools can replace a fixed ordering, thereby considerably extending the class of applicable reduction orders [107], and (3) Klein and Hirokawa proposed to formulate completion as a constraint satisfaction problem and presented a competitive implementation via an encoding as a maximal satisfiability problem [56].

Multi-completion with termination tools combines the ideas of (1) and (2) [90]. The potential of this approach was demonstrated by the prototype implementation mkbTT, which was the first tool to automatically complete CGE_3. This PhD thesis focuses on thoroughly exploring approaches (1)–(3) in the settings of standard, ordered and normalized completion. Our aims can thus be summarized by the following central research questions:

- Can the tool mkbTT benefit from well-established optimizations in ATP such as term indexing, selection strategies, and critical pair criteria?
- How can the approach of multi-completion with termination tools be carried over to ordered completion and normalized completion?
- How can maximal completion procedures for ordered completion and normalized completion be designed?

We will furthermore comment on maximal completion-like procedures in rewriting induction.

These aims are reflected in this thesis as follows.

In Chapter 2 we collect preliminaries required in the remainder of the thesis. Chapter 3 defines an abstract notion of an inference system and derives some general results. These are then used to describe classical Knuth-Bendix completion. We include a simpler presentation for finite runs. Chapter 4 gives an account of multi-completion and completion of termination tools before describing their combination in multi-completion with termination tools. We conclude this chapter by proposing critical pair criteria and isomorphisms as optimizations. Ordered completion is investigated from a similar viewpoint in Chapter 5. We first recall the well-known calculus of ordered completion, including refutational theorem proving, critical pair criteria and a simplification for finite runs. Then the applications and limitations of termination tools in ordered completion are investigated. Finally, a method embedding the use of termination tools in a multi-completion setting is presented. We turn to normalized completion in Chapter 6. Our presentation of normalized completion slightly differs from the original proposal as a central ingredient was found to require corrections. Again a simplification for finite runs and the use of critical pair criteria are discussed. Moreover, normalized completion with termination tools as well as its use in a multi-completion setting are described. In Chapter 7 we outline completion procedures based on the constrained equality framework, which constitutes a generalization of maximal completion. This approach is then carried over to ordered completion and normalized completion. We also recall a rewriting induction procedure based on the constrained equality framework. Chapter 8 describes our tool mkbTT performing standard, ordered, and normalized multi-completion with termination tools. We also outline a prototype implementation of a tool based on the constrained equality framework. The chapter is concluded by an evaluation of our tools for all completion variants discussed. For mkbTT, we report on experiments with critical pair criteria, term indexing techniques, selection strategies, and isomorphisms. The usefulness of different termination methods is evaluated for both tools. Finally, Chapter 9 draws conclusions.

Chapter 2

Preliminaries

In this section basic concepts, terminology and notation related to rewriting are introduced. For further details we refer to [4, 101].

2.1 Term Rewriting

We consider a finite set of function symbols $\mathcal{F}$ with fixed arities, also called a *signature*, and an infinite disjoint set of *variables* $\mathcal{V}$. Function symbols with arity 0 are called *constants*. We write $\mathcal{T}(\mathcal{F}, \mathcal{V})$ for the set of *terms* built over $\mathcal{F}$ and $\mathcal{V}$. By $|t|_x$ we denote how often a variable x occurs in a term t. Terms without variables are called *ground*, and the set of ground terms built over $\mathcal{F}$ is denoted by $\mathcal{T}(\mathcal{F})$. The *root symbol* of a term t is defined as $\mathrm{root}(t) = f$ if $t = f(t_1, \ldots, t_n)$ and $\mathrm{root}(t) = t$ if $t \in \mathcal{V}$. The set of variables occurring in a term t is denoted by $\mathcal{V}\mathrm{ar}(t)$. For a term t its set of *positions* $\mathcal{P}\mathrm{os}(t)$ is inductively defined such that $\mathcal{P}\mathrm{os}(x) = \{\epsilon\}$ if $x \in \mathcal{V}$ and $\mathcal{P}\mathrm{os}(t) = \{\epsilon\} \cup \{i.p \mid p \in \mathcal{P}\mathrm{os}(t_i) \text{ and } 1 \leqslant i \leqslant n\}$ if $t = f(t_1, \ldots, t_n)$. The set $\mathcal{P}\mathrm{os}(t)$ is the union of the set of *variable positions* $\mathcal{P}\mathrm{os}_{\mathcal{V}}(t)$ and the set of *function symbol positions* $\mathcal{P}\mathrm{os}_{\mathcal{F}}(t)$. We define the *subterm* of t at position p, denoted by $t|_p$, as t if $p = \epsilon$ and $t_i|_q$ if $t = f(t_1, \ldots, t_n)$ and $p = i.q$ for some $1 \leqslant i \leqslant n$. We write $t \trianglerighteq s$ if s is a subterm of t at some position $p \in \mathcal{P}\mathrm{os}(t)$. If $t \trianglerighteq s$ and $s \neq t$ we write $t \triangleright s$ and call s a *proper* subterm of t. The result of replacing the subterm of t at position p by some term s is denoted $t[s]_p$. A *context* C is a term with a single occurrence of a designated constant $\Box$ called *hole*. If $\Box$ in C is replaced by some term t then the result is denoted by $C[t]$.

A *substitution* σ is a mapping from variables to terms such that $\sigma(x) \neq x$ for only finitely many variables x. The *domain* $\mathcal{D}\mathrm{om}(\sigma)$ is the set $\{x \in \mathcal{V} \mid \sigma(x) \neq x\}$. The natural extension of σ from terms to terms is ambiguously also denoted by σ and defined as $t\sigma = \sigma(x)$ if $x \in \mathcal{V}$, and $f(t_1, \ldots, t_n)\sigma = f(t_1\sigma, \ldots, t_n\sigma)$. A substitution σ is *ground* if $\sigma(x)$ is ground for all $x \in \mathcal{D}\mathrm{om}(\sigma)$, and *grounding* for a term t if $t\sigma$ is ground. A term s is called an *instance* of a term t if there exists a substitution σ such that $s = t\sigma$. Then t is also said to *subsume* s, which is expressed by writing $t \leqslant s$. We also define the *encompassment* relation $\trianglerighteq$ on terms, where t encompasses s, denoted by $t \trianglerighteq s$, if $t \trianglerighteq s\sigma$ for some substitution σ. The *proper* encompassment relation $\triangleright$ is defined as the strict part of $\trianglerighteq$, i.e., $t \triangleright s$ holds if t encompasses s but not vice versa. We denote by $\doteq$ the equivalence relation associated with $\trianglerighteq$, and call s a *variant* of t if $s \doteq t$ holds.

We consider equations $s \approx t$ between terms, and sets of equations $\mathcal{E}$ called *equational systems* (ESs). We write $s \simeq t$ to denote $s \approx t$ or $t \approx s$. The

equational theory induced by $\mathcal{E}$ is denoted by $\leftrightarrow^*_{\mathcal{E}}$. A *rewrite rule* $\ell \to r$ is a directed equation such that $\ell \notin \mathcal{V}$ and $\mathcal{V}\mathrm{ar}(r) \subseteq \mathcal{V}\mathrm{ar}(\ell)$, and a set of rewrite rules $\mathcal{R}$ is called a *term rewrite system* (TRS). For an ES $\mathcal{E}$ and a TRS $\mathcal{R}$ we sometimes write $\mathcal{E} \cup \mathcal{R}$ to denote the ES consisting of the equations in $\mathcal{E}$ together with all equations $\ell \approx r$ such that $\ell \to r \in \mathcal{R}$. Given a TRS $\mathcal{R}$, the set $\mathcal{F}_{\mathcal{D}} = \{\mathrm{root}(\ell) \mid \ell \to r \in \mathcal{R}\}$ is the set of *defined function symbols*. The set $\mathcal{F}_{\mathcal{C}} = \mathcal{F} \setminus \mathcal{F}_{\mathcal{D}}$ is the set of *constructor symbols*.

A TRS $\mathcal{R}$ induces a *rewrite relation* $\to_{\mathcal{R}}$ on terms, where $t \to_{\mathcal{R}} s$ holds if $t = C[\ell\sigma]$ and $s = C[r\sigma]$ for some context C, substitution σ and rewrite rule $\ell \to r$ in $\mathcal{R}$. The transitive (and reflexive) closure is denoted by $\to^+_{\mathcal{R}}$ ($\to^*_{\mathcal{R}}$). The *conversion* of $\mathcal{R}$ is the smallest equivalence relation containing $\to_{\mathcal{R}}$, denoted by $\leftrightarrow^*_{\mathcal{R}}$. A term t is called a *normal form* with respect to $\mathcal{R}$ if there is no term u such that $t \to_{\mathcal{R}} u$. If $s \to^*_{\mathcal{R}} t$ and t is a normal form we also write $s \to^!_{\mathcal{R}} t$. In the sequel subscripts will be omitted if the TRS is clear from the context. A TRS $\mathcal{R}$ is *terminating* if the relation $\to_{\mathcal{R}}$ is well-founded. We write $s \downarrow_{\mathcal{R}} t$ and call s and t *joinable* if there exists a term u such that $s \to^*_{\mathcal{R}} u \,{}^*_{\mathcal{R}}\!\leftarrow t$. A TRS $\mathcal{R}$ is *locally* (*ground*) *confluent* if for every (ground) peak $s \,{}_{\mathcal{R}}\!\leftarrow u \to_{\mathcal{R}} t$ we have $s \downarrow_{\mathcal{R}} t$, and (*ground*) *confluent* if $s \,{}^*_{\mathcal{R}}\!\leftarrow u \to^*_{\mathcal{R}} t$ implies $s \downarrow_{\mathcal{R}} t$. If $\mathcal{R}$ is both terminating and (ground) confluent it is called (*ground*) *convergent*. A TRS $\mathcal{R}$ is *convergent for* a set of equations $\mathcal{E}$ if $\mathcal{R}$ is convergent and $\leftrightarrow^*_{\mathcal{R}} = \leftrightarrow^*_{\mathcal{E}}$. Furthermore, a rewrite system $\mathcal{R}$ is *reduced* if for every rewrite rule $\ell \to r$ in $\mathcal{R}$ the term r is in normal form with respect to $\mathcal{R}$ and ℓ is in normal form with respect to $\mathcal{R} \setminus \{\ell \to r\}$. If a TRS is both convergent and reduced it is also called *canonical*. An *overlap* is a triple $\langle \ell_1 \to r_1, p, \ell_2 \to r_2 \rangle_\sigma$ where $\ell_1 \to r_1$ and $\ell_2 \to r_2$ are rewrite rules without common variables such that $p \in \mathcal{P}\mathrm{os}_{\mathcal{F}}(\ell_2)$, $\ell_2|_p$ and ℓ_1 are unifiable with most general unifier σ, and if $\ell_1 \to r_1$ and $\ell_2 \to r_2$ are variants then $p \neq \epsilon$. The term $\ell_2\sigma = \ell_2\sigma[\ell_1\sigma]_p$ can be rewritten in two different ways, resulting in the terms $\ell_2\sigma[r_1\sigma]_p$ and $r_2\sigma$. The equation $\ell_2\sigma[r_1\sigma]_p \approx r_2\sigma$ is called a *critical pair* and often written as $\ell_2\sigma[r_1\sigma]_p \leftarrow \rtimes \to r_2\sigma$. We write $s \leftarrow \times \to t$ to denote $s \leftarrow \rtimes \to t$ or $t \leftarrow \rtimes \to s$, for any terms s and t. The set of critical pairs among rules in $\mathcal{R}$ is denoted by $\mathrm{CP}(\mathcal{R})$. We sometimes write $\mathrm{CP}(\mathcal{R}_1, \mathcal{R}_2)$ to denote the critical pairs stemming from overlaps such that $\ell_1 \to r_1 \in \mathcal{R}_1$ and $\ell_2 \to r_2 \in \mathcal{R}_2$ A peak $s \,{}_{r_1 \leftarrow \ell_1}\!\overset{p}{\leftarrow} \cdot \to^q_{\ell_2 \to r_2} t$ is sometimes called a *non-overlap* if it is not an instance of an overlap. Formally, this is the case if either $p \parallel q$, or $q = pq'$ and $q' \notin \mathcal{P}\mathrm{os}_{\mathcal{F}}(\ell_1)$, or $p = qp'$ and $p' \notin \mathcal{P}\mathrm{os}_{\mathcal{F}}(\ell_2)$ hold.

2.2 Reduction Orders

Before discussing reduction orders as a means to show termination of TRSs we recall the lexicographic and multiset extensions of arbitrary orders.

Definition 2.1. Let $(>_1, \sim_1), \ldots, (>_n, \sim_n)$ be a sequence of pairs of a proper order $>_i$ and an equivalence $\sim_i$ on some set A_i, for all $1 \leqslant i \leqslant n$. Their *lexicographic combinations* $(>_{\mathsf{lex}}, \sim_{\mathsf{lex}})$ on $A_1 \times \cdots \times A_n$ are defined as follows:

– $(a_1, \ldots, a_n) \sim_{\mathsf{lex}} (b_1, \ldots, b_n)$ if $a_i \sim_i b_i$ for all $1 \leqslant i \leqslant n$, and

– $(a_1, \ldots, a_n) >_{\mathsf{lex}} (b_1, \ldots, b_n)$ if $n = 1$ and $a_1 >_1 b_1$, or $n > 1$ and (i) $(a_1, \ldots, a_{n-1}) >_{\mathsf{lex}} (b_1, \ldots, b_{n-1})$ or (ii) $(a_1, \ldots, a_{n-1}) \sim_{\mathsf{lex}} (b_1, \ldots, b_{n-1})$ and $a_n >_n b_n$.

Definition 2.2. Let $(>, \sim)$ be a proper order and an equivalence on a set A. The *multiset extensions* $>_{\mathsf{mul}}$ and $\sim_{\mathsf{mul}}$ are defined on multisets over A as follows:

– $M \sim_{\mathsf{mul}} N$ if $M = \{x_1, \ldots, x_n\}$, $N = \{y_1, \ldots, y_n\}$, and $x_i \sim y_i$ for all $1 \leqslant i \leqslant n$.

– $M >_{\mathsf{mul}} N$ if there are multisets $X \subseteq M$ and $Y \subseteq N$ such that $X \neq \varnothing$, $M \setminus X \sim_{\mathsf{mul}} N \setminus Y$ and for every $y \in Y$ there is some $x \in X$ with $x > y$.

A pair $(>, \sim)$ of a proper order and an equivalence is an *order pair* if *compatibility* $\sim \cdot > \cdot \sim \subseteq >$ holds. It is well known that for all order pairs $(>_1, \sim_1) \ldots, (>_n, \sim_n)$ their lexicographic combination $(>_{\mathsf{lex}}, \sim_{\mathsf{lex}})$ is an order pair as well, and for any order pair $(>, \sim)$ also $(>_{\mathsf{mul}}, \sim_{\mathsf{mul}})$ is an order pair. If in addition $>_1, \ldots, >_n$ and $>$ are well-founded then $>^{\mathsf{lex}}$ and $>^{\mathsf{mul}}$ are well-founded [34]. For the sake of brevity, the lexicographic combination of proper orders $>_1, \ldots, >_n$ refers to the lexicographic combination of $(>_1, =), \ldots, (>_n, =)$. Similarly, the multiset extension of a proper order $>$ refers to the multiset extension of $(>, =)$.

Reduction orders constitute a basic means to establish termination of a TRS. In general, a *reduction order* $>$ is a well-founded order on terms that is closed under contexts and substitutions, i.e., $s > t$ implies $C[s\sigma] > C[t\sigma]$ for all contexts C and substitutions σ. In the sequel, we call a relation R on terms *compatible* with a TRS $\mathcal{R}$ if $\ell \mathrel{R} r$ holds for all rules $\ell \to r$ in $\mathcal{R}$. The interest in reduction orders is due to the following result.

Theorem 2.3. *A TRS terminates if and only if it is compatible with a reduction order.*

A reduction order $>$ is a *simplification order* if it contains the proper subterm relation, so $\rhd \subseteq >$. A TRS is *simply terminating* if it compatible with a simplification order. Furthermore, a reduction order will be called *ground-total* if its restriction to $\mathcal{T}(\mathcal{F})$ is total. It is known that any ground-total reduction order is a simplification order [39].

One way to define reduction orders is via well-founded algebras. A *well-founded $\mathcal{F}$-algebra* $(\mathcal{A}, >)$ consists of a non-empty carrier A, a well-founded relation $>$ on A, and an interpretation function $f_{\mathcal{A}}$ for every $f \in \mathcal{F}$. By $[\alpha]_{\mathcal{A}}(\cdot)$ we denote the usual evaluation function of $\mathcal{A}$ according to an assignment α. An algebra $(\mathcal{A}, >)$ is called monotone if every $f_{\mathcal{A}}$ is monotone with respect to $>$. Any algebra $(\mathcal{A}, >)$ induces an order on terms: we set $s >_{\mathcal{A}} t$ if for any assignment α the condition $[\alpha]_{\mathcal{A}}(s) > [\alpha]_{\mathcal{A}}(t)$ holds. We say that a TRS $\mathcal{R}$ is compatible with an algebra $\mathcal{A}$ if $\ell \to r \in \mathcal{R}$ implies $\ell >_{\mathcal{A}} r$. It is well-known that the relation $>_{\mathcal{A}}$ constitutes a reduction order, so a TRS that is compatible with a monotone algebra is terminating.

A TRS $\mathcal{R}$ is *totally terminating* if it is compatible with a well-founded monotone algebra $(\mathcal{A}, >)$ such that $>$ is total [39]. Note that in this case the relation $>_{\mathcal{A}}$ is ground-total.

We will now recall the lexicographic path order (LPO) [50] and the Knuth-Bendix order (KBO) [57], two of the most widely used reduction orders in termination analysis and theorem proving.

Lexicographic Path Order

LPO was originally introduced by Kamin and Lévy [50]. Dershowitz proposed a more powerful variant which allows the use of quasi-orders as precedence [31]. Weiermann showed that the derivation length of a TRS compatible with LPO is bound by a multiply recursive function [108]. With their proposal to encode LPO as a satisfaction problem using binary decision diagrams (BDDs) [21], Kondo and Kurihara [63] paved the way towards simple yet highly efficient implementations of reduction orders. Codish *et al.* [26] proposed a more efficient encoding of precedences, and replaced BDDs by modern SAT solvers.

We will now recall the standard definition of LPO.

Definition 2.4 ([31,50])**.** Let $\succ$ be a precedence. The *lexicographic path order* $\succ_{\mathrm{lpo}}$ is inductively defined as follows: $s \succ_{\mathrm{lpo}} t$ if $s = f(s_1, \dots, s_n)$ and one of the following alternatives holds:

(1) $s_i \succeq_{\mathrm{lpo}} t$ for some $1 \leqslant i \leqslant n$, or

(2) $t = g(t_1, \dots, t_m)$, $f \succ g$, and $s \succ_{\mathrm{lpo}} t_i$ for all $1 \leqslant i \leqslant m$, or

(3) $t = f(t_1, \dots, t_n)$ and there is some k with $1 \leqslant k \leqslant n$ such that $s_1 = t_1, \dots, s_{k-1} = t_{k-1}$, $s_k \succ_{\mathrm{lpo}} t_k$, and $s \succ_{\mathrm{lpo}} t_j$ for all $k < j \leqslant n$.

Here $\succeq_{\mathrm{lpo}}$ refers to the reflexive closure of $\succ_{\mathrm{lpo}}$.

Theorem 2.5 ([50])**.** *The relation $\succ_{\mathrm{lpo}}$ is a simplification order.* □

A TRS $\mathcal{R}$ is compatible with LPO if there exists some precedence $\succ$ such that $\mathcal{R}$ is compatible with $\succ_{\mathrm{lpo}}$.

Theorem 2.6 ([39])**.** *A TRS is totally terminating if it is compatible with LPO.* □

Example 2.7. Let the TRS $\mathcal{R}$ consist of the following two rules:

$$\mathsf{f}(x) \to \mathsf{g}(\mathsf{g}(x)) \tag{1}$$
$$\mathsf{g}(\mathsf{g}(\mathsf{g}(\mathsf{g}(x)))) \to \mathsf{g}(x) \tag{2}$$

Termination of $\mathcal{R}$ can be proven by LPO using precedence $\mathsf{f} \succ \mathsf{g}$: Rule (1) is oriented by case (2) in Definition 2.4, and rule (2) is oriented because the right-hand side is a subterm of the left-hand side and LPO is a simplification order.

Example 2.8 ([11])**.** The following TRS $\mathcal{R}$:

$$\mathsf{c} \to \mathsf{a} \tag{1}$$
$$\mathsf{g}(x) \to x \tag{2}$$
$$\mathsf{f}(\mathsf{b}, x) \to x \tag{3}$$
$$\mathsf{f}(\mathsf{g}(x), y) \to \mathsf{f}(x, \mathsf{g}(y)) \tag{4}$$
$$\mathsf{f}(x, \mathsf{b}) \to \mathsf{c} \tag{5}$$

can be shown terminating by LPO with precedence $\mathsf{f} \succ \mathsf{c} \succ \mathsf{g} \succ \mathsf{b} \succ \mathsf{a}$. Rule (1) is oriented by case (2) in Definition 2.4. Rules (2) and (3) are oriented by case (1). Rule (4) can be handled by case (3) as $\mathsf{g}(x) \succ_{\mathrm{lpo}} x$ constitutes a strict decrease in the first argument, and $\mathsf{f}(\mathsf{g}(x), y) \succ_{\mathrm{lpo}} \mathsf{g}(y)$. Finally, also rule (5) is oriented by case (2).

Knuth-Bendix Order

KBO was invented by Knuth and Bendix in connection with the first proposal for a completion procedure [57]. Several extensions were proposed to enhance the semantic component beyond plain weight functions [30,31,41,81,96]. For standard KBO, Lepper could prove that the derivation length of a KBO-terminating TRS is bound by a multiply recursive function [69]. In [35], Dick, Kalmus, and Martin presented the first automation of KBO, which turned out to be quite intricate despite the fact that KBO orientability is decidable in polynomial time [59]. Recent automations relying on SAT, pseudo-boolean, and SMT encodings turned out to be far more efficient and easier to implement [114,115].

We recall the standard definition of KBO. A *weight function* for the signature $\mathcal{F}$ is a pair (w, w_0) consisting of a mapping $w\colon \mathcal{F} \to \mathbb{N}$ and a positive constant $w_0 \in \mathbb{N}$ such that $w(c) \geqslant w_0$ for every constant $c \in \mathcal{F}$. A weight function (w, w_0) is *admissible* for a precedence $\succ$ if for every unary $f \in \mathcal{F}$ with $w(f) = 0$ we have $f \succ g$ for all $g \in \mathcal{F} \setminus \{f\}$. The weight of a term is computed as follows: $w(x) = w_0$ for $x \in \mathcal{V}$ and $w(f(t_1, \ldots, t_n)) = w(f) + w(t_1) + \cdots + w(t_n)$.

Definition 2.9 ([35, 57, 96])**.** Let $\succ$ be a precedence and (w, w_0) a weight function. We define the *Knuth-Bendix order* $\succ_{\mathrm{kbo}}$ inductively as follows: $s \succ_{\mathrm{kbo}} t$ if $|s|_x \geqslant |t|_x$ for all variables $x \in \mathcal{V}$ and either $w(s) > w(t)$, or $w(s) = w(t)$ and one of the following alternatives holds:

(1) $s = f^k(x)$ and $t = x$ for some $k > 0$, or

(2) $s = f(s_1, \ldots, s_n)$, $t = g(t_1, \ldots, t_m)$, and $f \succ g$, or

(3) $s = f(s_1, \ldots, s_n)$, $t = f(t_1, \ldots, t_n)$, $s_1 = t_1, \ldots, s_{k-1} = t_{k-1}$, and $s_k \succ_{\mathrm{kbo}} t_k$ with $1 \leqslant k \leqslant n$.

Theorem 2.10 ([35,57])**.** *The relation $\succ_{\mathrm{kbo}}$ is a simplification order.* □

A TRS $\mathcal{R}$ is called compatible with KBO if there exists a weight function (w, w_0) admissible for a precedence $\succ$ such that $\mathcal{R}$ is compatible with $\succ_{\mathrm{kbo}}$ using (w, w_0).

Theorem 2.11 ([39])**.** *A TRS is totally terminating if it is compatible with KBO.* □

Example 2.12. For $k > 1$, let the TRS $\mathcal{R}_k$ consist of the following rules:

$$\mathsf{f}^{k+2}(\mathsf{g}(x)) \to \mathsf{g}^k(x) \tag{1}$$
$$\mathsf{g}(\mathsf{g}(x)) \to \mathsf{f}(\mathsf{g}(x)) \tag{2}$$

Termination of any $\mathcal{R}_k$ can be proven by KBO using the weight function (w_0, w) with $w_0 = w(\mathsf{f}) = w(\mathsf{g}) = 1$ and precedence $\mathsf{g} > \mathsf{f}$: First of all, note that $\mathcal{R}_k$ is non-duplicating. Moreover, for rule (1), we have $w(\mathsf{f}^{k+2}(\mathsf{g}(x))) = k + 4 > k + 1 = w(\mathsf{g}^k(x))$. For rule (2) we have $w(\mathsf{g}(\mathsf{g}(x))) = 3 = w(\mathsf{f}(\mathsf{g}(x)))$ but $\mathsf{g} \succ \mathsf{f}$ such that the rule can be oriented by case (2) in Definition 2.9. Note that the TRSs $\mathcal{R}_k$ cannot be oriented by LPO.

Example 2.13. Consider the TRS consisting of the single rule

$$\mathsf{h}(x, x) \to \mathsf{f}(x)$$

Termination can be shown by KBO using (w_0, w) with $w_0 = 3$, $w(\mathsf{h}) = 0$, and $w(\mathsf{f}) = 2$ as the constraint on the weight gives $6 > 5$.

It has recently been shown that if a TRS $\mathcal{R}$ is compatible with $\succ_{\mathrm{kbo}}$ using some weight function (w, w_0), then there also exist weight functions (w_k, k) such that $\mathcal{R}$ is compatible with $\succ_{\mathrm{kbo}}$ using (w_k, k), for all positive numbers k [113].

Example 2.14. For the TRS in Example 2.13, setting $w_0 = 1$ yields $2 \not> 3$ in the weight comparison. But a valid KBO proof with $w_0 = 1$ can be obtained by adapting the weights of other function symbols as well, for example when setting $w(\mathsf{h}) = w(\mathsf{f}) = 2$ the weight comparison amounts to $4 > 3$.

For the sake of simplicity, we will thus in later examples set $w_0 = 1$, and write $\succ^w_{\mathrm{kbo}}$ to indicate the weight function used for $\succ_{\mathrm{kbo}}$.

2.3 Equational Proofs

Let the tuple $(\mathcal{E}, \mathcal{R})$ consist of a set of equations $\mathcal{E}$ and a set of rewrite rules $\mathcal{R}$. An *equational proof step* $s \leftrightarrow^p_e t$ in $(\mathcal{E}, \mathcal{R})$ is an *equality step* if e is an equation $\ell \simeq r$ in $\mathcal{E}$ or a *rewrite step* if e is a rule $\ell \to r$ in $\mathcal{R}$, and either $s = u[\ell\sigma]_p$ and $t = u[r\sigma]_p$ or $s = u[r\sigma]_p$ and $t = u[\ell\sigma]_p$ hold for some substitution σ and term u with position p. We sometimes write $s \leftrightarrow t$ to express the existence of some proof step, omitting the position p and equation or rule e. An *equational proof* P of an equation $t_0 \approx t_n$ is a finite sequence

$$t_0 \underset{e_0}{\overset{p_0}{\longleftrightarrow}} t_1 \underset{e_1}{\overset{p_1}{\longleftrightarrow}} \cdots \underset{e_{n-1}}{\overset{p_{n-1}}{\longleftrightarrow}} t_n \tag{2.1}$$

of equational proof steps, where $n \geqslant 0$. Note that $(\mathcal{E}, \mathcal{R})$ admits an equational proof of $s \approx t$ if and only if $s \leftrightarrow^*_{\mathcal{E}\cup\mathcal{R}} t$ holds. A sequence Q of the form $t_i \leftrightarrow \cdots \leftrightarrow t_j$ with $0 \leqslant i \leqslant j \leqslant n$ is a *subproof* of P. We write $P[Q]$ to

express that P contains Q as a subproof. If P is an equational proof and σ a substitution then $P\sigma$ denotes the instantiated proof

$$t_0\sigma \underset{e_0}{\overset{p_0}{\longleftrightarrow}} t_1\sigma \underset{e_1}{\overset{p_1}{\longleftrightarrow}} \cdots \underset{e_{n-1}}{\overset{p_{n-1}}{\longleftrightarrow}} t_n\sigma$$

For a term u with position q and a proof P of the shape (2.1) we write $u[P]_q$ to denote the sequence

$$u[t_0]_q \underset{e_0}{\overset{qp_0}{\longleftrightarrow}} u[t_1]_q \underset{e_1}{\overset{qp_1}{\longleftrightarrow}} \cdots \underset{e_{n-1}}{\overset{qp_{n-1}}{\longleftrightarrow}} u[t_n]_q$$

which is again an equational proof. Proofs of the shape $t_0 \to \cdots \to t_i \leftarrow \cdots \leftarrow t_n$ are called *rewrite proofs*. A *proof order* $\succcurlyeq$ is a well-founded order on equational proofs such that

i. $P \succcurlyeq Q$ implies $u[P\sigma]_p \succcurlyeq u[Q\sigma]_p$ for all terms u, positions $p \in \mathcal{P}\mathrm{os}(u)$ and substitutions σ, and

ii. if P and P' prove the same equation then $P \succcurlyeq P'$ implies $Q[P] \succcurlyeq Q[P']$ for all proofs Q.

A *proof reduction relation* $\Rightarrow$ additionally satisfies

iii. $P \Rightarrow Q$ holds only if P and Q prove the same equation.

We will write $\Rightarrow^=$ to denote the reflexive closure of $\Rightarrow$.

2.4 Rewriting Modulo a Set of Equalities

A term s *rewrites* to t in $\mathcal{R}$ *modulo* $\mathcal{T}$, denoted by $s \to_{\mathcal{R}/\mathcal{T}} t$, whenever $s \leftrightarrow_{\mathcal{T}}^* \cdot \to_{\mathcal{R}} \cdot \leftrightarrow_{\mathcal{T}}^* t$ holds. The system $\mathcal{R}$ *terminates modulo* $\mathcal{T}$ whenever the relation $\to_{\mathcal{R}/\mathcal{T}}$ is well-founded. It is *convergent modulo* $\mathcal{T}$ if in addition for every conversion $s \leftrightarrow_{\mathcal{T}\cup\mathcal{R}}^* t$ we have $s \to_{\mathcal{R}/\mathcal{T}}^* \cdot \leftrightarrow_{\mathcal{T}}^* \cdot {}_{\mathcal{R}/\mathcal{T}}^*\!\leftarrow t$. To check termination of $\mathcal{R}$ modulo $\mathcal{T}$ one can use $\mathcal{T}$*-compatible* reduction orders $\succ$ which satisfy $\leftrightarrow_{\mathcal{T}}^* \cdot \succ \cdot \leftrightarrow_{\mathcal{T}}^* \subseteq \succ$. Since the relation $\to_{\mathcal{R}/\mathcal{T}}$ is undecidable in general, one often considers the rewrite relation $\to_{\mathcal{R},\mathcal{T}}$ where $s \to_{\mathcal{R},\mathcal{T}} t$ holds if $s|_p \leftrightarrow_{\mathcal{T}}^* \ell\sigma$ and $t = s[r\sigma]_p$ for some $p \in \mathcal{P}\mathrm{os}(s)$, rule $\ell \to r$ in $\mathcal{R}$ and substitution σ [83]. We obviously have $\to_{\mathcal{R}} \subseteq \to_{\mathcal{R},\mathcal{T}} \subseteq \to_{\mathcal{R}/\mathcal{T}}$. Thus, if $\to_{\mathcal{R}}$ is convergent modulo $\mathcal{T}$ then also $\to_{\mathcal{R},\mathcal{T}}$ is convergent modulo $\mathcal{T}$ [6], and defines the same normal forms as $\to_{\mathcal{R}/\mathcal{T}}$. Hence, if $\mathcal{T}$-matching is decidable then $\to_{\mathcal{R},\mathcal{T}}$ constitutes a decidable way to compute with respect to $\to_{\mathcal{R}/\mathcal{T}}$.

We now extend the notion of critical pairs to *critical pairs modulo* $\mathcal{T}$. A substitution σ constitutes a $\mathcal{T}$*-unifier* of two terms s and t if $s\sigma \leftrightarrow_{\mathcal{T}}^* t\sigma$ holds. If unification with respect to $\mathcal{T}$ is finitary and decidable, then there is a complete set of $\mathcal{T}$-unifiers Σ for every pair of terms s and t. More precisely, this means that every $\sigma \in \Sigma$ is a $\mathcal{T}$-unifier of s and t, and conversely for every $\mathcal{T}$-unifier τ of the two terms there is some $\sigma \in \Sigma$ and substitution ρ such that $x\tau \leftrightarrow_{\mathcal{T}}^* x\sigma\rho$ for all variables x.

Definition 2.15. Let $\mathcal{T}$ be a theory for which unification is finitary and decidable. A $\mathcal{T}$-*overlap* is a quadruple

$$\langle \ell_1 \to r_1, p, \ell_2 \to r_2 \rangle_\Sigma \tag{2.2}$$

consisting of rewrite rules $\ell_1 \to r_1$, $\ell_2 \to r_2$, a position $p \in \mathcal{P}\mathrm{os}_\mathcal{F}(\ell_2)$, and a complete set Σ of $\mathcal{T}$-unifiers of $\ell_2|_p$ and ℓ_1. Then $\ell_2\sigma[r_1\sigma]_p \leftarrow \rtimes \rightarrow r_2\sigma$ constitutes a $\mathcal{T}$-*critical pair* for every $\sigma \in \Sigma$. Again $s \leftarrow \times \rightarrow t$ abbreviates $s \leftarrow \rtimes \rightarrow t$ or $t \leftarrow \rtimes \rightarrow s$, for any terms s and t. For two sets of rewrite rules $\mathcal{R}_1$ and $\mathcal{R}_2$, we also write $\mathrm{CP}_\mathcal{T}(\mathcal{R}_1, \mathcal{R}_2)$ for the set of all $\mathcal{T}$-critical pairs emerging from an overlap (2.2) where $\ell_1 \to r_1 \in \mathcal{R}_1$ and $\ell_2 \to r_2 \in \mathcal{R}_2$, and $\mathrm{CP}_\mathcal{T}(\mathcal{R}_1)$ for the set of all $\mathcal{T}$-critical pairs such that $\ell_1 \to r_1$, $\ell_2 \to r_2 \in \mathcal{R}_1$.

Again we will call a peak $s \; {}^{p}_{r \leftarrow \ell}\!\leftarrow \cdot \leftrightarrow^*_\mathcal{T} \cdot \rightarrow^q_{u \to v} t$ a *non-overlap* if it is not an instance of a $\mathcal{T}$-overlap. In contrast to standard critical pairs, even if two rules lack $\mathcal{T}$-critical pairs, they can still give rise to a non-joinable overlap.

Example 2.16. Consider the TRS $\mathcal{R} = \{\mathsf{a} + \mathsf{b} \to \mathsf{c},\ \mathsf{a} + \mathsf{c} \to \mathsf{b}\}$, where $+$ is an AC symbol. Despite the absence of AC-critical pairs, the term $\mathsf{a} + \mathsf{b} + \mathsf{c}$ can be reduced to both $\mathsf{b} + \mathsf{b}$ and $\mathsf{c} + \mathsf{c}$, and these terms are not joinable.

This problem gave rise to the notion of *extended rules*.

Definition 2.17 ([83])**.** Let $\ell \to r$ be a rewrite rule in a TRS $\mathcal{R}$ and $u \simeq v$ be a variable-disjoint equation in $\mathcal{T}$. If $u|_p$ is $\mathcal{T}$-unifiable with ℓ then $u[\ell]_p \to u[r]_p$ is a $\mathcal{T}$-*extended rule* of $\mathcal{R}$. The set of all $\mathcal{T}$-extended rules of $\mathcal{R}$ is denoted by $\mathrm{EXT}_\mathcal{T}(\mathcal{R})$.

To achieve convergence modulo some theory $\mathcal{T}$, completion procedures thus have to consider critical pairs involving extended rules, too. Note that in general extended rules create extended rules themselves: For instance, if $+$ is an AC symbol a rule of the form $s + t \to u$ gives rise to the extended rule $x + (s + t) \to x + u$, which can in turn be extended to $y + (x + (s + t)) \to y + (x + u)$.

2.5 AC Rewriting

We will now focus on the theory AC of associative and commutative operators. Such operators occur in many practical problems based upon algebraic structures such as commutative monoids, groups, rings or modules. Unification modulo AC is finitary and decidable, much work has been dedicated to the implementation of efficient unification algorithms (for an overview see [5]). But also AC termination and AC completion techniques have been subject to extensive research. Since AC completion and normalized completion, which is based upon AC rewriting, are treated in later sections, the basics of AC rewriting will be recalled here.

Let $\mathcal{F}_\mathrm{AC} \subseteq \mathcal{F}$ denote the set of all AC operators in the signature $\mathcal{F}$. The theory AC is then given by $\mathrm{AC} = \{f(x, y) \approx f(y, x),\ f(f(x, y), z) \approx f(x, f(y, z)) \mid f \in \mathcal{F}_\mathrm{AC}\}$.

We will write $\trianglerighteq_{\mathrm{AC}}$ for the subterm relation modulo AC defined as $\leftrightarrow^*_{\mathrm{AC}} \cdot \trianglerighteq \cdot \leftrightarrow^*_{\mathrm{AC}}$, and $\triangleright_{\mathrm{AC}}$ for its proper part, i.e., the relation $\trianglerighteq_{\mathrm{AC}} \setminus \leftrightarrow^*_{\mathrm{AC}}$. Furthermore, $\unrhd_{\mathrm{AC}}$ denotes encompassment modulo AC, where $s \unrhd_{\mathrm{AC}} t$ holds if and only if there exists some substitution σ such that $s \trianglerighteq_{\mathrm{AC}} t\sigma$. The proper encompassment relation modulo AC is denoted by $\rhd_{\mathrm{AC}}$, where $s \rhd_{\mathrm{AC}} t$ holds if and only if $s \unrhd_{\mathrm{AC}} t$ but not $t \unrhd_{\mathrm{AC}} s$.

Extended Rules

As remarked above, extended rules generally produce extended rules themselves. Fortunately, in case of AC it suffices to consider a finite set of extended rules $\mathcal{R}^e \subseteq \mathrm{EXT}_{\mathrm{AC}}(\mathcal{R})$.

Definition 2.18 ([6])**.** For a rewrite rule $\ell \to r$ with $f \in \mathcal{F}_{\mathrm{AC}}$ we write $(\ell \to r)^e$ for the rule $f(\ell, x) \to f(r, x)$, where $x \in \mathcal{V}$ is fresh. The TRS $\mathcal{R}^e$ contains all rules in $\mathcal{R}$ plus all rules $f(\ell, x) \to f(r, x)$ such that $\ell \to r \in \mathcal{R}$, $f \in \mathcal{F}_{\mathrm{AC}}$ and there is no rule $\ell' \to r'$ and substitution σ such that $f(\ell, x) \leftrightarrow^*_{\mathrm{AC}} \ell'\sigma$ and $f(r, x) \to^*_{\mathrm{AC} \cup (\mathcal{R}/\mathrm{AC})} r'\sigma$.

Lemma 2.19 ([6])**.** *Every TRS $\mathcal{R}$ satisfies $(\mathcal{R}^e)^e = \mathcal{R}^e$.* □

Termination Modulo AC

Considerable efforts have been devoted to the adaptation of standard reduction orders to the AC case. Concerning orderings resembling the recursive path order (RPO), these include [7, 16, 32, 53, 88]. A first version of an AC-compatible Knuth-Bendix order was presented by Steinbach [94], and Korovin and Voronkov proposed another version in [59]. Polynomial interpretations have been adapted to the AC setting by Ben Cherifa and Lescanne [19]. It is not hard to adapt their criterion to matrix interpretations [37], as argued in [111].

We will now describe AC-RPO as presented by Rubio [87] in order to illustrate examples in Chapter 6.

Let $\succ$ be a precedence on $\mathcal{F}$. We will restrict to the case where $\succ$ is total, although AC-RPO can also be defined for partial precedences [87]. Let moreover all non-AC function symbols with positive arity be partitioned into disjoint sets *Lex* and *Mul* having lexicographic and multiset status, respectively.

The order $\succ_{\mathrm{acrpo}}$ requires some preliminary definitions. First of all, *top-flattening* with respect to a function symbol f is defined as

$$\mathrm{tf}_f(t) = \begin{cases} t_1, \ldots, t_n & \text{if } t = f(t_1, \ldots, t_n) \\ t & \text{if } \mathrm{root}(t) \neq f \end{cases}$$

Next we consider a restricted notion of embedding. The set $\mathit{EmbSmall}(s)$ consists of all terms which are embedded in s through a direct subterm that is rooted by a smaller symbol.

Definition 2.20. Let $s = f(s_1, \ldots, s_n)$ be a term such that $f \in \mathcal{F}_{\mathrm{AC}}$. The multiset $\mathit{EmbSmall}(s)$ consists of all terms $f(s_1, \ldots, s_{i-1}, \mathrm{tf}_f(u_j), s_{i+1}, \ldots, s_n)$ such that $s_i = h(u_1, \ldots, u_m)$ and $f \succ h$, where $1 \leqslant j \leqslant m$ and $1 \leqslant i \leqslant n$.

Example 2.21. Consider the signature $\mathcal{F} = \{+, \cdot, -, 0, 1\}$ to describe an associative and commutative ring, where $\mathcal{F}_{\mathrm{AC}} = \{+, \cdot\}$. Let $\cdot \succ - \succ + \succ 1 \succ 0$. Then $\mathit{EmbSmall}((x + y) \cdot z) = \{x \cdot z, y \cdot z\}$ and $\mathit{EmbSmall}((-x) \cdot y) = \{x \cdot y\}$.

For another example, consider the signature $\mathcal{F} = \{\mathsf{N}, \mathsf{L}, \mathsf{max}, \mathsf{s}, 0\}$ where $\mathcal{F}_{\mathrm{AC}} = \{\mathsf{N}\}$. Let $\mathsf{s} \succ \mathsf{L} \succ \mathsf{N} \succ \mathsf{max} \succ 0$. For $s = \mathsf{N}(\mathsf{max}(\mathsf{N}(x, 0)), \mathsf{max}(y), \mathsf{L}(z))$ we obtain $\mathit{EmbSmall}(s) = \{\mathsf{N}(x, 0, \mathsf{max}(y), \mathsf{L}(z)), \mathsf{N}(\mathsf{max}(\mathsf{N}(x, 0)), y, \mathsf{L}(z))\}$. For the terms $t = \mathsf{N}(\mathsf{L}(\mathsf{max}(\mathsf{N}(y, z))), x)$ and $u = \mathsf{N}(x, y, z)$ we have $\mathit{EmbSmall}(t) = \mathit{EmbSmall}(u) = \varnothing$.

We need notions to refer to two kinds of subsets of direct subterms.

Definition 2.22. For a term $s = f(s_1, \ldots, s_n)$ such that $f \in \mathcal{F}_{\mathrm{AC}}$ the multisets $\mathit{BigHead}(s)$ and $\mathit{NoSmallHead}(s)$ are defined as $\mathit{BigHead}(s) = \{s_i \mid \mathrm{root}(s_i) \succ f\}$ and $\mathit{NoSmallHead}(s) = \{s_i \mid \mathrm{root}(s_i) \not\prec f\}$.

While $\mathit{BigHead}(s)$ contains all direct subterms whose root symbol is greater than f, the multiset $\mathit{NoSmallHead}(s)$ includes also all direct subterms whose root symbol is not smaller than f. Thus the former includes variables but the latter does not.

Example 2.23. Consider the terms $t = \mathsf{N}(\mathsf{L}(\mathsf{max}(\mathsf{N}(y, z))), x)$ and $u = \mathsf{N}(x, y, z)$ with the above precedence. We have $\mathit{NoSmallHead}(t) = \{\mathsf{L}(\mathsf{max}(\mathsf{N}(y, z))), x\}$ and $\mathit{NoSmallHead}(u) = \{x, y, z\}$, but $\mathit{BigHead}(t) = \{\mathsf{L}(\mathsf{max}(\mathsf{N}(y, z)))\}$ and $\mathit{BigHead}(u) = \varnothing$.

Definition 2.24. For a term $s = f(s_1, \ldots, s_n)$ and $f \in \mathcal{F}_{\mathrm{AC}}$ we define $\#(s) = \#(s_1) + \ldots + \#(s_n)$ such that $\#(x) = x$ if $x \in \mathcal{V}$ and $\#(s_i) = 1$ otherwise.

For instance, $\#(x \cdot z \cdot 1) = x + z + 1$, $\#((x + y) \cdot z) = z + 1$ and $\#(x \cdot z) = x + z$. When comparing these linear polynomials we assume $\mathbb{N}_{>0}$ as a carrier. Thus for instance $\#(x \cdot z \cdot 1) > \#(x \cdot z)$ and $\#(x \cdot z \cdot 1) > \#((x + y) \cdot z)$ hold, but also $\#(x \cdot z) \geqslant \#((x + y) \cdot z)$.

Definition 2.25 ([87])**.** Let s and t be terms such that $s = f(s_1, \ldots, s_n)$. Then $s \succ_{\mathrm{acrpo}} t$ if and only if

(1) $s_i \succeq_{\mathrm{acrpo}} t$ for some $1 \leqslant i \leqslant n$,

(2) $t = g(t_1, \ldots, t_m)$, $f \succ g$, and $s \succ_{\mathrm{acrpo}} t_j$ for all $1 \leqslant j \leqslant m$

(3) $t = f(t_1, \ldots, t_n)$, $f \in \mathit{Lex}$, $(s_1, \ldots, s_n) \succ^{\mathsf{lex}}_{\mathrm{acrpo}} (t_1, \ldots, t_n)$, and $s \succ_{\mathrm{acrpo}} t_j$ for all $1 \leqslant j \leqslant n$,

(4) $t = f(t_1, \ldots, t_n)$, $f \in \mathit{Mul}$, and $\{s_1, \ldots, s_n\} \succ^{\mathsf{mul}}_{\mathrm{acrpo}} \{t_1, \ldots, t_n\}$,

(5) $t = f(t_1, \ldots, t_m)$, $f \in \mathcal{F}_{\mathrm{AC}}$, and $s' \succeq_{\mathrm{acrpo}} t$ for some $s' \in \mathit{EmbSmall}(s)$, or

(6) $t = f(t_1, \ldots, t_m)$, $f \in \mathcal{F}_{\mathrm{AC}}$, $s \succ_{\mathrm{acrpo}} t'$ for all terms $t' \in \mathit{EmbSmall}(t)$, $\mathit{NoSmallHead}(s) \succeq^{\mathsf{mul}}_{\mathrm{acrpo}} \mathit{NoSmallHead}(t)$, and

   (6a) $\mathit{BigHead}(s) \succ^{\mathsf{mul}}_{\mathrm{acrpo}} \mathit{BigHead}(t)$, or

   (6b) $\#(s) > \#(t)$, or

(6c) $\#(s) \geqslant \#(t)$ and $\{s_1, \ldots, s_n\} \succ^{\mathsf{mul}}_{\text{acrpo}} \{t_1, \ldots, t_n\}$.

where $\succeq_{\text{acrpo}}$ denotes $\succ_{\text{acrpo}} \cup \leftrightarrow^*_{\text{AC}}$.

Theorem 2.26 ([87])**.** *The relation $\succ_{\text{acrpo}}$ is an AC-compatible simplification order.* □

Example 2.27. Let $\mathcal{F} = \{+, \cdot, -, 0, 1\}$ and $\mathcal{F}_{\text{AC}} = \{+, \cdot\}$. The following TRS describing an associative and commutative ring is compatible with AC-RPO.

$$0 + x \to x \tag{1}$$
$$-x + x \to 0 \tag{2}$$
$$-0 \to 0 \tag{3}$$
$$-(-x) \to x \tag{4}$$
$$-(x + y) \to (-x) + (-y) \tag{5}$$
$$1 \cdot x \to x \tag{6}$$
$$(x + y) \cdot z \to x \cdot z + y\cdot \tag{7}$$
$$0 \cdot x \to 0 \tag{8}$$
$$(-x) \cdot y \to -(x \cdot y) \tag{9}$$

Let $\cdot \succ - \succ + \succ 1 \succ 0$ and $\mathit{Lex} = \{-\}$.

- Rules (1), (3), (4), (6), and (8) can be oriented by repeatedly applying case (1) in Definition 2.25 since the right-hand sides are subterms of the respective left-hand sides.
- Rule (2) can clearly be oriented by case (2) as $+ \succ 0$.
- Rule (5) can also be handled by case (2): since $- \succ +$ it remains to check that $-(x+y) \succ_{\text{acrpo}} -x$ and $-(x+y) \succ_{\text{acrpo}} -y$. This holds by cases (3) and (1) as $- \in \mathit{Lex}$ and x, y are subterms of $x + y$.
- Case (2) also applies to rule (7) as $\cdot \succ +$. It remains to check that $(x+y)\cdot z \succ_{\text{acrpo}} x\cdot z$ and $(x+y)\cdot z \succ_{\text{acrpo}} y\cdot z$. Since $\mathit{EmbSmall}((x+y)\cdot z) = \{x \cdot z, y \cdot z\}$ this holds by case (5).
- Finally, we can apply case (2) to rule (9) since $\cdot \succ -$. The remaining goal $(-x) \cdot y \succ_{\text{acrpo}} x \cdot y$ holds by case (5) as $x \cdot y \in \mathit{EmbSmall}((-x) \cdot y)$.

Note that in some of the comparisons required to orient rules (5), (7), and (9) one could also argue that $\succ_{\text{acrpo}}$ is a simplification order and thus contains the embedding relation.

The following TRS is a variation of a system presented in [64].

Example 2.28. Let $\mathcal{F} = \{\mathsf{N}, \mathsf{L}, \mathsf{max}, \mathsf{s}, 0\}$ and $\mathcal{F}_{\text{AC}} = \{\mathsf{N}\}$ such that $\mathsf{s} \succ \mathsf{L} \succ \mathsf{N} \succ \mathsf{max} \succ 0$ and $\mathit{Lex} = \{\mathsf{max}, \mathsf{s}\}$. Termination of the following TRS describing the maximum of a binary tree can be shown by AC-RPO.

$$\mathsf{max}(\mathsf{L}(x)) \to x \tag{1}$$
$$\mathsf{max}(\mathsf{N}(\mathsf{L}(0), \mathsf{L}(x))) \to x \tag{2}$$
$$\mathsf{s}(\mathsf{max}(\mathsf{N}(\mathsf{L}(x), \mathsf{L}(y)))) \to \mathsf{max}(\mathsf{N}(\mathsf{L}(\mathsf{s}(x)), \mathsf{L}(\mathsf{s}(y)))) \tag{3}$$
$$\mathsf{max}(\mathsf{N}(\mathsf{L}(\mathsf{max}(\mathsf{N}(y, z))), x)) \to \mathsf{max}(\mathsf{N}(x, y, z)) \tag{4}$$

– Rules (1) and (2) can be oriented by repeated application of case (1) in Definition 2.25.

– Concerning rule (3), we can repeatedly apply case (2) as s is greater than all of max, N, and L. This leaves to check $\mathsf{s}(\mathsf{max}(\mathsf{N}(\mathsf{L}(x), \mathsf{L}(y)))) \succ_{\text{acrpo}} \mathsf{s}(x)$ and $\mathsf{s}(\mathsf{max}(\mathsf{N}(\mathsf{L}(x), \mathsf{L}(y)))) \succ_{\text{acrpo}} \mathsf{s}(y)$, which holds by cases (3) and (1).

– To rule (4) we can apply case (3). It remains to check that the terms $s = \mathsf{N}(\mathsf{L}(\mathsf{max}(\mathsf{N}(y, z))), x)$ and $t = \mathsf{N}(x, y, z)$ satisfy $s \succ_{\text{acrpo}} t$. Here case (5) is not applicable as $\mathit{EmbSmall}(s) = \varnothing$. But case (6) applies since $\mathit{EmbSmall}(t) = \varnothing$, and we have $\mathit{NoSmallHead}(s) \succeq^{\mathsf{mul}}_{\text{acrpo}} \mathit{NoSmallHead}(t)$ because $\mathit{NoSmallHead}(s) = \{\mathsf{L}(\mathsf{max}(\mathsf{N}(y, z))), x\}$ and $\mathit{NoSmallHead}(t) = \{x, y, z\}$. As $\mathit{BigHead}(s) = \{\mathsf{L}(\mathsf{max}(\mathsf{N}(y, z)))\} \succeq^{\mathsf{mul}}_{\text{acrpo}} \varnothing = \mathit{BigHead}(t)$ subcase (6a) can be applied such that rule (4) is oriented.

The final example illustrates cases (6b) and (6c) in Definition 2.25.

Example 2.29 ([87])**.** Let $\mathcal{F} = \{\mathsf{f}, \mathsf{g}, \mathsf{h}\}$, $\mathit{Lex} = \{\mathsf{g}, \mathsf{h}\}$, and $\mathcal{F}_{\text{AC}} = \{\mathsf{f}\}$ such that $\mathsf{h} \succ \mathsf{f} \succ \mathsf{g}$ and consider the following TRS:

$$\mathsf{f}(\mathsf{g}(\mathsf{f}(\mathsf{h}(x), x)), x) \to \mathsf{f}(\mathsf{h}(x), x, x) \tag{1}$$
$$\mathsf{f}(\mathsf{h}(x), \mathsf{g}(x)) \to \mathsf{f}(\mathsf{g}(\mathsf{h}(x)), x) \tag{2}$$
$$\mathsf{f}(\mathsf{g}(\mathsf{h}(x)), x, x, y) \to \mathsf{f}(\mathsf{g}(\mathsf{f}(\mathsf{h}(x), y)), x) \tag{3}$$
$$\mathsf{f}(\mathsf{g}(\mathsf{g}(x)), x) \to \mathsf{f}(\mathsf{g}(x), \mathsf{g}(x)) \tag{4}$$

– Rule (1) is oriented by case (5) as for $\ell = \mathsf{f}(\mathsf{g}(\mathsf{f}(\mathsf{h}(x), x)), x)$ and $r = \mathsf{f}(\mathsf{h}(x), x, x)$ we have $r \in \mathit{EmbSmall}(\ell)$.

– In case of rule (2), for $\ell = \mathsf{f}(\mathsf{h}(x), \mathsf{g}(x))$ and $r = \mathsf{f}(\mathsf{g}(\mathsf{h}(x)), x)$ one obtains $\ell \succ_{\text{acrpo}} r$ by case (6a): We have $\mathit{EmbSmall}(r) = \{\mathsf{f}(\mathsf{h}(x), x)\}$ and $\ell \succ_{\text{acrpo}} \mathsf{f}(\mathsf{h}(x), x)$ by case (5), $\mathit{NoSmallHead}(\ell) = \{\mathsf{h}(x)\} \succeq^{\mathsf{mul}}_{\text{acrpo}} \{x\} = \mathit{NoSmallHead}(r)$, and $\mathit{BigHead}(\ell) = \{\mathsf{h}(x)\} \succ^{\mathsf{mul}}_{\text{acrpo}} \varnothing = \mathit{BigHead}(r)$.

– In case of rule (3), for $\ell = \mathsf{f}(\mathsf{g}(\mathsf{h}(x)), x, x, y)$ and $r = \mathsf{f}(\mathsf{g}(\mathsf{f}(\mathsf{h}(x), y)), x)$ we conclude $\ell \succ_{\text{acrpo}} r$ by case (6b): $\mathit{NoSmallHead}(\ell) = \{x, x, y\} \succeq^{\mathsf{mul}}_{\text{acrpo}} \{x\} = \mathit{NoSmallHead}(r)$, and $\#(\ell) = 2x + y + 1 > x + 1 = \#(r)$. Moreover, $\mathit{EmbSmall}(r)$ is singleton containing $r' = \mathsf{f}(\mathsf{h}(x), y, x)$ and we have $\ell \succ_{\text{acrpo}} r'$: by case (5), it suffices to show that $\ell' = \mathsf{f}(\mathsf{h}(x), x, x, y) \in \mathit{EmbSmall}(\ell)$ satisfies $\ell' \succ_{\text{acrpo}} r'$, and this holds by case (6b) because $\mathit{NoSmallHead}(\ell') = \{\mathsf{h}(x), x, x, y\} \succeq^{\mathsf{mul}}_{\text{acrpo}} \{\mathsf{h}(x), y, x\} = \mathit{NoSmallHead}(r')$, $\mathit{EmbSmall}(r') = \varnothing$, and $\#(\ell') = 2x + y + 1 > x + y + 1 = \#(r')$. Note that cases (5) and (6a) do not apply here.

– In case of rule (4), for $\ell = \mathsf{f}(\mathsf{g}(\mathsf{g}(x)), x)$ and $r = \mathsf{f}(\mathsf{g}(x), \mathsf{g}(x))$ we can apply case (6c). First of all, the terms in $\mathit{EmbSmall}(r) = \{\mathsf{f}(\mathsf{g}(x), x), \mathsf{f}(x, \mathsf{g}(x))\}$ are smaller than ℓ by case (5). Next, $\mathit{NoSmallHead}(\ell) = \{x\} \succeq^{\mathsf{mul}}_{\text{acrpo}} \varnothing = \mathit{NoSmallHead}(r)$ and $\#(\ell) = x + 1 \geqslant 2 = \#(r)$. Finally, we conclude $\ell \succ_{\text{acrpo}} r$ as $\{\mathsf{g}(\mathsf{g}(x)), x\} \succ^{\mathsf{mul}}_{\text{acrpo}} \{\mathsf{g}(x), \mathsf{g}(x)\}$ by case (3). Note that none of the other cases applies here.

Chapter 3

Knuth-Bendix Completion

Since the landmark paper by Knuth and Bendix [57], completion and its variants have been among the most influential approaches in automated reasoning. A completion procedure attempts to transform a set of input equalities $\mathcal{E}$ into a convergent rewrite system $\mathcal{R}$ with the same equational theory. It is parameterized by a reduction order $\succ$, the choice of which is critical to the outcome of a deduction. If successful, the derived TRS $\mathcal{R}$ constitutes a simple decision procedure for the validity problem of the theory under consideration: given an equation $s \approx t$, the two terms are convertible in $\mathcal{E}$ if and only if both terms reduce to the same normal form with respect to $\mathcal{R}$. As the validity problem is undecidable in general and decision procedures for equational theories have applications in many areas of formal methods and symbolic computation, such procedures are of great interest.

Although the first completion procedure is due to Knuth and Bendix, the method actually bears significant similarities with Buchberger's algorithm to compute Gröbner bases [22]. Since then completion procedures have been subject to extensive research. To name only some of the more influential contributions, Huet [46] gave the first correctness proof of a completion procedure with *inter-reduction*. Allowing for the simplification of already derived rewrite rules, inter-reduction significantly increases a completion procedure's efficiency. Abstracting from concrete algorithms, Bachmair, Dershowitz, and Hsiang [6, 12] provide a more flexible presentation of completion by means of an inference system. Plaisted and Sattler-Klein [85] showed that the proof lengths in completion procedures can get very long. Several tools implementing Knuth-Bendix completion were developed, e.g. [23,25,70]. To avoid failure due to unorientable equations, classical completion was extended to unfailing completion (see Chapter 5). As equational systems in algebra frequently feature common subtheories such as associative-commutative operators, completion procedures modulo built-in theories were developed (see Chapter 6). More recently, different completion approaches such as multi-completion and the use of termination tools and modern SAT/SMT solvers were developed (see Chapter 4).

In this chapter we will in Section 3.1 recall abstract equational inference systems as presented by Bachmair and Dershowitz [11]. This abstract setting allows us to state preliminaries about different completion-like procedures. Section 3.2 deals with classical Knuth-Bendix completion.

3.1 Abstract Equational Inference Systems

Let $\Rightarrow$ be a proof reduction relation, which is to capture some notion of a proof being preferable to another one. We will in the sequel consider inference sequences where each step can be simulated by applications of the following, very general inference rules.

Definition 3.1. Let $\mathcal{T}$ be a fixed equational theory, $\mathcal{E}$ a set of equations, and $\mathcal{R}$ a set of rewrite rules. For a proof reduction relation $\Rightarrow$, let *expansion* and *contraction* rules be defined as follows:

$$\textsf{expand}\quad \frac{\mathcal{E},\mathcal{R}}{\mathcal{E}\cup\{s\approx t\},\mathcal{R}} \qquad \text{if } s \leftrightarrow^*_{\mathcal{E}\cup\mathcal{R}\cup\mathcal{T}} t$$

$$\frac{\mathcal{E},\mathcal{R}}{\mathcal{E},\mathcal{R}\cup\{s\to t\}} \qquad \text{if } s \leftrightarrow^*_{\mathcal{E}\cup\mathcal{R}\cup\mathcal{T}} t$$

$$\textsf{contract}\quad \frac{\mathcal{E}\uplus\{s\simeq t\},\mathcal{R}}{\mathcal{E},\mathcal{R}} \qquad \text{if } s \leftrightarrow^\epsilon_{s\approx t} t \Rightarrow^= s \leftrightarrow^*_{\mathcal{E}\cup\mathcal{R}\cup\mathcal{T}} t$$

$$\frac{\mathcal{E},\mathcal{R}\uplus\{s\to t\}}{\mathcal{E},\mathcal{R}} \qquad \text{if } s \leftrightarrow^\epsilon_{s\to t} t \Rightarrow^= s \leftrightarrow^*_{\mathcal{E}\cup\mathcal{R}\cup\mathcal{T}} t$$

A derivation γ of the form $(\mathcal{E}_0,\mathcal{R}_0)\vdash_{\mathcal{S}}(\mathcal{E}_1,\mathcal{R}_1)\vdash_{\mathcal{S}}(\mathcal{E}_2,\mathcal{R}_2)\vdash_{\mathcal{S}}\cdots$ with respect to a set of inference rules $\mathcal{S}$ where each step can be simulated by a combination of expansion and contraction steps is called an *equational* inference sequence with respect to $\Rightarrow$ and $\mathcal{T}$.

Note that this abstract notion of equational inference sequences resembles the approach taken in [11]. In contrast to [11] we distinguish between equations and rules since the inference systems discussed in the next sections strongly rely on this distinction. In particular, replacing an equation by a rule as done in the orient rule in KB needs to be reflected in the proof order.

In the remainder of this section we will consider equational inference sequences with respect to an inference system $\mathcal{S}$, a proof reduction relation $\Rightarrow$ and some fixed theory $\mathcal{T}$. We write $(\mathcal{E},\mathcal{R})\vdash_{\mathcal{S}}(\mathcal{E}',\mathcal{R}')$ if $(\mathcal{E}',\mathcal{R}')$ was obtained from $(\mathcal{E},\mathcal{R})$ by applying an inference rule in $\mathcal{S}$. An inference sequence

$$(\mathcal{E}_0,\mathcal{R}_0)\vdash_{\mathcal{S}}(\mathcal{E}_1,\mathcal{R}_1)\vdash_{\mathcal{S}}(\mathcal{E}_2,\mathcal{R}_2)\vdash_{\mathcal{S}}\cdots$$

is also called a *run*. The sets of *persistent* equations and rules are defined as $\mathcal{E}_\omega=\bigcup_{i\geqslant 0}\bigcap_{j\geqslant i}\mathcal{E}_i$ and $\mathcal{R}_\omega=\bigcup_{i\geqslant 0}\bigcap_{j\geqslant i}\mathcal{R}_i$, respectively. For finite inference sequences $(\mathcal{E}_0,\mathcal{R}_0)\vdash^*_{\mathcal{S}}(\mathcal{E}_n,\mathcal{R}_n)$ the limit $(\mathcal{E}_\omega,\mathcal{R}_\omega)$ coincides with $(\mathcal{E}_n,\mathcal{R}_n)$. We will write $\vdash^=_{\mathcal{S}}$ to denote the reflexive closure of $\vdash_{\mathcal{S}}$. Sometimes we will also write $(\mathcal{E}_0,\mathcal{R}_0)\vdash^\alpha_{\mathcal{S}}(\mathcal{E}_\alpha,\mathcal{R}_\alpha)$ to indicate that the sequence has length α, where $\alpha=\omega$ if it is infinite.

It is not hard to see that equational inference systems are *sound* in that the equational theory is not modified in the course of a derivation.

Soundness Lemma 3.2. *For any equational run* $\gamma\colon(\mathcal{E}_0,\mathcal{R}_0)\vdash^\alpha_{\mathcal{S}}(\mathcal{E},\mathcal{R})$ *such that* $\alpha\leqslant\omega$ *the conversion* $\leftrightarrow^*_{\mathcal{E}_0\cup\mathcal{R}_0\cup\mathcal{T}}$ *coincides with* $\leftrightarrow^*_{\mathcal{E}\cup\mathcal{R}\cup\mathcal{T}}$.

Proof. As γ is an equational run it can be simulated by an inference sequence $(\mathcal{E}_0, \mathcal{R}_0) \vdash (\mathcal{E}_1, \mathcal{R}_1) \vdash (\mathcal{E}_2, \mathcal{R}_2) \vdash \cdots$ using expand and contract steps such that $(\mathcal{E}_\alpha, \mathcal{R}_\alpha) = (\mathcal{E}, \mathcal{R})$ for some $\alpha \leqslant \omega$.

We apply transfinite induction on α and abbreviate $\leftrightarrow^*_{\mathcal{E}_\beta \cup \mathcal{R}_\beta \cup \mathcal{T}}$ by $\leftrightarrow^*_\beta$ for any $\beta \leqslant \omega$. The statement clearly holds for $\alpha = 0$. Now consider an inference sequence $(\mathcal{E}_0, \mathcal{R}_0) \vdash^* (\mathcal{E}_\alpha, \mathcal{R}_\alpha) \vdash (\mathcal{E}_{\alpha+1}, \mathcal{R}_{\alpha+1})$. By the induction hypothesis $s \leftrightarrow^*_0 t$ if and only if $s \leftrightarrow^*_\alpha t$. Assume in step $(\mathcal{E}_\alpha, \mathcal{R}_\alpha) \vdash (\mathcal{E}_{\alpha+1}, \mathcal{R}_{\alpha+1})$ an expand rule is applied. Clearly, $\leftrightarrow^*_\alpha \subseteq \leftrightarrow^*_{\alpha+1}$ as no equation or rule is removed, and $\leftrightarrow^*_{\alpha+1} \subseteq \leftrightarrow^*_\alpha$ is true because the only additional equation $s \simeq t$ in $\mathcal{E}_{\alpha+1}$ or rule $s \to t$ in $\mathcal{R}_{\alpha+1}$ satisfies $s \leftrightarrow^*_{\mathcal{E}_\alpha \cup \mathcal{R}_\alpha \mathcal{T}} t$ by definition. If a contract rule is applied then $\leftrightarrow^*_{\alpha+1} \subseteq \leftrightarrow^*_\alpha$ is obvious, and the reverse direction holds as for the only removed equation $s \simeq t$ or rule $s \to t$ the side condition demands the existence of an equational proof $s \leftrightarrow^*_{\alpha+1} t$.

If $\alpha = \omega$ then by the induction hypothesis all $\leftrightarrow^*_n$ coincides with $s \leftrightarrow^*_0 t$ for all $n < \omega$. By the definition of $\mathcal{E}_\omega$ and $\mathcal{R}_\omega$ this also holds for $\leftrightarrow^*_\omega$. □

The following two lemmas from [11] show that equational proofs are preserved by inference steps and never increase with respect to $\Rightarrow$. In particular, every equational proof which is possible in some state of an inference sequence has a *persisting* proof in the limit.

Reflection Lemma 3.3. *For any finite equational run $\gamma\colon (\mathcal{E}_0, \mathcal{R}_0) \vdash^*_{\mathcal{S}} (\mathcal{E}, \mathcal{R})$ and equational proof P in $(\mathcal{E}_0 \cup \mathcal{T}, \mathcal{R}_0)$ there exists a proof Q in $(\mathcal{E} \cup \mathcal{T}, \mathcal{R})$ such that $P \Rightarrow^= Q$.*

Proof. As γ is an equational run it can be simulated by an inference sequence $(\mathcal{E}_0, \mathcal{R}_0) \vdash (\mathcal{E}_1, \mathcal{R}_1) \vdash (\mathcal{E}_2, \mathcal{R}_2) \vdash \cdots \vdash (\mathcal{E}_n, \mathcal{R}_n)$ using expand and contract steps such that $(\mathcal{E}_n, \mathcal{R}_n) = (\mathcal{E}, \mathcal{R})$.

We apply induction on the length n of this sequence. If $n = 0$ then we can set $Q = P$ such that $P \Rightarrow^= Q$. Now consider an inference sequence $(\mathcal{E}_0, \mathcal{R}_0) \vdash (\mathcal{E}_n, \mathcal{R}_n) \vdash (\mathcal{E}_{n+1}, \mathcal{R}_{n+1})$ and some proof P in $(\mathcal{E}_0 \cup \mathcal{T}, \mathcal{R}_0)$. By the induction hypothesis there is some proof Q' in $(\mathcal{E}_n \cup \mathcal{T}, \mathcal{R}_n)$ such that $P \Rightarrow^= Q'$. If an expand rule is applied in the last step, or if contract is applied but the removed equation $s \simeq t$ or rule $s \to t$ does not occur in Q' then we can set $Q = Q'$. Otherwise, $Q' = Q'[u \leftrightarrow_{s \approx t} v]$ or $Q' = Q'[u \leftrightarrow_{s \to t} v]$, so let $u = C[s\sigma]$ and $v = C[t\sigma]$ for some context C and substitution σ. By definition of contract there is a proof $P'\colon s \leftrightarrow_{\mathcal{E}_{n+1} \cup \mathcal{R}_{n+1} \cup \mathcal{T}} t$ which is smaller or equal than $s \leftrightarrow^\epsilon_{s \approx t} t$ or $s \leftrightarrow^\epsilon_{s \to t} t$, correspondingly. Thus we also have $Q' \Rightarrow Q'[C[P'\sigma]]$, and hence $P \Rightarrow Q'[C[P'\sigma]]$. □

Persistence Lemma 3.4. *Let $(\mathcal{E}_0, \mathcal{R}_0) \vdash_{\mathcal{S}} (\mathcal{E}_1, \mathcal{R}_1) \vdash_{\mathcal{S}} (\mathcal{E}_2, \mathcal{R}_2) \vdash_{\mathcal{S}} \cdots$ be a (possibly infinite) equational run and P a proof in $(\mathcal{E}_i \cup \mathcal{T}, \mathcal{R}_i)$ for some $i \geqslant 0$. Then there is a proof Q in $(\mathcal{E}_\omega \cup \mathcal{T}, \mathcal{R}_\omega)$ such that $P \Rightarrow^= Q$.*

Proof. By Lemma 3.3, for any proof P_i in $(\mathcal{E}_i \cup \mathcal{T}, \mathcal{R}_i)$ for some i there are proofs P_j in $(\mathcal{E}_j \cup \mathcal{T}, \mathcal{R}_j)$ for all $j \geqslant i$ such that $P_i \Rightarrow^= P_{i+1} \Rightarrow^= P_{i+2} \Rightarrow^= \cdots$. Since $\Rightarrow$ is terminating, there must be some k such that $P_j = P_k$ for all $j \geqslant k$. Hence P_k can use only equations in $\mathcal{E}_\omega \cup \mathcal{T}$ and rules in $\mathcal{R}_\omega$, so the proof $Q = P_k$ persists in the limit. □

In Chapters 3, 5, and 6, different inference systems for variants of completion will be discussed. In order to exploit results of this section, the presentations of completion-like procedures will pursue the following plan. We will first define a suitable proof reduction relation $\Rightarrow$ and identify a class of desirable normal form proofs. We next present a set of inference rules and show that every derivation constitutes an equational inference sequence with respect to $\Rightarrow$. The results on fairness, correctness, and completeness will then frequently rely on the soundness and persistence properties discussed on an abstract level in this section.

3.2 Standard Completion

The presentation of standard completion adopted here follows the similar approach pursued in [11] to match the results for equational inference systems in Section 3.1.

Given a set of input equalities $\mathcal{E}$, Knuth-Bendix completion aims to construct a TRS $\mathcal{R}$ such that for all terms s, t with $s \leftrightarrow_{\mathcal{E}}^* t$ there is a *rewrite proof* $s \to_{\mathcal{R}}^* \cdot \,{}_{\mathcal{R}}^*\!\leftarrow t$ in $\mathcal{R}$. An equational proof is considered in normal form if and only if it is a rewrite proof. Let $\succ$ be a reduction order. Knuth-Bendix completion operates on tuples $(\mathcal{E}, \mathcal{R})$ of a set of equations $\mathcal{E}$ and a set of rewrite rules $\mathcal{R}$. The latter will always be contained in $\succ$. We will use the following proof reduction relation $\Rightarrow_{\mathsf{KB}}^{\succ}$ (cf. [11]).

Definition 3.5. Let $\mathcal{E}$ be a set of equations, $\mathcal{R}$ a rewrite system and $\succ$ a reduction order containing $\mathcal{R}$. The *cost* of an equational proof step differentiates rewrite steps from equation steps and is defined as follows:

$$c(s \xleftarrow[\ell \to r]{p} t) = c(t \xleftarrow[r \leftarrow \ell]{p} s) = (\{s\}, s|_p, \ell, t) \qquad \text{if } \ell \to r \in \mathcal{R}$$
$$c(s \xleftarrow[u \approx v]{} t) = (\{s, t\}, \bot, \bot, \bot) \qquad \text{if } u \simeq v \in \mathcal{E}$$

To compare cost tuples we use the lexicographic combination of $\succ_{\mathsf{mul}}$, $\rhd$, $\trianglerighteq$ and $\succ$, where $\bot$ is considered minimal in the latter three orderings. The cost of an equational proof is the multiset consisting of the costs of its steps. The proof order $\succcurlyeq_{\mathsf{KB}}$ on equational proofs is the multiset extension of the order on proof step costs, and $\Rightarrow_{\mathsf{KB}}^{\succ}$ is the restriction of $\succcurlyeq_{\mathsf{KB}}$ such that $P \Rightarrow_{\mathsf{KB}}^{\succ} Q$ holds only if P and Q prove the same equation.

Lemma 3.6 ([11])**.** *The relation $\Rightarrow_{\mathsf{KB}}^{\succ}$ is a proof reduction relation.* □

Figure 3.1 displays the inference system KB of standard completion. A KB inference sequence of the form $(\mathcal{E}_0, \mathcal{R}_0) \vdash (\mathcal{E}_1, \mathcal{R}_1) \vdash (\mathcal{E}_2, \mathcal{R}_2) \vdash \cdots$ will be referred to as a *run*. We will in the sequel assume that $\mathcal{R}_0 = \varnothing$ although all results in the remainder of this section generalize to the setting where $\mathcal{R}_0$ is non-empty, provided that $\mathcal{R}_0 \subseteq \succ$. It is easy to see that this implies $\mathcal{R}_i \subseteq \succ$ for all $i \geqslant 0$. A run *fails* if $\mathcal{E}_\omega$ is not empty, it *succeeds* if $\mathcal{E}_\omega$ is empty and $\mathcal{R}_\omega$ is convergent for $\mathcal{E}_0$.

The following lemma shows that KB runs are equational inference sequences (where the theory $\mathcal{T}$ from Definition 3.1 is empty). We will drop the subscript

orient	$\dfrac{\mathcal{E} \uplus \{s \simeq t\}, \mathcal{R}}{\mathcal{E}, \mathcal{R} \cup \{s \to t\}}$	if $s \succ t$
deduce	$\dfrac{\mathcal{E}, \mathcal{R}}{\mathcal{E} \cup \{s \approx t\}, \mathcal{R}}$	if $s \leftarrow \rtimes \rightarrow t \in \mathrm{CP}(\mathcal{R})$
delete	$\dfrac{\mathcal{E} \uplus \{s \approx s\}, \mathcal{R}}{\mathcal{E}, \mathcal{R}}$	
simplify	$\dfrac{\mathcal{E} \uplus \{s \simeq t\}, \mathcal{R}}{\mathcal{E} \cup \{s \simeq u\}, \mathcal{R}}$	if $t \to_{\mathcal{R}} u$
compose	$\dfrac{\mathcal{E}, \mathcal{R} \uplus \{s \to t\}}{\mathcal{E}, \mathcal{R} \cup \{s \to u\}}$	if $t \to_{\mathcal{R}} u$
collapse	$\dfrac{\mathcal{E}, \mathcal{R} \uplus \{t \to s\}}{\mathcal{E} \cup \{u \approx s\}, \mathcal{R}}$	if $t \to_{\mathcal{R}} u$ using $\ell \to r$ such that $t \rhd \ell$

Figure 3.1: The inference system KB of standard completion.

and write $\gtrdot$ instead of $\gtrdot_{\mathsf{KB}}$ if the intended definition is clear from the context.

Lemma 3.7. *Every* KB *run* $(\mathcal{E}_0, \varnothing) \vdash (\mathcal{E}_1, \mathcal{R}_1) \vdash (\mathcal{E}_2, \mathcal{R}_2) \vdash \cdots$ *is an equational inference sequence with respect to* $\Rightarrow^{\succ}_{\mathsf{KB}}$.

Proof. We show that every step $(\mathcal{E}_i, \mathcal{R}_i) \vdash (\mathcal{E}_{i+1}, \mathcal{R}_{i+1})$ $i \geqslant 0$ can be modeled by expand and contract according to Definition 3.1. Note that we use the fact that $\mathcal{R}_i \subseteq \succ$.

- Every orient inference can be modeled by an expansion step adding $s \to t$ and a subsequent contraction step removing $s \simeq t$. These steps are valid because $s \leftrightarrow_{s \approx t} t$ has cost $(\{s,t\}, \ldots)$ but $s \leftrightarrow_{s \to t} t$ has cost $(\{s\}, \ldots)$.
- A deduce step is an instance of expand as $\mathrm{CP}(\mathcal{R}_i) \subseteq \leftrightarrow^*_{\mathcal{E} \cup \mathcal{R}_i}$.
- A delete step constitutes a contraction as clearly $s \leftrightarrow_{s \approx s} s \Rightarrow^{\succ}_{\mathsf{KB}} s$.
- An application of simplify can be viewed as an expansion inference adding $s \simeq u$ followed by a contraction step removing $s \simeq t$. The contraction is valid as the proof transformation $s \leftrightarrow_{s \approx t} t \Rightarrow^{\succ}_{\mathsf{KB}} s \leftrightarrow_{s \approx u} u \,{}^{p}_{\mathcal{R}_i}\!\leftarrow t$ is decreasing since $t \succ u$ implies $\{(\{s,t\}, \ldots)\} \gtrdot \{(\{s,u\}, \ldots), (\{t\}, \ldots)\}$. The expansion inference replaces $s \leftrightarrow_{s \approx t} t \to_{\mathcal{R}_i} u$ by $s \leftrightarrow_{s \approx u} u$. Decreasingness of this proof transformation is a consequence of the previous cost assessment.
- A compose step can be viewed as an expansion adding $s \to u$ followed by a contraction step removing $s \to t$. The latter is valid as the proof transformation $s \leftrightarrow^{\epsilon}_{s \to t} t \Rightarrow^{\succ}_{\mathsf{KB}} s \leftrightarrow^{\epsilon}_{s \to u} u \,{}_{\mathcal{R}_i}\!\leftarrow t$ results in the decrease $\{(\{s\}, s, s, t)\} \gtrdot \{(\{s\}, s, s, u), (\{t\}, \ldots)\}$ because of $s \succ t$ and $t \succ u$.

From this comparison it also follows that the expansion replacing $s \leftrightarrow^{\epsilon}_{s \to t} t \to_{\mathcal{R}_i} u$ by $s \leftrightarrow^{\epsilon}_{s \to u} u$ constitutes a decrease.

– Finally, a collapse step can be obtained by an expansion adding $u \approx s$ followed by a contraction step removing $t \to s$. By the definition of collapse there must be a rewrite step $t \to^{p}_{\ell \to r} u$ for some rule $\ell \to r$ and position $p \in \mathcal{P}\mathrm{os}(t)$ such that $t \rhd \ell$. The contraction amounts to a cost decrease since $t \leftrightarrow^{\epsilon}_{t \to s} s \Rightarrow^{\succ}_{\mathsf{KB}} t \leftrightarrow^{p}_{\ell \to r} u \leftrightarrow_{u \approx s} s$ because $t \succ u$, $t \succ s$, $t \trianglerighteq t|_p$ and $t \rhd \ell$ imply $\{(\{t\}, t, t, \ldots)\} \succcurlyeq \{(\{t\}, t|_p, \ell, \ldots), (\{u, s\}, \ldots)\}$. Consequently also the expansion performing the proof transformation $u \leftrightarrow_{\ell \leftarrow r} t \to_{t \to s} s \Rightarrow^{\succ}_{\mathsf{KB}} u \leftrightarrow_{u \approx s} s$ constitutes a cost reduction. □

We next recall the Critical Pair Lemma, a result originally due to Knuth and Bendix [57]. It shows that critical pairs deserve their name in that these peaks are the only critical overlaps as far as local confluence is concerned, and thus justifies restricting deduce steps to critical pairs.

Critical Pair Lemma 3.8. *Suppose a set of rewrite rules $\mathcal{R}$ admits a peak $s \,{}_{\mathcal{R}}\!\leftarrow u \to_{\mathcal{R}} t$. Then $s \downarrow_{\mathcal{R}} t$ unless $s = C[s'\sigma]$ and $t = C[t'\sigma]$ for some critical pair $s' \leftarrow \times \to t'$ in $\mathrm{CP}(\mathcal{R})$.*

Proof. Assume there is an overlap $\langle \ell_1 \to r_1, p, \ell_2 \to r_2 \rangle_\tau$ which gives rise to a peak $\ell_2\tau[r_1\tau]_p \leftarrow \ell_2\tau \to r_2\tau$ such that $s' = \ell_2\tau[r_1\tau]_p$, $t' = r_2\tau$ and $s = C[s'\sigma]$, $t = C[t'\sigma]$ for some context C and substitution σ (see also Figure 3.2(a)). Clearly, $s' \leftarrow \times \to t'$ is a critical pair in $\mathrm{CP}(\mathcal{R})$.

It remains to consider a peak $s \,{}^{p,\sigma}_{r_1 \leftarrow \ell_1}\!\leftarrow u \to^{q,\sigma}_{\ell_2 \to r_2} t$ which does not contain an overlap. Note that we assume the two rules to be variable-disjoint such that the same substitution σ is used in both rewrite steps. If $\ell_1 \to r_1$ and $\ell_2 \to r_2$ are variants and $p = q$ then $s = t$ and $s \downarrow t$ trivially holds.

Otherwise, if $p \parallel q$ the situation is as illustrated in Figure 3.2 (b). In this case we have $s \to^{q,\sigma}_{\ell_2 \to r_2} u[r_1\sigma]_p[r_2\sigma]_q \,{}^{p,\sigma}_{r_1 \leftarrow \ell_1}\!\leftarrow t$.

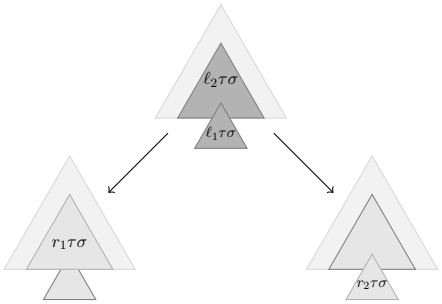
If $p \not\parallel q$ one position must be below the other. Without loss of generality we assume that $q = \epsilon$ and $p > \epsilon$, as illustrated in Figure 3.2 (c). So the peak under consideration is of the form $s \,{}^{p,\sigma}_{r_1 \leftarrow \ell_1}\!\leftarrow u \to^{\epsilon,\sigma}_{\ell_2 \to r_2} t$ and we have $u = \ell_2\sigma$, $s = u[r_1\sigma]_p = \ell_2\sigma[r_1\sigma]_p$ and $t = r_2\sigma$. As the peak does not constitute an overlap there must be some $q' \in \mathcal{P}\mathrm{os}_{\mathcal{V}}(\ell_2)$ such that $\ell_2|_{q'} = x$ and $p = q'p'$. We have $\sigma(x)|_{p'} = \ell_1\sigma$. Let the substitution σ' be defined as

$$\sigma'(y) = \begin{cases} \sigma(x)[r_1\sigma]_{p'} & \text{if } y = x \\ \sigma(y) & \text{otherwise} \end{cases}$$

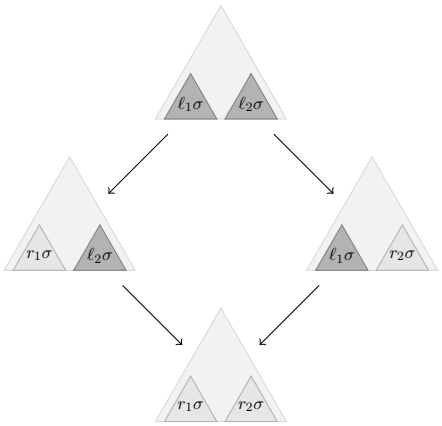
Then $\sigma(x) \to^* \sigma'(x)$ for all $x \in \mathcal{V}$ and therefore $r_2\sigma \to^* r_2\sigma'$. Moreover, we have $\ell_2\sigma[r_1\sigma]_p = \ell_2\sigma[\sigma'(x)]_{q'} \to^*_{\ell_1 \to r_1} \ell_2\sigma'$. This combines to

$$s = \ell_2\sigma[r_1\sigma]_p \to^* \ell_2\sigma' \to^{\epsilon}_{\ell_2 \to r_2} r_2\sigma' \,{}^*\!\leftarrow r_2\sigma = t$$

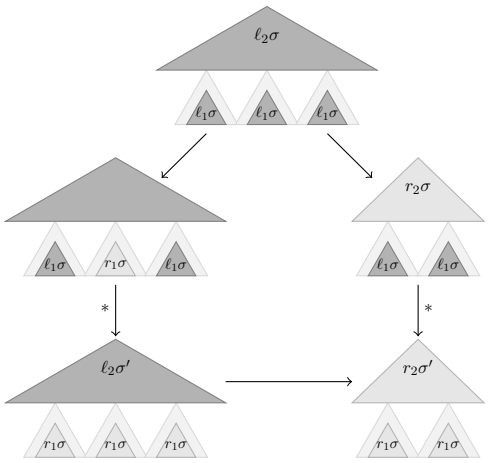
which proves $s \downarrow_{\mathcal{R}} t$. □



(a) proper overlap



(b) parallel redexes.



(c) variable overlap

Figure 3.2: Peak analysis.

Using the Critical Pair Lemma it is also easy to show that commutation of non-overlaps as well as critical pairs give rise to smaller proofs than the respective peaks. This result will be required in later proofs.

Corollary 3.9. *Let a set of rewrite rules $\mathcal{R}$ admit a peak $P\colon s \mathrel{{}_{\mathcal{R}}\!\leftarrow} u \rightarrow_{\mathcal{R}} t$.*

(a) If P constitutes a proper overlap then there is some $s' \leftarrow \times \rightarrow t' \in \mathrm{CP}(\mathcal{R})$ such that $P \Rightarrow^{\succ}_{\mathsf{KB}} s \leftrightarrow_{s'\approx t'} t$,

(b) otherwise there exists a rewrite proof Q in $\mathcal{R}$ such that $P \Rightarrow^{\succ}_{\mathsf{KB}} Q$.

Proof.

(a) By Lemma 3.8 we have $s = C[s'\sigma]$ and $t = C[t'\sigma]$ for some critical pair $s' \leftarrow \times \rightarrow t' \in \mathrm{CP}(\mathcal{R})$. The costs of the respective equational proofs are $c(P) = \{(\{u\}, \ldots), (\{u\}, \ldots)\}$ and $c(s \leftrightarrow_{s'\approx t'} t) = \{(\{s, t\}, \ldots)\}$ such that $P \Rightarrow^{\succ}_{\mathsf{KB}} s \leftrightarrow_{s'\approx t'} t$ because of $u \succ s, t$.

(b) According to Lemma 3.8 there is a rewrite proof $Q\colon s \rightarrow^*_{\mathcal{R}} \cdot \mathrel{{}^*_{\mathcal{R}}\!\leftarrow} t$. While $c(P) = \{(\{u\}, \ldots), (\{u\}, \ldots)\}$ the cost of Q involves only tuples $(\{v\}, \ldots)$ such that $s \succeq v$ or $t \succeq v$ holds, so $P \Rightarrow^{\succ}_{\mathsf{KB}} Q$. □

A KB run is intended to eventually admit rewrite proofs for all valid equations. Obviously, not even all nonfailing runs have this property, for instance if all equations get oriented but no critical pairs are computed. We thus use the notion of *fairness* to characterize runs that allow to simplify non-rewrite proofs:

Definition 3.10. A nonfailing KB run $(\mathcal{E}_0, \mathcal{R}_0) \vdash (\mathcal{E}_1, \mathcal{R}_1) \vdash (\mathcal{E}_2, \mathcal{R}_2) \vdash \cdots$ is *fair* if for any proof P in $\mathcal{R}_\omega$ which is not a rewrite proof there is a proof Q in some $(\mathcal{E}_i, \mathcal{R}_i)$ such that $P \Rightarrow^{\succ}_{\mathsf{KB}} Q$.

Note that this definition coincides with the notion of fairness in [6, 11] when restricted to nonfailing runs. In practice, fairness is often ensured by the following sufficient condition.

Lemma 3.11 ([11])**.** *If a nonfailing* KB *run satisfies $\mathrm{CP}(\mathcal{R}_\omega) \subseteq \bigcup_i \mathcal{E}_i$ then it is fair.*

Proof. Note that in a nonfailing run the set $\mathcal{E}_\omega$ is empty. We show that for every proof P in $\mathcal{R}_\omega$ which is not a rewrite proof there exists a proof Q in $\mathcal{R}_\omega$ such that $P \Rightarrow^{\succ}_{\mathsf{KB}} Q$. Since P is not a rewrite proof it must contain a peak $P'\colon s \leftarrow u \rightarrow t$. According to Corollary 3.9 there exists a smaller proof unless this peak constitutes a proper overlap. So assume $s = C[\ell\sigma]$ and $t = C[r\sigma]$ for some critical pair $\ell \leftarrow \times \rightarrow r \in \mathrm{CP}(\mathcal{R}_\omega)$, hence $P' \Rightarrow^{\succ}_{\mathsf{KB}} s \leftrightarrow^{\epsilon}_{\ell\approx r} t$ according to Corollary 3.9. By assumption $\ell \simeq r$ occurs in some set $\mathcal{E}_i$. By the Persistence Lemma 3.4 there is a proof Q' in $\mathcal{R}_\omega$ such that $s \leftrightarrow^{\epsilon}_{\ell\approx r} t$ $(\Rightarrow^{\succ}_{\mathsf{KB}})^{=}$ Q', and as $\Rightarrow^{\succ}_{\mathsf{KB}}$ is a proof reduction relation we have $P = P[P'] \Rightarrow^{\succ}_{\mathsf{KB}} P[Q']$. □

Correctness of KB runs can now be proven in the same fashion as in [6].

Correctness Theorem 3.12. *Any nonfailing* KB *run which uses a reduction order $\succ$ and is fair with respect to $\Rightarrow^{\succ}_{\mathsf{KB}}$ succeeds.*

Proof. Let $\gamma : (\mathcal{E}_0, \varnothing) \vdash (\mathcal{E}_1, \mathcal{R}_1) \vdash (\mathcal{E}_2, \mathcal{R}_2) \vdash \cdots$ be the run under consideration. In order to show convergence of $\mathcal{R}_\omega$, we show that for all terms s, t such that $s \leftrightarrow^*_{\mathcal{E}_0} t$ there is a persisting rewrite proof in the limit. By the Persistence Lemma 3.4 the TRS $\mathcal{R}_\omega$ admits a proof of any such equation $s \approx t$. Let P be a minimal such proof, and assume it is not a rewrite proof. By fairness, there exists a proof Q in $(\mathcal{E}_i, \mathcal{R}_i)$ for some $i \geqslant 0$ such that $P \Rightarrow^{\succ}_{\mathsf{KB}} Q$. By the Persistence Lemma 3.4, there is a proof Q' in $\mathcal{R}_\omega$ such that $Q \, (\Rightarrow^{\succ}_{\mathsf{KB}})^{=} \, Q'$, and hence $P \Rightarrow^{\succ}_{\mathsf{KB}} Q'$. This contradicts minimality of P, so P must be a rewrite proof. Since $\mathcal{R}_\omega$ has the same equational theory as $\mathcal{E}_0$ according to Lemma 3.2 it is convergent for $\mathcal{E}_0$. □

The following result states completeness in the sense that whenever a finite canonical TRS exists, a nonfailing fair KB run using a compatible reduction order succeeds in finite time [11].

Completeness Theorem 3.13. *Let $\mathcal{R}$ be a finite convergent TRS for $\mathcal{E}$ such that $\mathcal{R} \subseteq \succ$. Then any nonfailing fair* KB *run γ starting from $(\mathcal{E}, \varnothing)$ using $\succ$ derives a finite convergent TRS $\mathcal{R}'$ in finitely many steps.*

Proof. We denote by $\mathcal{Q}$ the TRS obtained from $\mathcal{R}$ when replacing each right-hand side by its $\mathcal{R}$-normal form. Obviously $\mathcal{Q}$ is terminating as it is contained in $\succ$, and it is also convergent: For every equation $s \approx t$ such that $s \leftrightarrow^*_{\mathcal{E}} t$ there is a rewrite proof of the form $s \to^*_{\mathcal{R}} u \, {}^*_{\mathcal{R}}\!\leftarrow t$. Let P be such a proof where u is in normal form with respect to $\mathcal{R}$. Now, as normal forms in $\mathcal{Q}$ and $\mathcal{R}$ coincide, there is also a rewrite proof $s \to^*_{\mathcal{Q}} u \, {}^*_{\mathcal{Q}}\!\leftarrow t$, so $\mathcal{Q}$ is convergent.

Let $\mathcal{R}_\omega$ be the set of persistent rules of γ. We next show that for every rule $\ell \to r$ in $\mathcal{Q}$ we have $\ell \to^*_{\mathcal{R}_\omega} r$. Since each rule $\ell \to r$ in $\mathcal{Q}$ belongs to the equational theory of $\mathcal{E}$ the run γ admits a persistent equational proof P of $\ell \approx r$ after a finite number of steps. As $\mathcal{R}_\omega$ is convergent by Theorem 3.12 this must be a rewrite proof. Note that r cannot be reducible in $\mathcal{R}_\omega$: suppose to the contrary there was a rewrite step $r \to_{\mathcal{R}_\omega} r'$. As $\mathcal{R}'$ and $\mathcal{Q}$ have the same equational theory, there must also be a proof of $r \simeq r'$ in $\mathcal{Q}$. But $r \to_{\mathcal{R}_\omega} r'$ implies $r \succ r'$, so r need also be reducible in $\mathcal{Q}$, which contradicts irreducibility of right-hand sides in $\mathcal{Q}$. Therefore P must have the form $\ell \to^*_{\mathcal{R}_\omega} r$.

Finally, this finite number of rewrite proofs $\ell \to^*_{\mathcal{R}_\omega} r$ derived in a finite number of steps requires only a finite subset of rules $\mathcal{R}' \subseteq \mathcal{R}_\omega$. Since $\to_{\mathcal{Q}} \subseteq \to^+_{\mathcal{R}'}$ also $\mathcal{R}'$ is convergent. □

It has been shown that if an equational theory allows for an equivalent canonical TRS compatible with some reduction order $\succ$ then it is unique modulo variable renaming [24,80]. We call a run *simplifying* if $\mathcal{R}_\omega$ is reduced. Thus the following result immediately follows from Theorem 3.13.

Corollary 3.14. *Let $\mathcal{R}$ be a finite canonical TRS for $\mathcal{E}$ such that $\mathcal{R} \subseteq \succ$. Then any nonfailing simplifying run γ starting from $\mathcal{E}$ using $\succ$ succeeds with a TRS $\mathcal{R}'$ in finitely many steps, and $\mathcal{R}$ coincides with $\mathcal{R}'$ modulo variable renaming.*

In the following example we derive two convergent systems for a given equational theory using different reduction orders.

Example 3.15. Consider the following set of equations:

$$\mathsf{f}(\mathsf{f}(x)) \approx \mathsf{g}(x) \tag{1}$$
$$\mathsf{f}(\mathsf{g}(\mathsf{f}(x))) \approx \mathsf{f}(x) \tag{2}$$

We consider an LPO with precedence $\mathsf{f} \succ \mathsf{g}$. In two orient steps we can thus direct both equations from left to right. This gives rise to the four critical overlaps $\langle (1), 1, (1) \rangle$, $\langle (1), 11, (2) \rangle$, $\langle (2), 1, (1) \rangle$, and $\langle (2), 11, (2) \rangle$. Respective deduce steps add the following equalities:

$$\mathsf{f}(\mathsf{g}(x)) \approx \mathsf{g}(\mathsf{f}(x)) \tag{3}$$
$$\mathsf{f}(\mathsf{g}(\mathsf{g}(x))) \approx \mathsf{f}(\mathsf{f}(x)) \tag{4}$$
$$\mathsf{f}(\mathsf{f}(x)) \approx \mathsf{g}(\mathsf{g}(\mathsf{f}(x))) \tag{5}$$
$$\mathsf{f}(\mathsf{g}(\mathsf{f}(x))) \approx \mathsf{f}(\mathsf{g}(\mathsf{f}(x))) \tag{6}$$

Equation (6) can be removed by delete. Next, orient may turn (3) into the rule

$$\mathsf{f}(\mathsf{g}(x)) \to \mathsf{g}(\mathsf{f}(x)) \tag{3}$$

Now a collapse step applying rule (3) to rule (2) removes the latter, instead adding an equation $\mathsf{g}(\mathsf{f}(\mathsf{f}(x))) \approx \mathsf{f}(x)$ which is in a simplify step using rule (1) further reduced to

$$\mathsf{g}(\mathsf{g}(x)) \approx \mathsf{f}(x) \tag{7}$$

Now, orient can be applied to this equation, which results in the rewrite rule

$$\mathsf{f}(x) \to \mathsf{g}(\mathsf{g}(x)) \tag{7}$$

With rule (7) both (4) and (5) are simplified into the same trivial equation $\mathsf{g}(\mathsf{g}(\mathsf{g}(\mathsf{g}(x)))) \approx \mathsf{g}(\mathsf{g}(\mathsf{g}(\mathsf{g}(x))))$, which is removed by delete. Another trivial equation $\mathsf{g}(\mathsf{g}(\mathsf{g}(x))) \approx \mathsf{g}(\mathsf{g}(\mathsf{g}(x)))$ is obtained when applying rule (7) in a compose and a collapse step to rule (3). Another collapse step using rule (7) turns (1) into the equation $\mathsf{g}(\mathsf{g}(\mathsf{g}(\mathsf{g}(x)))) \approx \mathsf{g}(x)$, which is subsequently oriented into the rewrite rule

$$\mathsf{g}(\mathsf{g}(\mathsf{g}(\mathsf{g}(x)))) \to \mathsf{g}(x) \tag{8}$$

This new rule (8) gives rise to three critical pairs such that deduce adds the equations

$$\mathsf{g}(\mathsf{g}(x)) \approx \mathsf{g}(\mathsf{g}(x))$$
$$\mathsf{g}(\mathsf{g}(\mathsf{g}(x))) \approx \mathsf{g}(\mathsf{g}(\mathsf{g}(x)))$$
$$\mathsf{g}(\mathsf{g}(\mathsf{g}(\mathsf{g}(x)))) \approx \mathsf{g}(\mathsf{g}(\mathsf{g}(\mathsf{g}(x))))$$

which can be deleted. Thus all critical pairs among the remaining rules (7) and (8) were considered, such that by Lemma 3.11 the run is fair. As no equations are left, it is also nonfailing. Thus by Theorem 3.12 the TRS consisting of the rules (7) and (8) is convergent. Note that it is also reduced.

A different canonical TRS is obtained when taking LPO with precedence $\mathsf{g} \succ \mathsf{f}$. Then equation (1) is reversed, such that we have

$$\mathsf{g}(x) \to \mathsf{f}(\mathsf{f}(x)) \tag{1}$$

and a simplify step can reduce equation (2) to

$$\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{f}(x)))) \approx \mathsf{f}(x) \tag{9}$$

which is subsequently oriented into the rule

$$\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{f}(x)))) \to \mathsf{f}(x) \tag{9}$$

Rule (9) gives rise to three trivial critical pairs, which are deleted. Thus all critical pairs among the remaining rules (1) and (9) were considered and no equations are left. It follows that the TRS consisting of the rules (1) and (9) is also convergent.

Note that the second run in the previous example is considerably shorter than the first one, which hints at the significant impact the reduction order has on the length of a run. But an unfortunate choice of the reduction order can also lead to failure or prevent a completion procedure from finding a finite convergent TRS, as illustrated by the following examples.

Example 3.16. Assume a KB run starts with the initial equations

$$\mathsf{f}(\mathsf{g}(x), \mathsf{h}(y)) \approx \mathsf{k}(x, y) \tag{1}$$
$$\mathsf{f}(\mathsf{h}(y), \mathsf{g}(x)) \approx \mathsf{k}(x, y) \tag{2}$$

Both equations can be oriented from left to right, for example with an LPO setting $\mathsf{f} \succ \mathsf{k}$. Then we immediately obtain a convergent TRS as there are no overlaps. On the other hand, if we choose as reduction order LPO with some precedence $\succ$ such that $\mathsf{k} \succ \mathsf{f}$, $\mathsf{k} \succ \mathsf{g}$, and $\mathsf{k} \succ \mathsf{h}$ then the two rules produce an unorientable equation $\mathsf{f}(\mathsf{g}(x), \mathsf{h}(y)) \approx \mathsf{f}(\mathsf{h}(y), \mathsf{g}(x))$, and the run fails.

Example 3.17. Consider the following set of equations:

$$\mathsf{f}(\mathsf{f}(\mathsf{g}(x))) \approx x \tag{1}$$
$$\mathsf{g}(\mathsf{g}(x)) \approx \mathsf{f}(\mathsf{g}(x)) \tag{2}$$

and choose as reduction order a KBO with $w_0 = w(\mathsf{f}) = w(\mathsf{g}) = 1$ and precedence $\mathsf{g} \succ \mathsf{f}$. We thus orient both equations from left to right:

$$\mathsf{f}(\mathsf{f}(\mathsf{g}(x))) \to x \tag{1}$$
$$\mathsf{g}(\mathsf{g}(x)) \to \mathsf{f}(\mathsf{g}(x)) \tag{2}$$

From the overlap $\langle (2), 11, (1) \rangle$ the critical pair $\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{g}(x)))) \leftarrow \rtimes \rightarrow \mathsf{g}(x)$ is created, which is oriented as

$$\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{g}(x)))) \to \mathsf{g}(x) \tag{3}$$

This in turn allows for a deduce step from the overlap $\langle (2), 111, (3) \rangle$ such that the equation $\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{g}(x))))) \approx \mathsf{g}(\mathsf{g}(x))$ is added and subsequently oriented as

$$\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{g}(x))))) \to \mathsf{g}(\mathsf{g}(x)) \tag{4}$$

When continuously performing respective deduce and orient steps with the new rules, the run will ultimately succeed in the limit with the infinite TRS consisting of rules

$$\mathsf{f}^{n+2}(\mathsf{g}(x)) \to \mathsf{g}^n(x) \qquad\qquad \mathsf{g}(\mathsf{g}(x)) \to \mathsf{f}(\mathsf{g}(x))$$

for all $n > 0$.

On the other hand, when using an LPO with precedence with $\mathsf{f} \succ \mathsf{g}$, equation (2) is oriented as

$$\mathsf{f}(\mathsf{g}(x)) \to \mathsf{g}(\mathsf{g}(x)) \tag{2}$$

such that equation (1) is in two steps simplified to

$$\mathsf{g}(\mathsf{g}(\mathsf{g}(x))) \approx x \tag{5}$$

When orienting (5), the resulting rule

$$\mathsf{g}(\mathsf{g}(\mathsf{g}(x))) \to x \tag{5}$$

gives rise to two trivial critical overlaps that are deleted. Moreover, the overlap $\langle (5), 1, (2) \rangle$ creates the equation $\mathsf{f}(x) \approx \mathsf{g}(\mathsf{g}(\mathsf{g}(\mathsf{g}(x))))$ which can be simplified using rule (5) to obtain $\mathsf{f}(x) \approx \mathsf{g}(x)$. This equation is oriented as

$$\mathsf{f}(x) \to \mathsf{g}(x) \tag{6}$$

such that (2) collapses into the trivial equation $\mathsf{g}(\mathsf{g}(x)) \approx \mathsf{g}(\mathsf{g}(x))$, which can be deleted. As there are no further critical overlaps to be considered and no equations are left the TRS consisting of rules (5) and (6) is convergent.

However, not only the reduction order has significant impact on success. Also the order in which equations are processed influences whether a run succeeds, as the following example shows.

Example 3.18 ([6,33])**.** Take LPO with precedence $\mathsf{a} \succ \mathsf{b} \succ \mathsf{c} \succ \mathsf{d}$ and consider the following set of equations:

$$\mathsf{a}(x, y) \approx \mathsf{b}(x) \tag{1}$$
$$\mathsf{a}(x, y) \approx \mathsf{c}(y) \tag{2}$$
$$\mathsf{f}(\mathsf{b}(x)) \approx \mathsf{b}(x) \tag{3}$$
$$\mathsf{f}(\mathsf{a}(x, y)) \approx \mathsf{d} \tag{4}$$

Assume a completion run starts by orienting (2):

$$\mathsf{a}(x, y) \to \mathsf{c}(y) \tag{2}$$

If simplification is applied eagerly then equations (1) and (4) are simplified to

$$\mathsf{c}(y) \approx \mathsf{b}(x) \tag{5}$$
$$\mathsf{f}(\mathsf{c}(y)) \approx \mathsf{d} \tag{6}$$

Then equations (3) and (6) can be oriented:

$$\mathsf{f}(\mathsf{b}(x)) \to \mathsf{b}(x) \tag{3}$$
$$\mathsf{f}(\mathsf{c}(y)) \to \mathsf{d} \tag{6}$$

The rules (2), (3), and (6) have no critical pairs such that no further step is possible. Hence the run fails as equation (5) remains unorientable. Note that after the decision to start with equation (2) we did not have a choice in this run (and no reduction order could have performed different orientations). Hence we conclude that any completion procedure which initially picks (2) and applies eager simplification necessarily fails.

Now assume a run uses the same reduction order but starts by orienting (1):

$$\mathsf{a}(x, y) \to \mathsf{b}(x) \tag{1}$$

This allows to simplify equations (2) and (4):

$$\mathsf{b}(x) \approx \mathsf{c}(y) \tag{5}$$
$$\mathsf{f}(\mathsf{b}(x)) \approx \mathsf{d} \tag{7}$$

Both (3) and (7) can be oriented:

$$\mathsf{f}(\mathsf{b}(x)) \to \mathsf{b}(x) \tag{3}$$
$$\mathsf{f}(\mathsf{b}(x)) \to \mathsf{d} \tag{7}$$

and these two rule give rise to a critical pair, which is oriented as

$$\mathsf{b}(x) \to \mathsf{d} \tag{8}$$

Now rule (8) can be applied to terms in (1), (3), (5), and (7). Orienting the resulting equations yields the following TRS $\mathcal{R}$:

$$\mathsf{a}(x, y) \to \mathsf{d} \qquad \mathsf{b}(x) \to \mathsf{d} \qquad \mathsf{c}(x) \to \mathsf{d} \qquad \mathsf{f}(\mathsf{d}) \to \mathsf{d}$$

As no equations are left and the TRS $\mathcal{R}$ has no critical pairs it is convergent.

The inference system KB might raise the question why the $\mathsf{collapse}$ rule prohibits rewrite steps where the left-hand side of the applied rule ℓ is a variant of the reduced term t. The following example shows how such an inference step can prevent success.

Example 3.19 ([11])**.** The following rewrite rules can be oriented by LPO with precedence $\mathsf{f} \succ \mathsf{c} \succ \mathsf{g} \succ \mathsf{b} \succ \mathsf{a}$ (cf. Example 2.8):

$$\mathsf{c} \to \mathsf{a} \tag{1}$$
$$\mathsf{g}(x) \to x \tag{2}$$
$$\mathsf{f}(\mathsf{b}, x) \to x \tag{3}$$
$$\mathsf{f}(\mathsf{g}(x), y) \to \mathsf{f}(x, \mathsf{g}(y)) \tag{4}$$
$$\mathsf{f}(x, \mathsf{b}) \to \mathsf{c} \tag{5}$$

collapse	$\dfrac{\mathcal{E}, \mathcal{R} \uplus \{t \to s\}}{\mathcal{E} \cup \{u \approx s\}, \mathcal{R}}$	if $t \to_{\mathcal{R}} u$

Figure 3.3: The collapse rule in KB'.

When orienting the critical pairs associated with $\langle(3), \epsilon, (5)\rangle$ and $\langle(4), \epsilon, (5)\rangle$

$$\mathsf{c} \to \mathsf{b} \tag{6}$$
$$\mathsf{f}(x, \mathsf{g}(\mathsf{b})) \to \mathsf{c} \tag{7}$$

are added. Without inter-reduction, these rules create the overlaps $\langle(3), \epsilon, (7)\rangle$ and $\langle(4), \epsilon, (7)\rangle$, resulting in critical pairs that are oriented as

$$\mathsf{c} \to \mathsf{g}(\mathsf{b}) \tag{8}$$
$$\mathsf{f}(x, \mathsf{g}(\mathsf{g}(\mathsf{b}))) \to \mathsf{c} \tag{9}$$

With a relaxed collapse rule, (8) can be used to simplify (6) to $\mathsf{g}(\mathsf{b}) \to \mathsf{b}$, which is simplified away by (2). Thus, as (6) is not persistent the critical pair between (1) and (6) need not be considered. But in the next step additional rules

$$\mathsf{c} \to \mathsf{g}(\mathsf{g}(\mathsf{b})) \tag{10}$$
$$\mathsf{f}(x, \mathsf{g}(\mathsf{g}(\mathsf{g}(\mathsf{b})))) \to \mathsf{c} \tag{11}$$

are created, such that (10) can in turn be used to collapse (8). Repeating respective deduce steps yields the infinite TRS $\mathcal{R}_\omega$ consisting of the rules

$$\mathsf{c} \to \mathsf{a} \quad \mathsf{g}(x) \to x \quad \mathsf{f}(\mathsf{b}, x) \to x \quad \mathsf{f}(\mathsf{g}(x), y) \to \mathsf{f}(x, \mathsf{g}(y)) \quad \mathsf{f}(x, \mathsf{g}^n(\mathsf{b})) \to \mathsf{c}$$

for all $n \geqslant 0$. But $\mathcal{R}_\omega$ is not confluent as a and b are not joinable, despite the peak $\mathsf{b} \leftarrow \mathsf{f}(\mathsf{b}, \mathsf{b}) \to \mathsf{c} \to \mathsf{a}$.

3.2.1 Finite Runs

Example 3.19 shows that the side condition of the rewrite step applied in a collapse inference is essential for correctness. However, it was shown that for *finite* runs arbitrary rewrite steps in collapse inferences are sound [97].

We define the inference system KB' to consist of the relaxed collapse rule in Figure 3.3 together with all remaining rules from KB. In order to prove correctness of this relaxed completion procedure a modified definition of the corresponding proof order is required. In this new definition we need to uniquely identify rewrite rules, i.e., we need to distinguish between (variants of) the same rewrite rule developed in the course of a run. For this purpose we assume that all equations and rewrite rules occurring during a run are variable-disjoint, i.e., whenever a new equation or rule is created it contains only fresh variables. We will in the sequel of Section 3.2.1 restrict to *finite* KB' runs of some length $n \in \mathbb{N}$:

$$(\mathcal{E}_0, \varnothing) \vdash (\mathcal{E}_1, \mathcal{R}_1) \vdash (\mathcal{E}_2, \mathcal{R}_2) \vdash \cdots \vdash (\mathcal{E}_n, \mathcal{R}_n) \tag{3.1}$$

A run *fails* if $\mathcal{E}_n \neq \varnothing$, and it *succeeds* if $\mathcal{E}_n = \varnothing$ and $\mathcal{R}_n$ is convergent for $\mathcal{E}_0$.

We define a modified proof order $\Rightarrow_{\mathsf{KB}}^{\succ,n}$. In contrast to the proof orders defined earlier, it depends on the actual length n of the run.

Definition 3.20. Consider a run of the form (3.1) which has length $n \in \mathbb{N}$ and uses the reduction order $\succ$. The *cost* c_n of an equational proof step in $\bigcup_i \mathcal{E}_i \cup \mathcal{R}_i$ is defined as follows:

$$c_n(s \underset{\ell \to r}{\overset{p}{\longleftrightarrow}} t) = c_n(t \underset{r \leftarrow \ell}{\overset{p}{\longleftrightarrow}} s) = (\{s\}, s|_p, n-k) \quad \text{where } k \text{ is maximal such that } \ell \to r \in \mathcal{R}_k$$

$$c_n(s \underset{u \approx v}{\longleftrightarrow} t) = (\{s,t\}, \bot, 0)$$

To compare cost tuples we use the lexicographic combination of $\succ_{\mathsf{mul}}$, $\rhd$, and the standard order $>$ on $\mathbb{N}$, where $\bot$ is considered minimal in $\rhd$. Again the cost of an equational proof is the multiset of its steps' costs, the proof order $\succnsim_{\mathsf{KB}}^{n}$ is the multiset extension of the order on proof step costs, and $P \Rightarrow_{\mathsf{KB}}^{\succ,n} Q$ holds if and only if $P \succnsim_{\mathsf{KB}}^{n} Q$ and P and Q prove the same equation.

As a lexicographic combination of well-founded orders $\succnsim$ is terminating, the following result is easily established.

Lemma 3.21. *The relation $\Rightarrow_{\mathsf{KB}}^{\succ,n}$ is a proof reduction relation.* □

Like KB also KB' yields a decrease with respect to $\Rightarrow_{\mathsf{KB}}^{\succ,n}$ in every inference step (where the theory $\mathcal{T}$ from Definition 3.1 is again empty). We will write $\succnsim$ instead of $\succnsim_{\mathsf{KB}}^{n}$ if the intended definition is clear from the context.

Lemma 3.22. *Every KB' run of the form* (3.1) *is an equational inference sequence with respect to $\Rightarrow_{\mathsf{KB}}^{\succ,n}$.*

Proof. We show that every KB' step $(\mathcal{E}_i, \mathcal{R}_i) \vdash_{\mathsf{KB}'} (\mathcal{E}_{i+1}, \mathcal{R}_{i+1})$ for $1 \leqslant i < n$ can be modeled by expand and contract steps according to Definition 3.1. Note that the third and fourth component of the cost given in Definition 3.5 are only needed for compose and collapse steps in Lemma 3.7. Except for these cases we can thus argue as in this earlier proof.

A compose step can be viewed as an expansion inference adding $s \to u$ followed by a contraction step removing $s \to t$. Note that by the assumption on variable-disjoint rules i must be maximal such that $\mathcal{R}_i$ contains $s \to t$, while for the maximal number j such that $s \to u \in \mathcal{R}_j$ we have $j > i$. The cost of the step $s \leftrightarrow_{s \to t}^{\epsilon} t$ is thus $(\{s\}, s, n-i)$ while the cost of $s \leftrightarrow_{s \to u}^{\epsilon} u \leftarrow t$ is $\{(\{s\}, s, n-j), (\{t\}, \ldots)\}$. So because of $s \succ t$ and $n - i > n - j$ we have $s \leftrightarrow_{s \to t}^{\epsilon} t \Rightarrow_{\mathsf{KB}}^{\succ,n} s \leftrightarrow_{s \to u}^{\epsilon} u \leftarrow t$. The expansion replaces $s \leftrightarrow_{s \to t}^{\epsilon} t \to u$ by $s \leftrightarrow_{s \to u}^{\epsilon} u$, which results in the decrease $\{(\{s\}, s, n-i), (\{t\}, \ldots)\} \succnsim \{(\{s\}, s, n-j)\}$.

Also a collapse step is obtained by an expansion inference adding $u \approx s$ and a subsequent contraction step which removes $t \to s$ because $t \to_{\ell \to r}^{p} u$. By the assumption on variable-disjoint rules i must be maximal such that $\mathcal{R}_i$ contains $t \to s$, while for the maximal number j such that $\ell \to r \in \mathcal{R}_j$ we have $j > i$. The cost of the step $t \leftrightarrow_{t \to s}^{\epsilon} s$ is therefore $(\{t\}, t, n-i)$ while the cost of $t \leftrightarrow_{\ell \to r}^{p} u \leftrightarrow_{u \approx s} s$ is $\{(\{t\}, t|_p, n-j), (\{u, s\}, \ldots)\}$. Since $t \succ u, s$, $t \trianglerighteq t|_p$, and

$n-i > n-j$ we have $t \leftrightarrow^{\epsilon}_{t\to s} s \Rightarrow^{\succ,n}_{\mathsf{KB}} t \leftrightarrow^{p}_{\ell\to r} u \leftrightarrow_{u\approx s} s$. The expansion replaces $u \leftrightarrow^{p}_{r\leftarrow\ell} t \to^{\epsilon}_{t\to s} s$ by $u \leftrightarrow_{u\approx s} s$ and as $\{(\{t\},\ldots),(\{t\},\ldots)\} \succ\!\!\succ \{(\{u,s\},\ldots)\}$ also this step is decreasing. □

Due to this result, the Persistence Lemma 3.4 holds for KB'. Fairness of nonfailing KB' runs is defined exactly as in Definition 3.10, except that the modified proof order $\Rightarrow^{\succ,n}_{\mathsf{KB}}$ is used. We can then prove the following results on fairness, correctness, and completeness in exactly the same way as for KB.

Lemma 3.23. *If a nonfailing and finite* KB' *run* $(\mathcal{E},\varnothing) \vdash^n (\varnothing,\mathcal{R}_n)$ *satisfies* $\mathrm{CP}(\mathcal{R}_n) \subseteq \bigcup_i \mathcal{E}_i$ *then it is fair with respect to* $\Rightarrow^{\succ,n}_{\mathsf{KB}}$. □

Correctness Theorem 3.24. *A nonfailing, finite* KB' *run* $(\mathcal{E},\varnothing) \vdash^n (\varnothing,\mathcal{R}_n)$ *which uses a reduction order* $\succ$ *and is fair with respect to* $\Rightarrow^{\succ,n}_{\mathsf{KB}}$ *succeeds.* □

Completeness Theorem 3.25. *Let* $\mathcal{R}$ *be a finite convergent TRS for* $\mathcal{E}$ *such that* $\mathcal{R} \subseteq \succ$. *Then any nonfailing fair* KB' *run* γ *which starts from* $\mathcal{E}$ *and uses* $\succ$ *succeeds with a TRS* $\mathcal{R}'$ *in finitely many steps.* □

3.2.2 Critical Pair Criteria

In order to limit the number of equations deduced in a completion run, several *critical pair criteria* were proposed as a means to filter out critical pairs that can be ignored without compromising completeness [9, 51, 61, 109]. In a later work, Bachmair and Dershowitz [11] showed that all these criteria match the more general criterion of *compositeness*.

Let $\mathcal{E}$ be a set of equations and $\mathcal{R}$ be a set of rewrite rules. Formally, a critical pair criterion CPC maps $(\mathcal{E},\mathcal{R})$ to a subset $\mathrm{CPC}(\mathcal{E},\mathcal{R})$ of $\mathrm{CP}(\mathcal{R})$. Intuitively, $\mathrm{CPC}(\mathcal{E},\mathcal{R})$ contains those critical pairs that are considered redundant.

Definition 3.26. A KB run $(\mathcal{E}_0,\varnothing) \vdash (\mathcal{E}_1,\mathcal{R}_1) \vdash (\mathcal{E}_2,\mathcal{R}_2) \vdash \cdots$ is *fair with respect to a criterion* CPC *and a proof order* $\Rightarrow$ if for every critical overlap P associated with a critical pair $s \leftarrow \rtimes \rightarrow t \in \mathrm{CP}(\mathcal{R}_\omega) \setminus \bigcup_i \mathrm{CPC}(\mathcal{E}_i,\mathcal{R}_i)$ there exists a proof Q in $\bigcup_i \mathcal{E}_i \cup \mathcal{R}_i$ such that $P \Rightarrow Q$. We say that a criterion is *correct* for $\Rightarrow$ if a nonfailing run is fair whenever it is fair with respect to CPC and $\Rightarrow$.

Obviously, correct critical pair criteria do not harm fairness. We will focus on the following abstract criterion of compositeness.

Definition 3.27. Let $\mathcal{E}$ be a set of equations, $\mathcal{R}$ a set of rewrite rules, and $\succ\!\!\succ$ a proof order using reduction order $\succ$. An equational proof P that has the form of a peak $s \leftarrow u \rightarrow t$ is *composite* with respect to $(\mathcal{E},\mathcal{R})$ and a proof order $\succ\!\!\succ$ if there exist terms $u_0,\ldots,u_{n+1}$ such that $s = u_0$, $t = u_{n+1}$, and $u \succ u_i$ for all $1 \leqslant i \leqslant n$, together with proofs $P_0,\ldots,P_n$ in $(\mathcal{E},\mathcal{R})$ such that P_i proves $u_i \approx u_{i+1}$ and $P \succ\!\!\succ P_i$ holds for all $1 \leqslant i \leqslant n$.

For a fixed proof order $\succ\!\!\succ$, the *compositeness criterion* $\mathrm{CCP}(\mathcal{E},\mathcal{R})$ returns all critical pairs between rules in $\mathcal{R}$ for which the associated overlaps are composite with respect to $(\mathcal{E},\mathcal{R})$ and $\succ\!\!\succ$. We can now relax the result of Lemma 3.11 and show that ignoring composite critical pairs does not compromise fairness.

Lemma 3.28. *Consider a nonfailing* KB *run* $\gamma\colon (\mathcal{E}_0, \varnothing) \vdash (\mathcal{E}_1, \mathcal{R}_1) \vdash (\mathcal{E}_2, \mathcal{R}_2) \vdash \cdots$ *and let* $\mathcal{C}$ *be a subset of* $\bigcup_i \mathrm{CCP}(\mathcal{E}_i, \mathcal{R}_i)$*, where* CCP *is computed with respect to* $\succ\!\!\succ_{\mathsf{KB}}$*. If* $\mathrm{CP}(\mathcal{R}_\omega) \setminus \mathcal{C} \subseteq \bigcup_i \mathcal{E}_i$ *then* γ *is fair.*

Proof. Just like in the proof of Lemma 3.11 we show that any non-rewrite proof in $\mathcal{R}_\omega$ can be transformed into a rewrite proof.

We apply induction on $\succ\!\!\succ_{\mathsf{KB}}$. Any non-rewrite proof must contain a peak $P\colon s \leftarrow u \rightarrow t$. If this peak does not constitute a proper overlap then according to Corollary 3.9 there exists a proof P' such that $P \Rightarrow^{\succ}_{\mathsf{KB}} P'$, and by induction hypothesis P' can be transformed into a rewrite proof. Otherwise, $s = C[\ell\sigma]$ and $t = C[r\sigma]$ for some critical pair $\ell \leftarrow\!\times\!\rightarrow r$ in $\mathrm{CP}(\mathcal{R}_\omega)$. We have $P \Rightarrow^{\succ}_{\mathsf{KB}} s \leftrightarrow^{\epsilon}_{\ell\approx r} t$. If $\ell \simeq r$ occurs in some set $\mathcal{E}_i$ then by the Persistence Lemma 3.4 there is a proof P' in $\mathcal{R}_\omega$ such that $s \leftrightarrow^{\epsilon}_{\ell\approx r} t\ (\Rightarrow^{\succ}_{\mathsf{KB}})^{=} P'$. By the induction hypothesis there is a rewrite proof for P' in $\mathcal{R}_\omega$, and thus also for P. If $\ell \simeq r$ does not occur in any set $\mathcal{E}_i$ then we must have $\ell \leftarrow\!\times\!\rightarrow r \in \mathrm{CCP}(\mathcal{E}_i, \mathcal{R}_i)$ for some i. Let the corresponding overlap be $P'\colon \ell \leftarrow v \rightarrow r$. By definition, there are terms $v_1, \ldots, v_{n+1}$ such that $\ell = v_0$, $r = v_{n+1}$ and $v \succ v_i$, and $(\mathcal{E}_i, \mathcal{R}_i)$ admits proofs P_i of $v_i \approx v_{i+1}$ which are smaller than P'. Thus the proofs $C[P_i\sigma]$ prove $C[v_i\sigma] \approx C[v_{i+1}\sigma]$ for all $1 \leqslant i \leqslant n$, and by the Persistence Lemma 3.4 there are respective proofs P_i' in $\mathcal{R}_\omega$ such that $C[P_i\sigma]\ (\Rightarrow^{\succ}_{\mathsf{KB}})^{=} P_i'$. By the induction hypothesis all these proofs P_i' can be transformed into rewrite proofs Q_i in $\mathcal{R}_\omega$. Consequently all terms in the combined proof $Q_1 \cdots Q_n$ must be smaller than $u = C[v\sigma]$, and therefore we have $P \Rightarrow^{\succ}_{\mathsf{KB}} Q_1 \cdots Q_n$. By the induction hypothesis $Q_1 \cdots Q_n$ can be transformed into a rewrite proof, and thus this also holds for P. □

A completely analogous proof can be used to show a corresponding result for KB', thereby relaxing Lemma 3.23.

Lemma 3.29. *The compositeness criterion using* $\succ\!\!\succ^{n}_{\mathsf{KB}}$ *is correct for* $\Rightarrow^{\succ,n}_{\mathsf{KB}}$. □

Note that compositeness is not computable, so this general criterion cannot be applied in practice. However, criteria to filter out superfluous critical pairs in completion procedures that were previously proposed in the literature actually turned out to be special cases of compositeness.

Primality

Kapur *et al.* [51] introduced the notion of primality for critical pairs. An overlap $\langle \ell_1 \rightarrow r_1, p, \ell_2 \rightarrow r_2 \rangle_\sigma$ is *prime* if $\ell_2\sigma$ is not reducible at any position strictly below p. The primality criterion $\mathrm{PCP}(\mathcal{E}, \mathcal{R})$ returns all critical pairs among rules in $\mathcal{R}$ for which the associated overlaps are not prime.

Lemma 3.30. *Every non-prime critical pair is composite with respect to* $\succ\!\!\succ_{\mathsf{KB}}$ *and* $\succ\!\!\succ^{n}_{\mathsf{KB}}$.

Proof. Let $\mathcal{R}$ be a TRS such that $\ell_1 \rightarrow r_1, \ell_2 \rightarrow r_2 \in \mathcal{R}$ admit an overlap $\langle \ell_1 \rightarrow r_1, p, \ell_2 \rightarrow r_2 \rangle$ which corresponds to the peak $P\colon s \;{}_{r_1 \leftarrow \ell_1}\!\!\overset{p}{\leftarrow} u \rightarrow^{\epsilon}_{\ell_2 \rightarrow r_2} t$. Suppose the critical pair $s \leftarrow\!\rtimes\!\rightarrow t$ is not prime since $u \rightarrow^{q}_{\mathcal{R}} v$ for some

position $q > p$. We obviously have $u \succ s, t, v$ as $\mathcal{R} \subseteq \succ$. Let proofs P_1 and P_2 be defined by $P_1 \colon s \;{}_{r_1 \leftarrow \ell_1}^{p}\!\leftarrow u \to_{\mathcal{R}}^{q} v$ and $P_2 \colon v \;{}_{\mathcal{R}}^{q}\!\leftarrow u \to_{\ell_2 \to r_2}^{\epsilon} t$. Note that $P_1 P_2$ proves $s \approx t$. For the cost measures underlying both $\succnsim_{\mathsf{KB}}$ and $\succnsim_{\mathsf{KB}}^{n}$ we obtain costs of the form $c(P) = \{(\{u\}, u|_p, \ldots), (\{u\}, u, \ldots)\}$, $c(P_1) = \{(\{u\}, u|_p, \ldots), (\{u\}, u|_q, \ldots)\}$, and $c(P_2) = \{(\{u\}, u|_q, \ldots), (\{u\}, u, \ldots)\}$. Since $u \trianglerighteq u|_p \trianglerighteq u|_q$ both $P \succnsim P_1$ and $P \succnsim P_2$ hold, independent of whether $\succnsim$ is $\succnsim_{\mathsf{KB}}$ or $\succnsim_{\mathsf{KB}}^{n}$. Hence P is composite. □

A special case of PCP is captured by the *unblockedness criterion* BCP [9]. A critical pair stemming from a peak $\ell_2\sigma[r_1\sigma]_p \;{}_{r_1 \leftarrow \ell_1}\!\leftarrow \ell_2\sigma[\ell_1\sigma]_p = \ell_2\sigma \to_{\ell_2 \to r_2} r_2\sigma$ is *blocked* if $x\sigma$ is irreducible in $\mathcal{R}$ for all variables $x \in \mathcal{V}\mathrm{ar}(\ell_1) \cup \mathcal{V}\mathrm{ar}(\ell_2)$. The set $\mathrm{BCP}(\mathcal{E}, \mathcal{R})$ contains all unblocked critical pairs among rules in $\mathcal{R}$.

Connectedness

Küchlin [61] introduced the notion of *connectedness* to limit equational consequences deduced in a completion procedure. Given a set of equations $\mathcal{E}$ and a set of rewrite rules $\mathcal{R}$, a critical pair $s \leftarrow\!\!\rtimes \to t$ originating from an overlap $s \leftarrow u \to t$ is *connected below* u if there exists a proof of the form $s = u_0 \leftrightarrow u_1 \leftrightarrow \cdots \leftrightarrow u_{n+1} = t$ in $(\mathcal{E}, \mathcal{R})$ such that $u \succ u_i$ for all $1 \leqslant i \leqslant n$.

Lemma 3.31. *If a critical pair $s \leftarrow\!\!\times \to t$ originating from a peak $s \leftarrow u \to t$ is connected below u then it is composite with respect to $\succnsim_{\mathsf{KB}}$ and $\succnsim_{\mathsf{KB}}^{n}$.*

Proof. Let P denote the peak $s \leftarrow u \to t$ and P_i the single-step proof $u_i \leftrightarrow u_{i+1}$ in $(\mathcal{E}, \mathcal{R})$ for all $0 \leqslant i \leqslant n$. For both proof orders under consideration $c(P) = \{(\{u\}, \ldots), (\{u\}, \ldots)\}$ while $c(P_i)$ contains only a single tuple $(\{u_i\}, \ldots)$, $(\{u_{i+1}\}, \ldots)$, or $(\{u_i, u_{i+1}\}, \ldots)$ such that $u \succ u_i, u_{i+1}$. Therefore $P \succnsim P_i$ for all $0 \leqslant i \leqslant n$. As $P_0 \cdots P_n$ proves $s \approx t$ the critical pair is composite. □

In practice, this still very general connectedness criterion needs to be approximated. Küchlin concentrates on finding terms $u_1, \ldots, u_n$ such that $u \to^+ u_i$. As a special case the following *weak connectivity test* is proposed. Given an overlap $\langle \ell_1 \to r_1, p, \ell_2 \to r_2 \rangle$ which corresponds to the peak $P \colon s \;{}_{r_1 \leftarrow \ell_1}^{p}\!\leftarrow u \to_{\ell_2 \to r_2}^{\epsilon} t$ between rules $\ell_1 \to r_1$ and $\ell_2 \to r_2$, the associated critical pair is *weakly connected* if there exists a reduction step $u \to_{\ell_3 \to r_3}^{q} v$ such that the following (non-exclusive) properties are satisfied:

(i) if $q \in \mathcal{P}\mathrm{os}_{\mathcal{F}}(\ell_2)$ then the overlap $\langle \ell_3 \to r_3, q, \ell_2 \to r_2 \rangle$ was already considered,

(ii) if $q = pq'$ and $q' \in \mathcal{P}\mathrm{os}_{\mathcal{F}}(\ell_1)$ then $\langle \ell_3 \to r_3, q', \ell_1 \to r_1 \rangle$ was already considered, and

(iii) if $p = qp'$ and $p' \in \mathcal{P}\mathrm{os}_{\mathcal{F}}(\ell_3)$ then $\langle \ell_2 \to r_2, p', \ell_3 \to r_3 \rangle$ was already considered.

It is easy to see that all weakly connected critical pairs are indeed connected and thus composite. This criterion can also be generalized to a full connectivity test

where the critical pair is connected via an arbitrary sequence $v_1, \ldots, v_n$ instead of a single intermediate term v.

In the sequel the connectedness criterion returning weakly connected critical pairs among rules in $\mathcal{R}$ will be referred to as $\mathrm{WCP}(\mathcal{E}, \mathcal{R})$.

Since both WCP and PCP are special cases of compositeness these criteria can also be combined. This *mixed* criterion filters out critical pairs that are redundant according to one of the criteria, it will in the sequel be referred to as MCP.

Example 3.32. A KB (or KB′) run on group theory may encounter the following rules:

$$\mathsf{e} \cdot x \to x \tag{1}$$
$$\mathsf{i}(x) \cdot x \to \mathsf{e} \tag{2}$$
$$x \cdot \mathsf{e} \to x \tag{3}$$
$$\mathsf{i}(\mathsf{e}) \to \mathsf{e} \tag{4}$$
$$\mathsf{i}(\mathsf{i}(x)) \to x \tag{5}$$
$$(y \cdot \mathsf{i}(x)) \cdot x \to y \tag{6}$$

Consider the overlap $\langle (2), 1, (6) \rangle$ corresponding to the peak $\mathsf{e} \cdot x \,{}^{1}\!\leftarrow (\mathsf{i}(\mathsf{i}(x)) \cdot \mathsf{i}(x)) \cdot x \to^{\epsilon} \mathsf{i}(\mathsf{i}(x))$. Note that $(\mathsf{i}(\mathsf{i}(x)) \cdot \mathsf{i}(x)) \cdot x \to^{11}_{(5)} (x \cdot \mathsf{i}(x)) \cdot x$, and position 11 is a variable position in the left-hand side of rule (6). Hence the critical pair is not blocked (and consequently also not prime), so it can be ignored according to the criteria BCP and PCP.

Consider the overlap $\langle (2), \epsilon, (3) \rangle$ corresponding to the peak $\mathsf{e} \,{}^{\epsilon}\!\leftarrow \mathsf{i}(\mathsf{e}) \cdot \mathsf{e} \to^{\epsilon} \mathsf{i}(\mathsf{e})$. Since $\mathsf{i}(\mathsf{e}) \cdot \mathsf{e} \to^{1}_{(4)} \mathsf{e} \cdot \mathsf{e}$ the corresponding critical pair is not prime (but nevertheless blocked). Thus it can be ignored according to the criterion PCP.

The overlap $\langle (4), 1, (2) \rangle$ corresponds to the peak $P \colon \mathsf{e} \cdot \mathsf{e} \,{}^{1}\!\leftarrow \mathsf{i}(\mathsf{e}) \cdot \mathsf{e} \to^{\epsilon} \mathsf{e}$. We also have $\mathsf{i}(\mathsf{e}) \cdot \mathsf{e} \to^{\epsilon}_{(3)} \mathsf{i}(\mathsf{e})$, but PCP is not applicable. When applying the weak connectivity test the peak P gets decomposed as follows:

$$\mathsf{e} \cdot \mathsf{e} \;{}^{1}_{(4)}\!\leftarrow \mathsf{i}(\mathsf{e}) \cdot \mathsf{e} \to^{\epsilon}_{(3)} \mathsf{i}(\mathsf{e}) \qquad \text{and} \qquad \mathsf{i}(\mathsf{e}) \;{}^{\epsilon}_{(3)}\!\leftarrow \mathsf{i}(\mathsf{e}) \cdot \mathsf{e} \to^{\epsilon}_{(2)} \mathsf{e}$$

Indeed the first peak is only a variable overlap and the critical pair emerging from the second peak was already considered. Therefore the overlap is weakly connected and can be ignored according to WCP.

Chapter 4

Multi-Completion with Termination Tools

Classical Knuth-Bendix completion requires a reduction order as input. As illustrated by several examples in Chapter 3, this parameter is critical for success but an appropriate choice is hard to determine in advance. Bofill *et al.* [20] described a non-failing completion procedure which can even be applied with a non-total and non-monotonic reduction order. But since their method relies on enumerating all equational consequences it is of rather theoretical nature and hardly applicable in practice.

Kondo and Kurihara [62] challenged the limitation of a single ordering in a different way by proposing completion with multiple reduction orders. In this method several classical runs are simulated in parallel, but common inference steps are shared to gain efficiency. For the sake of brevity we will in the sequel refer to this idea by *multi-completion*. Completion with termination tools [107] constitutes another approach to tackle the challenge of picking the right ordering: Wehrman, Stump, and Westbrook proposed to employ modern termination tools for the orientation of equations that are generated in a completion run. This not only relieves users from a tricky choice. Automatic termination analysis also emerged as a highly active research area over the past years. This gave rise to a wide variety of sophisticated techniques that can be employed to establish termination of a TRS. Hence completion procedures relying on termination tools exhibit a considerable gain in power—they can complete far more systems than those compatible with standard reduction orders like LPO or KBO.

In this chapter we will recall multi-completion in Section 4.1 and completion with termination tools in Section 4.2 before we present a unified approach in Section 4.3.

4.1 Multi-Completion

We consider a set of reduction order $\mathcal{O} = \{\succ_1, \ldots, \succ_n\}$. Multi-completion simulates the parallel execution of completion runs for all of these orderings but shares common inference steps. The key idea to sharing is a data structure called *node*.

Definition 4.1. A *node* is a tuple $\langle s : t, R_0, R_1, E\rangle$ where the *data* $s : t$ consist of terms s, t and the *labels* R_0, R_1, E are subsets of $\mathcal{O}$. The *node condition* requires that R_0, R_1 and E are mutually disjoint, $s \succ_i t$ holds for all $\succ_i \in R_0$, and $t \succ_i s$ for all $\succ_i \in R_1$.

orient	$\dfrac{N \uplus \{\langle s : t, R_0, R_1, E \uplus R\rangle\}}{N \cup \{\langle s : t, R_0 \cup R, R_1, E\rangle\}}$
	if $s \succ_i t$ for all $\succ_i \in R$ and $R \neq \varnothing$
deduce	$\dfrac{N}{N \cup \{\langle s : t, \varnothing, \varnothing, R \cap R'\rangle\}}$
	if N contains $\langle \ell_1 : r_1, R, \ldots\rangle$ and $\langle \ell_2 : r_2, R', \ldots\rangle$, $R \cap R' \neq \varnothing$ and $s \leftarrow \rtimes \rightarrow t \in \mathrm{CP}(\ell_1 \rightarrow r_1, \ell_2 \rightarrow r_2)$
delete	$\dfrac{N \uplus \{\langle s : s, \varnothing, \varnothing, E\rangle\}}{N}$
	if $E \neq \varnothing$
$\mathsf{rewrite}_1$	$\dfrac{N \uplus \{\langle s : t, R_0, R_1, E\rangle\}}{N \cup \{\langle s : t, R_0 \setminus R, R_1, E \setminus R\rangle,\ \langle s : u, R_0 \cap R, \varnothing, E \cap R\rangle\}}$
	if $\langle \ell : r, R, \ldots\rangle \in N$, $t \rightarrow_{\ell \rightarrow r} u$ such that $t \doteq \ell$, and $(R_0 \cup E) \cap R \neq \varnothing$
$\mathsf{rewrite}_2$	$\dfrac{N \uplus \{\langle s : t, R_0, R_1, E\rangle\}}{N \cup \{\langle s : t, R_0 \setminus R, R_1 \setminus R, E \setminus R\rangle,\ \langle s : u, R_0 \cap R, \varnothing, (E \cup R_1) \cap R\rangle\}}$
	if $\langle \ell : r, R, \ldots\rangle \in N$, $t \rightarrow_{\ell \rightarrow r} u$ such that $t \rhd \ell$, and $(R_0 \cup R_1 \cup E) \cap R \neq \varnothing$
gc	$\dfrac{N \uplus \{\langle s : t, \varnothing, \varnothing, \varnothing\rangle\}}{N}$
subsume	$\dfrac{N \uplus \{\langle s : t, R_0, R_1, E\rangle,\ \langle s' : t', R_0', R_1', E'\rangle\}}{N \cup \{\langle s : t, R_0 \cup R_0', R_1 \cup R_1', E''\rangle\}}$
	if $s : t$, $s' : t'$ are variants, and $E'' = (E \setminus (R_0' \cup R_1')) \cup (E' \setminus (R_0 \cup R_1))$

Figure 4.1: The inference system MKB of multi-completion.

Intuitively, a node $\langle s : t, R_0, R_1, E\rangle$ captures the state of the term pair $s : t$ in all simulated completion processes. All orders in the *equation label* E regard the data as an equation $s \approx t$ while orders in the *rewrite labels* R_0 and R_1 consider it as rewrite rules $s \rightarrow t$ and $t \rightarrow s$, respectively. Hence the node $\langle s : t, R_0, R_1, E\rangle$ is identified with $\langle t : s, R_1, R_0, E\rangle$.

Multi-completion is described by the inference system MKB operating on sets of nodes as displayed in Figure 4.1. The rules gc and subsume are optional, they serve the purpose of garbage collection and avoiding redundant nodes. To relate a node set N to the simulated KB processes, projection functions are used.

Definition 4.2. For a node $n = \langle s : t, R_0, R_1, E\rangle$ and an order $\succ_i \in \mathcal{O}$, *equation*

and *rule projections* of n to $\succ_i$ are defined as

$$E[n,i] = \begin{cases} \{s \approx t\} & \text{if } \succ_i \in E \\ \varnothing & \text{otherwise} \end{cases} \qquad R[n,i] = \begin{cases} \{s \to t\} & \text{if } \succ_i \in R_0 \\ \{t \to s\} & \text{if } \succ_i \in R_1 \\ \varnothing & \text{otherwise} \end{cases}$$

These projections are naturally extended to node sets by defining $E[N,i] = \bigcup_{n \in N} E[n,i]$ and $R[N,i] = \bigcup_{n \in N} R[n,i]$.

For a set of input equalities $\mathcal{E}$, an MKB run $N_0 \vdash_{\mathsf{MKB}} N_1 \vdash_{\mathsf{MKB}} N_2 \vdash_{\mathsf{MKB}} \cdots$ starts with the *initial node set* $N_0 = N_{\mathcal{E}} = \{\langle s : t, \varnothing, \varnothing, \mathcal{O}\rangle \mid s \approx t \in \mathcal{E}\}$. We define the set of *persistent nodes* as $N_\omega = \bigcup_i \bigcap_{j \geqslant i} N_j$. The projection functions can now be used to express relationships between MKB inference sequences and simulated KB runs for orders $\succ_i \in \mathcal{O}$. The following simulation results [62] are crucial to establish correctness of MKB.

Simulation Soundness Lemma 4.3. *If* $N_0 \vdash^{\alpha}_{\mathsf{MKB}} N_\alpha$ *then for all* $\succ_i \in \mathcal{O}$ *there is some* $\beta \leqslant \alpha$ *such that* $(E[N_0,i], R[N_0,i]) \vdash^{\beta}_{\mathsf{KB}} (E[N_\alpha,i], R[N_\alpha,i])$.

Proof. We first consider a single step $N \vdash_{\mathsf{MKB}} N'$ and apply case analysis to prove validity of the KB step $(E[N,i], R[N,i]) \vdash^{=}_{\mathsf{KB}} (E[N',i], R[N',i])$ which we refer to by $(*)$.

- Assume orient replaced the node $n = \langle s : t, R_0, R_1, E \uplus R\rangle$ by $n' = \langle s : t, R_0 \cup R, R_1, E\rangle$. For all $\succ_i \in \mathcal{O}$ we have $E[N \setminus \{n\}, i] = E[N' \setminus \{n'\}, i]$ and $R[N \setminus \{n\}, i] = R[N' \setminus \{n'\}, i]$. A case distinction reveals two possibilities: If $\succ_i \in R$, then $s \succ_i t$ must hold and we have $R[n,i] = \varnothing$ and $E[n,i] = \{s \approx t\}$, whereas $R[n',i] = \{s \to t\}$ and $E[n',i] = \varnothing$. Thus $(*)$ is a valid orient step in KB. Otherwise, if $\succ_i \notin R$ then we have $E[n,i] = E[n',i]$ and $R[n,i] = R[n',i]$. The projection of the considered MKB inference to $\succ_i$ is thus an identity step, that is, $(E[N,i], R[N,i]) = (E[N',i], R[N',i])$.

- If deduce adds a node $\langle s : t, \varnothing, \varnothing, R \cap R'\rangle$ then for all $\succ_i \in R \cap R'$ both $\ell_1 \to r_1$ and $\ell_2 \to r_2$ occur in $R[N,i]$ such that $s \leftarrow \times \rightarrow t \in \mathrm{CP}(\ell_1 \to r_1, \ell_2 \to r_2)$, and $s \simeq t \in E[N',i]$. Therefore $(*)$ is a valid deduce step in KB. For all $\succ_i \notin R \cap R'$ an identity step is obtained.

- Assume delete removes a node $\langle s : s, \varnothing, \varnothing, E\rangle$ and $\succ_i \in E$. Then $s \approx s \in E[N,i]$ and $s \approx s \notin E[N',i]$, hence $(*)$ is a valid delete step in KB. For all $\succ_i \notin E$ an identity step is obtained.

- Next, assume $\mathsf{rewrite}_1$ was used. For every order $\succ_i \notin (R_0 \cup E) \cap R$ we obtain an identity step. Otherwise, two cases have to be distinguished, which are distinct due to the node condition.

  i. If $\succ_i \in R_0 \cap R$ then $R[N,i]$ contains rules $s \to t$ and $\ell \to r$ such that $t \to_{\ell \to r} u$. Hence compose can be applied to replace $s \to t$ by $s \to u$. Indeed we have $s \to t \notin R[N',i]$ but $s \to u \in R[N',i]$, so $(*)$ holds.

ii. If $\succ_i \in E \cap R$ there is an equation $s \simeq t$ in $E[N, i]$ and a rule $\ell \to r$ in $R[N, i]$ such that $t \to_{\ell \to r} u$. Thus, in a KB step simplify can turn $s \simeq t$ into $s \simeq u$. As we have $s \simeq t \notin E[N', i]$ but $s \simeq u \in E[N', i]$, the step $(*)$ is valid in KB.

– In the case where $\mathsf{rewrite}_2$ was applied, the inference is an identity step for every order $\succ_i \notin (R_0 \cup R_1 \cup E) \cap R$. Otherwise, three disjoint possibilities can be distinguished. If $\succ_i \in R_0 \cap R$ or $\succ_i \in E \cap R$ then compose or simplify can be applied, as argued in the case for $\mathsf{rewrite}_1$.

iii. If $\succ_i \in R_1 \cap R$ then there are rules $\ell \to r$ and $t \to s$ in $R[N, i]$ such that a collapse step can turn the latter into an equation $u \approx s$ because $t \rhd \ell$. As we have $t \to s \notin R[N', i]$ but $u \approx s \in E[N', i]$, $(*)$ is a valid KB inference.

– If gc was applied then $(*)$ obviously corresponds to an identity step on the level of KB for every $\succ_i \in \mathcal{O}$, and the same holds for subsume.

By transfinite induction on α, we show that for a run $N_0 \vdash_{\mathsf{MKB}}^{\alpha} N_\alpha$ and $\succ_i \in \mathcal{O}$ there is some $\beta \leqslant \alpha$ such that there is a valid KB run $(E[N_0, i], R[N_0, i]) \vdash_{\mathsf{KB}}^{\beta} (E[N_\alpha, i], R[N_\alpha, i])$. The claim clearly holds if $\alpha = 0$. Let $\alpha = n + 1 \in \mathbb{N}$ and $\succ_i \in \mathcal{O}$. By the induction hypothesis there is some $m \leqslant n$ admitting a run $(E[N_0, i], R[N_0, i]) \vdash_{\mathsf{KB}}^{m} (E[N_n, i], R[N_n, i])$. According to the above case distinction also the step $N_n \vdash_{\mathsf{MKB}} N_{n+1}$ is reflected in a (possibly empty) KB step $(E[N_n, i], R[N_n, i]) \vdash_{\mathsf{KB}}^{=} (E[N_{n+1}, i], R[N_{n+1}, i])$, so $(E[N_0, i], R[N_0, i]) \vdash_{\mathsf{KB}}^{m'} (E[N_{n+1}, i], R[N_{n+1}, i])$ for $m' \leqslant n + 1$. Finally, if $\alpha = \omega$ then by the induction hypothesis for all $n \in \mathbb{N}$ and $\succ_i \in \mathcal{O}$ there is some $m \leqslant n$ such that $(E[N_0, i], R[N_0, i]) \vdash_{\mathsf{KB}}^{m} (E[N_n, i], R[N_n, i])$ is a valid KB run. Since $E[N_\omega, i] = E[\bigcup_k \bigcap_{j \geqslant k} N_j, i] = \bigcup_k \bigcap_{j \geqslant k} E[N_j, i]$ and similarly $R[N_\omega, i] = \bigcup_k \bigcap_{j \geqslant k} R[N_j, i]$, in the limit also $(E[N_0, i], R[N_0, i]) \vdash_{\mathsf{KB}}^{\omega} (E[N_\omega, i], R[N_\omega, i])$ is valid. □

Thus any MKB run γ can be projected to a valid KB run for every $\succ_i \in \mathcal{O}$, which we denote by γ_i.

Simulation Completeness Lemma 4.4. *Let N_0 be a node set such that $\mathcal{E}_0 = E[N_0, i]$ and $\mathcal{R}_0 = R[N_0, i]$ for some $\succ_i \in \mathcal{O}$ and there is a run $(\mathcal{E}_0, \mathcal{R}_0) \vdash_{\mathsf{KB}}^{\alpha} (\mathcal{E}_\alpha, \mathcal{R}_\alpha)$. Then there is a node set N_α such that $N_0 \vdash_{\mathsf{MKB}}^{\alpha} N_\alpha$, $\mathcal{E}_\alpha = E[N_\alpha, i]$, and $\mathcal{R}_\alpha = R[N_\alpha, i]$ hold.*

Proof. We first show that for any single step $(\mathcal{E}, \mathcal{R}) \vdash_{\mathsf{KB}} (\mathcal{E}', \mathcal{R}')$ such that $\mathcal{E} = E[N, i]$ and $\mathcal{R} = R[N, i]$ for some $\succ_i \in \mathcal{O}$ there is a node set N' such that $N \vdash_{\mathsf{MKB}} N'$, $\mathcal{E}' = E[N', i]$, and $\mathcal{R}' = R[N', i]$ hold. In the following case analysis on the applied KB rule, $(*)$ refers to the proof obligations $\mathcal{E}' = E[N', i]$ and $\mathcal{R}' = R[N', i]$.

– Assume orient was applied to replace some equation $s \simeq t \in \mathcal{E}$ by the rule $s \to t \in \mathcal{R}'$. Then there must be a node $n = \langle s : t, R_0, R_1, E \rangle$ in N such that $\succ_i \in E$ and $s \succ_i t$. We can thus apply orient in MKB with $R = \{\succ_i\}$ to obtain $N' = (N \setminus \{n\}) \cup \{\langle s : t, R_0 \cup \{\succ_i\}, R_1, E \setminus \{\succ_i\} \rangle\}$, so $s \simeq t \notin E[N', i]$ but $s \to t \in R[N', i]$ such that $(*)$ holds.

– If deduce generates $s \approx t$ from an overlap involving rules $\ell_1 \to r_1$ and $\ell_2 \to r_2$, there are nodes $\langle \ell_1 : r_1, R, \dots \rangle$ and $\langle \ell_2 : r_2, R', \dots \rangle$ in N such that $\succ_i \in R \cap R'$. So we can apply deduce in MKB to obtain $N' = N \cup \{\langle s : t, \varnothing, \varnothing, \{\succ_i\}, \varnothing, \varnothing\rangle\}$ and as $s \approx t \in E[N', i]$ $(*)$, is satisfied.

– If delete removes some equation $s \approx s$ from $\mathcal{E}$ then N must contain a node $n = \langle s : s, R_0, R_1, E \uplus \{\succ_i\}\rangle$. Since no reduction order can orient the equation $s \approx s$, the sets R_0 and R_1 must be empty. Thus n can be removed by delete in MKB, and obviously $(*)$ holds.

– If simplify reduces an equation $s \simeq t$ to $s \simeq u$ using a rule $\ell \to r$, there are nodes $n = \langle s : t, R_0, R_1, E\rangle$ and $\langle \ell : r, R, \dots \rangle$ in N such that $\succ_i \in E \cap R$. If t is a variant of ℓ we can therefore use $\mathsf{rewrite}_1$ and otherwise $\mathsf{rewrite}_2$ to infer $N' = (N \setminus \{n\}) \cup \{\langle s : t, R_0, R_1, E \setminus \{\succ_i\}\rangle\} \cup \{\langle s : u, \varnothing, \varnothing, \{\succ_i\}\rangle\}$ Since $s \simeq u$ instead of $s \simeq t$ occurs in $E[N', i]$, $(*)$ holds.

– If compose rewrites $s \to t$ to $s \to u$ using a rule $\ell \to r$, N contains nodes $n = \langle s : t, R_0, R_1, E\rangle$ and $\langle \ell : r, R, \dots \rangle$ such that $\succ_i \in R_0 \cap R$. Thus $\mathsf{rewrite}_1$ or $\mathsf{rewrite}_2$ applies, depending on whether $t \doteq \ell$ or $t \rhd \ell$. We obtain $N' = (N \setminus \{n\}) \cup \{\langle s : t, R_0 \setminus \{\succ_i\}, R_1, E\rangle, \langle s : u, \{\succ_i\}, \varnothing, \varnothing\rangle\}$ and as $s \to u$ instead of $s \to t$ occurs in $R[N', i]$, $(*)$ is satisfied.

– Finally, assume collapse turns a rule $t \to s$ into an equation $u \approx s$ using $\ell \to r$. Then $t \rhd \ell$ must hold, and the set N must contain nodes $n = \langle s : t, R_0, R_1, E\rangle$ and $\langle \ell : r, R, \dots \rangle$ such that $\succ_i$ occurs in $R_1 \cap R$. So again $(*)$ can be satisfied if N' is obtained from N with an application of $\mathsf{rewrite}_2$ such that $N' = (N \setminus \{n\}) \cup \{\langle s : t, R_0, R_1 \setminus \{\succ_i\}, E\rangle\} \cup \{\langle s : u, \varnothing, \varnothing, \{\succ_i\}\rangle\}$. Since $t \to s \notin R[N', i]$ but $s \approx u \in E[N', i]$, $(*)$ holds.

By transfinite induction on α, we now show that whenever $\mathcal{E}_0 = E[N_0, i]$ and $\mathcal{R}_0 = R[N_0, i]$ for some $\succ_i \in \mathcal{O}$ and there is a run $(\mathcal{E}_0, \mathcal{R}_0) \vdash_{\mathsf{KB}}^{\alpha} (\mathcal{E}_\alpha, \mathcal{R}_\alpha)$ then there is also a node set N_α such that $N \vdash_{\mathsf{MKB}}^{\alpha} N_\alpha$, $\mathcal{E}_\alpha = E[N_\alpha, i]$, and $\mathcal{R}_\alpha = R[N_\alpha, i]$ hold. The claim clearly holds if $\alpha = 0$. If $\alpha = n + 1 \in \mathbb{N}$ then by the induction hypothesis there is some N_n such that $N \vdash_{\mathsf{MKB}}^{n} N_n$, $\mathcal{E}_n = E[N_n, i]$, and $\mathcal{R}_n = R[N_n, i]$ hold. According to the above case distinction also the step $(\mathcal{E}_n, \mathcal{R}_n) \vdash_{\mathsf{KB}} (\mathcal{E}_{n+1}, \mathcal{R}_{n+1})$ can be reflected in a step $N_n \vdash_{\mathsf{MKB}} N_{n+1}$ such that $\mathcal{E}_{n+1} = E[N_{n+1}, i]$ and $\mathcal{R}_{n+1} = R[N_{n+1}, i]$ hold. Finally, if $\alpha = \omega$ then by the induction hypothesis there are valid MKB runs $N \vdash_{\mathsf{MKB}}^{n} N_n$ such that $\mathcal{E}_n = E[N_n, i]$ and $\mathcal{R}_n = R[N_n, i]$ for all $n \in \mathbb{N}$. Thus in the limit also $N \vdash_{\mathsf{MKB}}^{\omega} N_\omega$ is valid, and we have

$$\mathcal{E}_\omega = \bigcup_k \bigcap_{j>k} \mathcal{E}_j = \bigcup_k \bigcap_{j>k} E[N_j, i] = E[\bigcup_k \bigcap_{j>k} N_j, i] = E[N_\omega, i]$$

and similarly also $\mathcal{R}_\omega = R[N_\omega, i]$. □

Definition 4.5. A run γ *succeeds* if γ_i succeeds for some $\succ_i \in \mathcal{O}$, and it *fails* if γ_i fails for all $\succ_i \in \mathcal{O}$. It is *fair* if it is finite and some γ_i is a nonfailing and

fair[1] KB run, or if it is infinite and all γ_i are either failing or fair.

From the simulation properties it is straightforward to derive correctness and completeness results for MKB.

Correctness Theorem 4.6. *Any nonfailing and fair run* $N_{\mathcal{E}} \vdash^{\alpha}_{\mathsf{MKB}} N$ *with* $\alpha \leqslant \omega$ *succeeds.*

Proof. By Lemma 4.3 and fairness there is some $\succ_i \in \mathcal{O}$ such that the KB run γ_i is fair and nonfailing. By Theorem 3.12 the run γ_i succeeds. □

Completeness Theorem 4.7. *Let* $\mathcal{R}$ *be a finite convergent TRS for* $\mathcal{E}$ *such that* $\mathcal{R} \subseteq \succ_i$ *for some* $\succ_i \in \mathcal{O}$*. If the* KB *run* γ_i *associated with an* MKB *run* γ *starting from* $\mathcal{N}_{\mathcal{E}}$ *is fair and nonfailing then* γ *succeeds with a TRS* $\mathcal{R}'$ *in finitely many steps.*

Proof. As γ_i is fair and nonfailing this is immediate from Theorem 3.13 and Definition 4.5. □

Example 4.8. Consider the set of equations from Example 3.17:

$$\mathsf{f}(\mathsf{f}(\mathsf{g}(x))) \approx x \tag{1}$$
$$\mathsf{g}(\mathsf{g}(x)) \approx \mathsf{f}(\mathsf{g}(x)) \tag{2}$$

and let $\succ_1$ and $\succ_2$ be KBOs with $w_0 = w(\mathsf{f}) = w(\mathsf{g}) = 1$, where $\succ_1$ has precedence $\mathsf{g} \succ \mathsf{f}$ while in $\succ_2$ we have $\mathsf{f} \succ \mathsf{g}$. Then the initial node set $N_{\mathcal{E}}$ contains the nodes

$$\langle \mathsf{f}(\mathsf{f}(\mathsf{g}(x))) : x, \varnothing, \varnothing, \{\succ_1, \succ_2\}\rangle \tag{1}$$
$$\langle \mathsf{g}(\mathsf{g}(x)) : \mathsf{f}(\mathsf{g}(x)), \varnothing, \varnothing, \{\succ_1, \succ_2\}\rangle \tag{2}$$

When applying orient to both nodes, we obtain

$$\langle \mathsf{f}(\mathsf{f}(\mathsf{g}(x))) : x, \{\succ_1, \succ_2\}, \varnothing, \varnothing\rangle \tag{1}$$
$$\langle \mathsf{g}(\mathsf{g}(x)) : \mathsf{f}(\mathsf{g}(x)), \{\succ_1\}, \{\succ_2\}, \varnothing\rangle \tag{2}$$

As the rules $\mathsf{f}(\mathsf{f}(\mathsf{g}(x))) \to x$ and $\mathsf{g}(\mathsf{g}(x)) \to \mathsf{f}(\mathsf{g}(x))$ give rise to a critical overlap, deduce adds the node

$$\langle \mathsf{f}(\mathsf{f}(\mathsf{f}(\mathsf{g}(x)))) : \mathsf{g}(x), \varnothing, \varnothing, \{\succ_1\}\rangle \tag{3}$$

When this node is oriented and the process using $\succ_1$ is advanced further, a convergent system is only derived in the limit, generating infinitely many nodes with data $\mathsf{f}^{n+2}(\mathsf{g}(x)) : \mathsf{g}^n(x)$ for all $n \geqslant 0$. But a fair MKB run also needs to consider the process using $\succ_2$ at some point. Then $\mathsf{rewrite}_2$ can apply node (2) to (1), which changes (1) and adds a new node:

$$\langle \mathsf{f}(\mathsf{f}(\mathsf{g}(x))) : x, \{\succ_1\}, \varnothing, \varnothing\rangle \tag{1}$$
$$\langle \mathsf{f}(\mathsf{g}(\mathsf{g}(x))) : x, \varnothing, \varnothing, \{\succ_2\}\rangle \tag{4}$$

1 According to the original definition in [62] a finite MKB run is fair if it fair for some i. But then a finite, fair and nonfailing MKB run γ need not succeed: a run γ which is fair for only one i would still be fair even if γ_i is failing. We thus use the modified notion defined above.

$$\textsf{rewrite}\quad \frac{N \uplus \{\langle s : t, R_0, R_1, E\rangle\}}{N \cup \{\langle s : t, R_0 \setminus R, R_1 \setminus R, E \setminus R\rangle,\ \langle s : u, R_0 \cap R, \varnothing, (E \cup R_1) \cap R\rangle\}}$$

$$\text{if } \langle \ell : r, R, \ldots\rangle \in N,\ t \to_{\ell \to r} u, \text{ and } (R_0 \cup R_1 \cup E) \cap R \neq \varnothing$$

Figure 4.2: The rewrite rule in MKB′.

Another application of $\textsf{rewrite}_2$ using (2) yields

$$\langle \mathsf{f}(\mathsf{g}(\mathsf{g}(x))) : x, \varnothing, \varnothing, \varnothing\rangle \tag{4}$$
$$\langle \mathsf{g}(\mathsf{g}(\mathsf{g}(x))) : x, \varnothing, \varnothing, \{\succ_2\}\rangle \tag{5}$$

Node (4) can be removed by gc, and node (5) is oriented as

$$\langle \mathsf{g}(\mathsf{g}(\mathsf{g}(x))) : x, \{\succ_2\}, \varnothing, \varnothing\rangle \tag{5}$$

Now deduce produces the following additional nodes:

$$\langle \mathsf{f}(x) : \mathsf{g}(\mathsf{g}(\mathsf{g}(\mathsf{g}(x)))), \varnothing, \varnothing, \{\succ_2\}\rangle \tag{6}$$
$$\langle \mathsf{g}(\mathsf{g}(\mathsf{g}(\mathsf{g}(x)))) : \mathsf{g}(\mathsf{g}(\mathsf{g}(\mathsf{g}(x)))), \varnothing, \varnothing, \{\succ_2\}\rangle \tag{7}$$
$$\langle \mathsf{g}(\mathsf{g}(\mathsf{g}(\mathsf{g}(\mathsf{g}(x))))) : \mathsf{g}(\mathsf{g}(\mathsf{g}(\mathsf{g}(\mathsf{g}(x))))), \varnothing, \varnothing, \{\succ_2\}\rangle \tag{8}$$

While (7) and (8) are subject to delete, we can apply (5) in a $\textsf{rewrite}_2$ step to (6). This results in (5) being empty and a new node which is oriented as

$$\langle \mathsf{f}(x) : \mathsf{g}(x), \{\succ_2\}, \varnothing, \varnothing\rangle \tag{9}$$

For the current node set N we now have $R[N, 2] = \{\mathsf{g}(\mathsf{g}(\mathsf{g}(x))) \to x, \mathsf{f}(x) \to \mathsf{g}(x)\}$ and $E[N, 2] = \varnothing$. As $R[N, 2]$ admits no further critical pairs, the projected run γ_2 is fair, so $R[N, 2]$ is convergent.

Finite Runs

When restricting to finite runs, multi-completion can be defined to emulate KB′, which simplifies the inference system in that both rewrite inference rules can be combined. We thus define finite multi-completion by the inference system MKB′, which contains the orient, deduce, and delete rules (possibly together with the optional subsume and gc) from MKB, plus the single rewrite rule displayed in Figure 4.2.

With proofs very similar to those of Lemmas 4.3 and 4.4, we obtain the following results.

Simulation Soundness Lemma 4.9. *If $N_0 \vdash^*_{\mathsf{MKB}'} N_n$ is a finite run then for all $\succ_i \in \mathcal{O}$ we have $(E[N_0, i], R[N_0, i]) \vdash^*_{\mathsf{KB}'} (E[N_n, i], R[N_n, i])$.* □

As for MKB, any MKB′ run γ can thus be projected to a valid KB′ run for every $\succ_i \in \mathcal{O}$, which we denote by γ_i.

Simulation Completeness Lemma 4.10. *Let N_0 be a node set such that $\mathcal{E}_0 = E[N_0, i]$, and $\mathcal{R}_0 = R[N_0, i]$ for some $\succ_i \in \mathcal{O}$ and there is a finite run $(\mathcal{E}_0, \mathcal{R}_0) \vdash^*_{\mathsf{KB}'} (\mathcal{E}_\alpha, \mathcal{R}_\alpha)$. Then there is a node set N_n such that $N_0 \vdash^*_{\mathsf{MKB}'} N_n$, $\mathcal{E}_n = E[N_n, i]$, and $\mathcal{R}_n = R[N_n, i]$ hold.* □

Success, failure and fairness of MKB′ runs is defined as for MKB runs. From the simulation properties, correctness and completeness of KB′ (Theorems 3.12 and 3.13) it is thus straightforward to derive correctness and completeness results for MKB′ in the same way as for MKB.

Correctness Theorem 4.11. *Any finite nonfailing and fair run $N_\mathcal{E} \vdash^n_{\mathsf{MKB}'} N$ succeeds.* □

Completeness Theorem 4.12. *Let $\mathcal{R}$ be a finite convergent TRS for $\mathcal{E}$ such that $\mathcal{R} \subseteq \succ_i$ for some $\succ_i \in \mathcal{O}$. If the finite* KB′ *run γ_i associated with an* MKB *run γ starting from $\mathcal{N}_\mathcal{E}$ is fair and nonfailing then γ succeeds with a TRS $\mathcal{R}'$ in finitely many steps.* □

4.2 Completion with Termination Tools

As standard completion procedures critically depend on the choice of the reduction order supplied as input, the evolution of powerful modern termination provers offers the potential to guarantee termination by means of automatic tools instead of a fixed order. This approach due to Wehrman, Stump and Westbrook [107] was implemented in the tool Slothrop. Some care has to be taken because it is known that changing the reduction order during a completion run may result in a non-confluent rewrite system [91]. The inference system KBtt underlying Slothrop thus operates on triples $(\mathcal{E}, \mathcal{R}, \mathcal{C})$ consisting of a set of equations $\mathcal{E}$, a rewrite system $\mathcal{R}$ and an additional rewrite system $\mathcal{C}$. This extra *constraint system* ensures that orientations are never reversed throughout a run, thereby guaranteeing confluence of the derived system.

The resulting inference system is shown in Figure 4.3. A KBtt inference sequence of the form

$$(\mathcal{E}_0, \mathcal{R}_0, \mathcal{C}_0) \vdash (\mathcal{E}_1, \mathcal{R}_1, \mathcal{C}_1) \vdash (\mathcal{E}_2, \mathcal{R}_2, \mathcal{C}_2) \vdash \cdots$$

is called a *run*. We will in the sequel assume that $\mathcal{R}_0 = \mathcal{C}_0 = \varnothing$ although all results in the remainder of this section generalize to the setting where $\mathcal{R}_0$ and $\mathcal{C}_0$ are non-empty, provided that $\mathcal{R}_0 = \mathcal{C}_0 \subseteq \succ$.

Correctness of KBtt can be shown by relating it to a KB run that uses as reduction order the transitive closure of the rewrite relation induced by the final constraint system $\mathcal{C}$. Since $\mathcal{C}$ is obtained as the union of all oriented rewrite rules and the limit of a sequence of terminating TRSs need not be terminating, correctness of KBtt is only guaranteed for finite runs [107]. A simple counterexample for infinite runs is the following:

Example 4.13. Consider the set of equations consisting of the following two equations:

$$\mathsf{f}(\mathsf{f}(\mathsf{g}(x))) \approx \mathsf{f}(\mathsf{g}(\mathsf{g}(x))) \qquad\qquad \mathsf{f}(\mathsf{g}(\mathsf{a})) \approx \mathsf{f}(\mathsf{g}(\mathsf{g}(\mathsf{a})))$$

orient	$\dfrac{\mathcal{E} \uplus \{s \simeq t\}, \mathcal{R}, \mathcal{C}}{\mathcal{E}, \mathcal{R} \cup \{s \to t\}, \mathcal{C} \cup \{s \to t\}}$	if $\mathcal{C} \cup \{s \to t\}$ terminates
deduce	$\dfrac{\mathcal{E}, \mathcal{R}, \mathcal{C}}{\mathcal{E} \cup \{s \approx t\}, \mathcal{R}, \mathcal{C}}$	if $s \leftarrow \rtimes \rightarrow t \in \mathrm{CP}(\mathcal{R})$
delete	$\dfrac{\mathcal{E} \uplus \{s \approx s\}, \mathcal{R}, \mathcal{C}}{\mathcal{E}, \mathcal{R}, \mathcal{C}}$	
simplify	$\dfrac{\mathcal{E} \uplus \{s \simeq t\}, \mathcal{R}, \mathcal{C}}{\mathcal{E} \cup \{s \simeq u\}, \mathcal{R}, \mathcal{C}}$	if $t \to_{\mathcal{R}} u$
compose	$\dfrac{\mathcal{E}, \mathcal{R} \uplus \{s \to t\}, \mathcal{C}}{\mathcal{E}, \mathcal{R} \cup \{s \to u\}, \mathcal{C}}$	if $t \to_{\mathcal{R}} u$
collapse	$\dfrac{\mathcal{E}, \mathcal{R} \uplus \{t \to s\}, \mathcal{C}}{\mathcal{E} \cup \{u \approx s\}, \mathcal{R}, \mathcal{C}}$	if $t \to_{\mathcal{R}} u$

Figure 4.3: The inference system KBtt of completion with termination tools.

In a KBtt run both equations can be oriented from left to right:

$$\mathsf{f}(\mathsf{f}(\mathsf{g}(x))) \to \mathsf{f}(\mathsf{g}(\mathsf{g}(x))) \tag{1}$$
$$\mathsf{f}(\mathsf{g}(\mathsf{a})) \to \mathsf{f}(\mathsf{g}(\mathsf{g}(\mathsf{a}))) \tag{2}$$

as the rewrite system $\{(1), (2)\}$ terminates. The overlap $\langle (2), 1, (1) \rangle$ creates the critical pair $\mathsf{f}(\mathsf{f}(\mathsf{g}(\mathsf{g}(\mathsf{a})))) \leftarrow \rtimes \rightarrow \mathsf{f}(\mathsf{g}(\mathsf{g}(\mathsf{a})))$. The corresponding equation is simplified to $\mathsf{f}(\mathsf{g}(\mathsf{g}(\mathsf{g}(\mathsf{a})))) \approx \mathsf{f}(\mathsf{g}(\mathsf{g}(\mathsf{a})))$ using rule (1). This equation can be oriented as

$$\mathsf{f}(\mathsf{g}(\mathsf{g}(\mathsf{a}))) \to \mathsf{f}(\mathsf{g}(\mathsf{g}(\mathsf{g}(\mathsf{a})))) \tag{3}$$

because the constraint system $\{(1), (2), (3)\}$ terminates. In a similar fashion the overlap $\langle (3), 1, (1) \rangle$ creates a critical pair $\mathsf{f}(\mathsf{f}(\mathsf{g}(\mathsf{g}(\mathsf{g}(\mathsf{a}))))) \leftarrow \rtimes \rightarrow \mathsf{f}(\mathsf{g}(\mathsf{g}(\mathsf{g}(\mathsf{a})))))$, the corresponding equation is simplified using rule (1) and oriented to

$$\mathsf{f}(\mathsf{g}(\mathsf{g}(\mathsf{g}(\mathsf{a})))) \to \mathsf{f}(\mathsf{g}(\mathsf{g}(\mathsf{g}(\mathsf{g}(\mathsf{a})))))) \tag{4}$$

It is easy to see that this run subsequently creates the terminating rewrite systems $\mathcal{R}_n = \{\mathsf{f}(\mathsf{f}(\mathsf{g}(x))) \to \mathsf{f}(\mathsf{g}(\mathsf{g}(x)))\} \cup \{\mathsf{f}(\mathsf{g}^k(\mathsf{a})) \to \mathsf{f}(\mathsf{g}^{k+1}(\mathsf{a})) \mid 1 \leqslant k \leqslant n\}$. But in the limit we obtain $\mathcal{R}_\omega = \mathcal{C}_\omega = \{\mathsf{f}(\mathsf{f}(\mathsf{g}(x))) \to \mathsf{f}(\mathsf{g}(\mathsf{g}(x)))\} \cup \{\mathsf{f}(\mathsf{g}^k(\mathsf{a})) \to \mathsf{f}(\mathsf{g}^{k+1}(\mathsf{a})) \mid 1 \leqslant k\}$ which is non-terminating as it admits the infinite sequence

$$\mathsf{f}(\mathsf{g}(\mathsf{a})) \to \mathsf{f}(\mathsf{g}(\mathsf{g}(\mathsf{a}))) \to \mathsf{f}(\mathsf{g}(\mathsf{g}(\mathsf{g}(\mathsf{a})))) \to \cdots$$

We thus only consider finite KBtt runs

$$(\mathcal{E}_0, \mathcal{R}_0, \mathcal{C}_0) \vdash (\mathcal{E}_1, \mathcal{R}_1, \mathcal{C}_1) \vdash \cdots \vdash (\mathcal{E}_n, \mathcal{R}_n, \mathcal{C}_n)$$

For this reason Figure 4.3 differs from the original inference system [107] in that the collapse rule was relaxed as done for KB'. Success, failure and fairness are defined exactly as for KB'. The following simulation properties relate KBtt runs to KB' runs [107].

Simulation Soundness Lemma 4.14. *Any run* $(\mathcal{E}_0, \varnothing, \varnothing) \vdash^n_{\mathsf{KBtt}} (\mathcal{E}_n, \mathcal{R}_n, \mathcal{C}_n)$ *admits a* KB' *run* $(\mathcal{E}_0, \varnothing) \vdash^n_{\mathsf{KB}'} (\mathcal{E}_n, \mathcal{R}_n)$ *using reduction order* $\to^+_{\mathcal{C}_n}$.

Proof. Let $\succ_n$ denote $\to^+_{\mathcal{C}_n}$. We use induction on n. The claim is trivially true for $n = 0$. For a run of the form $(\mathcal{E}_0, \varnothing, \varnothing) \vdash^*_{\mathsf{KBtt}} (\mathcal{E}_n, \mathcal{R}_n, \mathcal{C}_n) \vdash_{\mathsf{KBtt}} (\mathcal{E}_{n+1}, \mathcal{R}_{n+1}, \mathcal{C}_{n+1})$, the induction hypothesis yields a corresponding KB' run $(\mathcal{E}_0, \varnothing) \vdash^*_{\mathsf{KB}'} (\mathcal{E}_n, \mathcal{R}_n)$ using the reduction order $\succ_n$. Since constraint rules are never removed we have $\mathcal{C}_k \subseteq \mathcal{C}_{n+1}$ and hence $\succ_k \subseteq \succ_n$ for all $k \leqslant n$ so the same run can be obtained with $\succ_{n+1}$. Case distinction on the applied KBtt rule shows that a step $(\mathcal{E}_n, \mathcal{R}_n) \vdash_{\mathsf{KB}'} (\mathcal{E}_{n+1}, \mathcal{R}_{n+1})$ using $\succ_{n+1}$ is possible: If orient added the rule $s \to t$ then $s \to t \in \mathcal{C}_{n+1}$, so $s \succ_{n+1} t$ holds by definition and KB' can apply orient as well. Clearly, in the remaining cases the inference step can be simulated by the corresponding KB' rule since no conditions on the order are involved. □

Simulation Completeness Lemma 4.15. *If* $(\mathcal{E}_0, \varnothing) \vdash^n_{\mathsf{KB}'} (\mathcal{E}_n, \mathcal{R}_n)$ *is a valid* KB' *run using* $\succ$ *then there is also a valid* KBtt *run* $(\mathcal{E}_0, \varnothing, \varnothing) \vdash^n_{\mathsf{KBtt}} (\mathcal{E}_n, \mathcal{R}_n, \mathcal{C}_n)$ *such that* $\mathcal{C}_n \subseteq \succ$.

Proof. By induction on n. For $n = 0$ the claim is trivially satisfied. If $\alpha = n + 1$ then the induction hypothesis yields a KBtt run $(\mathcal{E}_0, \varnothing, \varnothing) \vdash^n_{\mathsf{KBtt}} (\mathcal{E}_n, \mathcal{R}_n, \mathcal{C}_n)$ such that $\mathcal{C}_n \subseteq \succ$ An easy case distinction on the last inference step $(\mathcal{E}_n, \mathcal{R}_n) \vdash_{\mathsf{KB}'} (\mathcal{E}_{n+1}, \mathcal{R}_{n+1})$ shows that using $\succ$ for termination checks allows for a corresponding KBtt step.

If the applied inference rule is orient then $\mathcal{E}_n = \mathcal{E}_{n+1} \uplus \{s \simeq t\}$, $\mathcal{R}_{n+1} = \mathcal{R}_n \uplus \{s \to t\}$ and $s \succ t$. Thus also $\mathcal{C}_n \cup \{s \to t\} \subseteq \succ$, ensuring termination of the extended constraint system $\mathcal{C}_{n+1}$. Hence the KBtt rule orient can be applied to obtain $(\mathcal{E}_n, \mathcal{R}_n, \mathcal{C}_n) \vdash_{\mathsf{KBtt}} (\mathcal{E}_n \setminus \{s \simeq t\}, \mathcal{R}_n \cup \{s \to t\}, \mathcal{C}_n \cup \{s \to t\})$. In the remaining cases one can set $\mathcal{C}_{n+1} = \mathcal{C}_n$ and replace the applied rule by its KBtt counterpart since no conditions on the order are involved. □

Correctness Theorem 4.16. *Any finite nonfailing and fair* KBtt *run succeeds.*

Proof. Let γ be a run $(\mathcal{E}_0, \varnothing, \varnothing) \vdash^n_{\mathsf{KBtt}} (\varnothing, \mathcal{R}_n, \mathcal{C}_n)$ which is finite and fair. According to Lemma 4.14, there exists a valid nonfailing KB' run $(\mathcal{E}_0, \varnothing) \vdash^n_{\mathsf{KB}'} (\varnothing, \mathcal{R}_n)$ which is also fair. By Theorem 3.24 this run succeeds, and thus also γ succeeds. □

Example 4.17. Convergent rewrite systems representing group theory with commuting endomorphisms have applications in decision procedures for uninterpreted functions, as used in SMT solvers [99]. Consider the set of equations

CGE_2 describing two commuting group endomorphisms:

$$\begin{aligned} \mathsf{e} \cdot x &\approx x & \mathsf{f}(x \cdot y) &\approx \mathsf{f}(x) \cdot \mathsf{f}(y) \\ \mathsf{i}(x) \cdot x &\approx \mathsf{e} & \mathsf{g}(x \cdot y) &\approx \mathsf{g}(x) \cdot \mathsf{g}(y) \\ x \cdot (y \cdot z) &\approx (x \cdot y) \cdot z & \mathsf{f}(x) \cdot \mathsf{g}(y) &\approx \mathsf{g}(y) \cdot \mathsf{f}(x) \end{aligned}$$

For this set of equations no completion procedure using LPO or KBO can construct a convergent TRS since the equation $\mathsf{f}(x) \cdot \mathsf{g}(y) \approx \mathsf{g}(y) \cdot \mathsf{f}(x)$ is not orientable. Using completion with termination tools, Slothrop [107] was the first tool to succeed with the following convergent TRS:

$$\begin{aligned} \mathsf{e} \cdot x &\to x & \mathsf{f}(x) \cdot \mathsf{f}(y) &\to \mathsf{f}(x \cdot y) & x \cdot (y \cdot z) &\to (x \cdot y) \cdot z \\ x \cdot \mathsf{e} &\to x & \mathsf{f}(\mathsf{e}) &\to \mathsf{e} & (x \cdot y) \cdot \mathsf{i}(y) &\to x \\ \mathsf{i}(x) \cdot x &\to \mathsf{e} & \mathsf{i}(\mathsf{f}(x)) &\to \mathsf{f}(\mathsf{i}(x)) & (x \cdot \mathsf{i}(y)) \cdot y &\to x \\ x \cdot \mathsf{i}(x) &\to \mathsf{e} & \mathsf{g}(x) \cdot \mathsf{g}(y) &\to \mathsf{g}(x \cdot y) & \mathsf{f}(x) \cdot (\mathsf{f}(y) \cdot z) &\to \mathsf{f}(x \cdot y) \cdot z \\ \mathsf{i}(\mathsf{e}) &\to \mathsf{e} & \mathsf{g}(\mathsf{e}) &\to \mathsf{e} & \mathsf{g}(x) \cdot (\mathsf{g}(y) \cdot z) &\to \mathsf{g}(x \cdot y) \cdot z \\ \mathsf{i}(\mathsf{i}(x)) &\to x & \mathsf{i}(\mathsf{g}(x)) &\to \mathsf{g}(\mathsf{i}(x)) & \mathsf{g}(x) \cdot (\mathsf{f}(y) \cdot z) &\to \mathsf{f}(x) \cdot (\mathsf{g}(y) \cdot z) \\ \mathsf{i}(x \cdot y) &\to \mathsf{i}(y) \cdot \mathsf{i}(x) & \mathsf{g}(x) \cdot \mathsf{f}(y) &\to \mathsf{f}(y) \cdot \mathsf{g}(x) \end{aligned}$$

4.3 Multi-Completion with Termination Tools

The reduction order required as input by standard completion procedures is crucial for success, but hard to predict in advance. Completion with termination tools (KBtt) as outlined in Section 4.2 circumvents this critical parameter and instead employs automatic termination tools. But the orient rule in KBtt is often not deterministic. If given an equation $s \approx t$ and a set $\mathcal{C}$ of previously oriented rules such that both $\mathcal{C} \cup \{s \to t\}$ and $\mathcal{C} \cup \{t \to s\}$ terminate then an implementation encounters the challenge how to deal with this choice. Slothrop uses a best-first strategy to decide which branch to explore further. In contrast, *multi-completion with termination tools* keeps track of both orientations but avoids an explosion of the search space by integrating the concept of multi-completion to share common inferences. We start with some preliminary definitions.

Multi-completion with termination tools can be conceived as the parallel simulation of multiple KBtt branches. Every branch corresponds to a series of decisions on how to orient nodes, which we call a *process* and describe by a sequence of bits.

Definition 4.18. A *process* p is a bit string in $(0+1)^*$. A set of processes P is called *well-encoded* if there are no pairs of processes p and p' in P such that p' is a proper prefix of p. The initial process is represented by the empty string ϵ.

Definition 4.19. Let P and S be well-encoded process sets. The *splitting* operation is defined by $split_S(P) = (P \setminus S) \cup \{p0, p1 \mid p \in P \cap S\}$.

The following result is easily verified.

Lemma 4.20. *If P and Q are well-encoded process sets then $P \cap Q$, $P \setminus Q$, and $split_P(Q)$ are again well-encoded.* □

Multi-completion procedures share inference steps among multiple processes by employing a data structure called *node*. All multi-completion procedures discussed in this chapter operate on sets of nodes.

Definition 4.21. A *node* $\langle s : t, R_0, R_1, E, C_0, C_1 \rangle$ contains as *data* two terms s and t and as *labels* sets of processes R_0, R_1, E, C_0, C_1. The *node condition* requires that $R_0 \cup C_0$, $R_1 \cup C_1$ and E are mutually disjoint.

The process sets R_0, R_1 are called *rewrite labels*, $\mathcal{E}$ is the *equation label* and C_0, C_1 are the *constraint labels*. The node $\langle s : t, R_0, R_1, E, C_0, C_1 \rangle$ is identified with $\langle t : s, R_1, R_0, E, C_1, C_0 \rangle$, as is the case for MKB. The sets of all processes occurring in a node n or a node set N are denoted by $\mathcal{P}(n)$ and $\mathcal{P}(N)$, respectively.

Definition 4.22. For a set of equations $\mathcal{E}$, the *initial node set* $N_\mathcal{E}$ consists of all nodes $\langle s : t, \varnothing, \varnothing, \{\epsilon\}, \varnothing, \varnothing \rangle$ such that $s \approx t$ is in $\mathcal{E}$.

We call a node n (node set N) well-encoded if $\mathcal{P}(n)$ ($\mathcal{P}(N)$) is. A node set will be said to adhere to the node condition if all its nodes do. Note that for any set of equations $\mathcal{E}$, the initial node set $N_\mathcal{E}$ is well-encoded and satisfies the node condition.

To relate a node set N to the corresponding states of simulated KBtt processes, *projections* are used in a similar way as for MKB.

Definition 4.23. For a node $n = \langle s : t, R_0, R_1, E, C_0, C_1 \rangle$ satisfying the node condition and a process p the *equation*, *rule*, and *constraint projections* of n to p are defined as

$$E[n,p] = \begin{cases} \{s \approx t\} & \text{if } p \in E \\ \varnothing & \text{otherwise} \end{cases}$$

$$R[n,p] = \begin{cases} \{s \to t\} & \text{if } p \in R_0 \\ \{t \to s\} & \text{if } p \in R_1 \\ \varnothing & \text{otherwise} \end{cases}$$

$$C[n,p] = \begin{cases} \{s \to t\} & \text{if } p \in C_0 \\ \{t \to s\} & \text{if } p \in C_1 \\ \varnothing & \text{otherwise} \end{cases}$$

These projections are naturally extended to node sets by defining $E[N,p] = \bigcup_{n \in N} E[n,p]$, $R[N,p] = \bigcup_{n \in N} R[n,p]$ and $C[N,p] = \bigcup_{n \in N} C[n,p]$.

The inference rules of MKBtt are depicted in Figure 4.4. All rules are to be applied to well-encoded node sets which satisfy the node condition.[2] The following paragraphs add some clarifying remarks on the inference rules.

2 Note that the inference system presented here is slightly simpler than the one given in [112] as it was observed that the two different rewrite rules could actually be merged.

orient

$$\frac{N \uplus \{\langle s : t, R_0, R_1, E, C_0, C_1\rangle\}}{\mathit{split}_S(N) \cup \{\langle s : t, R_0 \cup R_{lr}, R_1 \cup R_{rl}, E', C_0 \cup R_{lr}, C_1 \cup R_{rl}\rangle\}}$$

if $E_{lr}, E_{rl} \subseteq E$, $E' = E \setminus (E_{lr} \cup E_{rl})$, $C[N,p] \cup \{s \to t\}$ terminates for all $p \in E_{lr}$ and $C[N,p] \cup \{t \to s\}$ terminates for all $p \in E_{rl}$, $S = E_{lr} \cap E_{rl}$, $R_{lr} = (E_{lr} \setminus E_{rl}) \cup \{p0 \mid p \in S\}$ and $R_{rl} = (E_{rl} \setminus E_{lr}) \cup \{p1 \mid p \in S\}$, and $E_{lr} \cup E_{rl} \neq \varnothing$

deduce

$$\frac{N}{N \cup \{\langle s : t, \varnothing, \varnothing, R \cap R', \varnothing, \varnothing\rangle\}}$$

if there exist nodes $\langle \ell_1 : r_1, R, \dots\rangle$ and $\langle \ell_2 : r_2, R', \dots\rangle$ in N such that $s \leftarrow \rtimes \rightarrow t \in \mathrm{CP}(\ell_1 \to r_1, \ell_2 \to r_2)$ and $R \cap R' \neq \varnothing$

delete

$$\frac{N \uplus \{\langle s : s, \varnothing, \varnothing, E, \varnothing, \varnothing\rangle\}}{N}$$

if $E \neq \varnothing$

rewrite

$$\frac{N \uplus \{\langle s : t, R_0, R_1, E, C_0, C_1\rangle\}}{\begin{array}{l} N \cup \{\langle s : t, R_0 \setminus R, R_1 \setminus R, E \setminus R, C_0, C_1\rangle\} \\ \quad \cup \{\langle s : u, R_0 \cap R, \varnothing, (R_1 \cup E) \cap R, \varnothing, \varnothing\rangle\}\end{array}}$$

if $\langle \ell : r, R, \dots\rangle \in N$, $t \to_{\ell \to r} u$, and $R \cap (R_0 \cup R_1 \cup E) \neq \varnothing$

gc

$$\frac{N \uplus \{\langle s : t, \varnothing, \varnothing, \varnothing, \varnothing, \varnothing\rangle\}}{N}$$

subsume

$$\frac{N \uplus \{\langle s : t, R_0, R_1, E, C_0, C_1\rangle\} \uplus \{\langle s' : t', R_0', R_1', E', C_0', C_1'\rangle\}}{N \cup \{\langle s : t, R_0 \cup R_0', R_1 \cup R_1', E'', C_0 \cup C_0', C_1 \cup C_1'\rangle\}}$$

if $s : t$ and $s' : t'$ are variants, and $E'' = (E \setminus (R_0' \cup R_1' \cup C_0' \cup C_1')) \cup (E' \setminus (R_0 \cup R_1 \cup C_0 \cup C_1))$

Figure 4.4: The inference system MKBtt.

– The orient rule applied to a node $\langle s : t, R_0, R_1, E, C_0, C_1\rangle$ attempts to turn the equation $s \approx t$ into a rule $s \to t$ or $t \to s$ for as many processes as possible. This is modelled in the node structure by moving processes $p \in E$ to rewrite labels. More precisely, the respective inference rule in KBtt is modelled by checking for every process $p \in E$ whether its current constraint system $C[N,p]$ terminates when extended with $s \to t$ or $t \to s$. If $C[N,p] \cup \{s \to t\}$ terminates then p is added to the set E_{lr}, and if $C[N,p] \cup \{t \to s\}$ terminates then p is added to the set E_{rl}. The set $E_{lr} \setminus E_{rl}$ ($E_{rl} \setminus E_{lr}$) thus collects processes which can only perform the orientation $s \to t$ ($t \to s$). These processes are added to R_0 and C_0 (R_1 and C_1). The set $S = E_{lr} \cap E_{rl}$ collects processes that allow both orientations. Thus every $p \in S$ is *split* into two child processes $p0$ and $p1$, and pi is added to R_i and C_i, for $i \in \{0,1\}$. Finally, $\mathit{split}_S(N)$ replaces every occurrence of a process p in S by its descendants $p0$ and $p1$ in all nodes in N.

- If the current node set N contains nodes with data $\ell_1 : r_1$ and $\ell_2 : r_2$ such that the rules $\ell_1 \to r_1$ and $\ell_2 \to r_2$ give rise to a critical pair $s \leftarrow\!\rtimes \to t$, deduce adds a respective node for all processes p that have both rules present in their current rewrite system $R[N, p]$.
- Nodes with equal terms may be removed by delete. Rewrite and constraint labels are assumed empty as a rule $s \to s$ would contradict termination.
- In standard completion, given a term pair $s : t$ and a rewrite step $t \xrightarrow{\ell \to r} u$, the rules compose, simplify and collapse create a term pair $s : u$. The MKBtt rule rewrite allows to simulate all three rules at once.
- To increase efficiency, the optional gc rule deletes nodes with empty labels.
- The rule subsume is optional as well, it merges pairs of nodes which have the same data up to variable renaming.

An MKBtt inference sequence $N_0 \vdash N_1 \vdash N_2 \vdash \cdots \vdash N_n$ where $N_0 = N_{\mathcal{E}}$ for some set of equations $\mathcal{E}$ is called a *run*. We will now show some simple yet crucial properties of MKBtt runs.

Lemma 4.24. *Consider an* MKBtt *run* $N_0 \vdash N_1 \vdash N_2 \vdash \cdots \vdash N_n$ *where* $N_0 = N_{\mathcal{E}}$ *for some set of equations* $\mathcal{E}$*, and let* $k \leqslant n$*. Then the rewrite system* $C[N_k, p]$ *is terminating and* $R[N_k, p] \subseteq \to^+_{C[N_k,p]}$ *for all* $p \in \mathcal{P}(N_k)$*.*

Proof. We apply induction on k and refer to the inclusion $R[N_k, p] \subseteq \to^+_{C[N_k,p]}$ by $(*)$. The claim holds for N_0 since for the single process ϵ occurring in N_0 we have $R[N_0, \epsilon] = C[N_0, \epsilon] = \varnothing$. For the induction step we assume that the claim holds for N_k and show that it is still true for N_{k+1} by a case distinction on the rule applied in $N_k \vdash N_{k+1}$, where the notation from Figure 4.4 is used.

- Assume orient is applied to a node $\langle s : t, R_0, R_1, E, C_0, C_1\rangle$. For every process $p \in \mathcal{P}(N_{k+1}) \setminus (R_{lr} \cup R_{rl})$ the claim holds as $R[N_k, p] = R[N_{k+1}, p]$ and $C[N_k, p] = C[N_{k+1}, p]$. Consider a process $p \in R_{lr}$. Thus $p \in E_{lr} \setminus E_{rl}$ or $p = q0$ such that $q \in S$. In the former case the TRS $C[N_{k+1}, p] = C[N_k, p] \cup \{s \to t\}$ terminates and $R[N_{k+1}, p] = R[N_k, p] \cup \{s \to t\}$, so by the induction hypothesis $(*)$ it follows that $R[N_{k+1}, p] \subseteq \to^+_{C[N_{k+1},p]}$. Similarly, in the latter case $C[N_{k+1}, p] = C[N_k, q] \cup \{s \to t\}$ terminates and $R[N_{k+1}, p] = R[N_k, q] \cup \{s \to t\}$, so the induction hypothesis $(*)$ applied to q implies $R[N_{k+1}, p] \subseteq \to^+_{C[N_{k+1},p]}$. The argument for a process in R_{rl} is symmetric.

In all remaining cases $C[N_k, p] = C[N_{k+1}, p]$, so $C[N_{k+1}, p]$ terminates according to the induction hypothesis for all $p \in \mathcal{P}(N_{k+1})$.

- Assume rewrite is applied. For $p \in R \cap R_0$ we have $R[N_{k+1}, p] = R[N_k, p] \setminus \{s \to t\} \cup \{s \to u\}$, and $R[N_k, p]$ contains rules $s \to t$ and $\ell \to r$ such that $t \to_{\ell \to r} u$. By $(*)$ and transitivity also $s \to^+_{C[N_{k+1},p]} u$ holds. For $p \in R \cap R_1$ we have $R[N_{k+1}, p] = R[N_k, p] \setminus \{s \to t\}$ and for $p \in R \cap E$ one obtains $R[N_{k+1}, p] = R[N_k, p]$, so the claim follows from the induction hypothesis.

– Assume subsume was applied. For all processes $p \in R'_0$ we have $R[N_{k+1}, p] = R[N_k, p] \setminus \{s' \to t'\} \cup \{s \to t\}$. Let $\mathcal{C}$ abbreviate $C[N_{k+1}, p]$. Since $s' \to t'$ and $s \to t$ are variants and $s' \to^+_{\mathcal{C}} t'$ by assumption, also $s \to^+_{\mathcal{C}} t$ holds. A similar argument applies to processes $p \in R'_1$. For all other processes we have $R[N_{k+1}, p] = R[N_k, p]$ such that the claim is satisfied.

– If delete, deduce, or gc was applied then $R[N_k, p] = R[N_{k+1}, p]$ for all $p \in \mathcal{P}(N_{k+1})$, so the proof obligations follow from the induction hypothesis. □

From this result we can immediately conclude that rewrite projections applied to node sets in MKBtt runs yield terminating rewrite systems.

Corollary 4.25. *In an* MKBtt *run* $N_0 \vdash N_1 \vdash N_2 \vdash \cdots$ *the rewrite system* $R[N_k, p]$ *is terminating for all node sets* N_k *with* $k \geqslant 0$ *and every process* $p \in \mathcal{P}(N_k)$. □

Lemma 4.26. *In an* MKBtt *run* $N_0 \vdash N_1 \vdash N_2 \vdash \cdots$ *every node set* N_k *is well-encoded and satisfies the node condition.*

Proof. We apply induction on k. The claim clearly holds for N_0. Otherwise, the well-encoded set N_k satisfies the node condition by the induction hypothesis. The following case distinction on the applied MKBtt rule shows that this also holds for N_{k+1}. We use the notation from Figure 4.4.

– Assume orient is applied to a node $n = \langle s : t, R_0, R_1, E, C_0, C_1 \rangle$. First we will argue that N_{k+1} is well-encoded. The only additional processes in N_{k+1} are those in $S_0 = \{p0 \mid p \in S\}$ and $S_1 = \{p1 \mid p \in S\}$. If some process $q \in \mathcal{P}(N_{k+1}) \setminus (S_0 \cup S_1) = \mathcal{P}(N_k) \setminus S$ were a prefix of a process $p0 \in S_0$ ($p1 \in S_1$) then it is either a prefix of $p \in \mathcal{P}(N_k)$ or $q = p0$ ($q = p1$). Both contradict well-encodedness of N_k. If a process $p0 \in S_0$ ($p1 \in S_1$) was a prefix of some process $q \in \mathcal{P}(N_k) \setminus S$ then also $p \in \mathcal{P}(N_k)$ must be a prefix of q. Since $p \in S$ but $q \notin S$ the processes p and q must be different, which contradicts well-encodedness of N_k. Hence N_{k+1} must be well-encoded.

  It remains to show that the additional node satisfies the node condition. But this is easy to see since n satisfies the node condition, the labels E_{lr} (E_{rl}) are moved from the equation label to the first (second) rewrite and constraint label, and also S_0 (S_1) is only added to the first (second) rewrite and constraint label.

In all remaining cases $\mathcal{P}(N_{k+1}) = \mathcal{P}(N_k)$, so N_{k+1} is clearly well-encoded.

– If deduce is applied then the single additional node in N_{k+1} has only one non-empty label, so N_{k+1} again adheres to the node condition.

– If delete or gc is applied then $N_{k+1} \subseteq N_k$, so N_{k+1} clearly satisfies the node condition.

– Assume rewrite is applied. For the node with data $s : u$, the label $R_0 \cap R$ is disjoint from $(R_1 \cup E) \cap R$ since R_0, R_1 and E are disjoint by assumption. As in the node with data $s : t$ processes are only removed from labels all nodes in N_{k+1} satisfy the node condition.

– Finally, consider the case where subsume is applied. First note that the equation label of the newly created node must be disjoint from all other labels: the processes in $E \setminus (R'_0 \cup R'_1 \cup C'_0 \cup C'_1)$ cannot occur in any other label as E is disjoint from all of R_0, R_1, C_0 and C_1 by the node condition, and a similar argument holds for the processes in $(E' \setminus (R_0 \cup R_1 \cup C_0 \cup C_1)$. Furthermore, no process in R_0 can occur in R_1 or C_1 by the well-encodedness assumption. Also the intersection $(R_0 \cup C_0) \cap (R'_1 \cup C'_1)$ must be empty: If there was a process $p \in R_0 \cap (R'_1 \cup C'_1)$ then according to Lemma 4.24 we have $s \to_{\mathcal{C}}^+ t$ and $t' \to_{\mathcal{C}}^+ s'$ for $\mathcal{C} = C[N_k, p]$. Since $s \to t$ and $s' \to t'$ are variants this contradicts termination of $C[N_k, p]$. A symmetric argument shows that $(R_1 \cup C_1) \cap (R'_0 \cup C'_0)$ is empty, so according to the definition of subsume the node set N_{k+1} satisfies the node condition. □

According to Lemma 4.26 all node sets occurring in a run are well-encoded and satisfy the node condition. This property will be used freely in the sequel. Note that we restrict to finite runs of some length n since the use of termination tools is only sound for finite runs (cf. Example 4.13).

Before we can state properties of MKBtt runs, notions to track process splits in the course of an inference sequence are required.

Definition 4.27. Consider an MKBtt inference step $N \vdash N'$. If orient was applied then the set of processes S which was split into two child processes is called the step's *split set*. For all other inference rules the split set is empty. For a step with split set S and $p' \in \mathcal{P}(N')$, we define the *predecessor* of p' as

$$pred_S(p') = \begin{cases} p & \text{if } p' = p0 \text{ or } p' = p1 \text{ for some } p \in S \\ p' & \text{otherwise} \end{cases}$$

In Lemmas 4.28 and 4.29 we prove that an MKBtt step corresponds to a (possibly non-proper) KBtt step for every process occurring in some node, and every KBtt step can be modelled by MKBtt. Here $\vdash^=$ denotes the reflexive closure of the KBtt inference relation $\vdash$.

Lemma 4.28. *Let N, N' be well-encoded node sets which satisfy the node condition. For an* MKBtt *step $N \vdash N'$ with split set S the* KBtt *step*

$$(E[N,p], R[N,p], C[N,p]) \vdash^= (E[N',p'], R[N',p'], C[N',p']) \tag{4.1}$$

is valid for all $p' \in \mathcal{P}(N')$ such that $p = pred_S(p')$. Moreover, there exists at least one process $p' \in \mathcal{P}(N')$ for which the step is not an equality step if the rule applied in $N \vdash N'$ is not gc *or* subsume.

Proof. By case analysis on the applied MKBtt rule in $N \vdash N'$.

– Assume orient with split set S replaced $n = \langle s : t, R_0, R_1, E, C_0, C_1\rangle$ by $n' = \langle s : t, R_0 \cup R_{lr}, R_1 \cup R_{rl}, E', C_0 \cup R_{lr}, C_1 \cup R_{rl}\rangle$. Let p' be a process in $\mathcal{P}(N')$ and $p = \mathit{pred}_S(p')$ be its predecessor with respect to S. We have $E[N \setminus \{n\}, p] = E[N' \setminus \{n'\}, p']$, $R[N \setminus \{n\}, p] = R[N' \setminus \{n'\}, p']$ and $C[N \setminus \{n\}, p] = C[N' \setminus \{n'\}, p']$. These sets will in the sequel be denoted by $\mathcal{E}$, $\mathcal{R}$ and $\mathcal{C}$, respectively. A further case distinction reveals three possibilities:

  i. If $p' \in R_{lr}$, by definition of orient $R[n', p'] = C[n', p'] = \{s \to t\}$ and $E[n', p'] = \varnothing$. Inference (4.1) is thus a valid orient step in KBtt if p happens to be in E. Since p' occurs in R_{lr}, either $p' \in E_{lr} \setminus E_{rl}$ or $p' = p0$ for some $p \in S$. If $p' \in E_{lr} \setminus E_{rl}$ then $p \in E$ follows from $p = \mathit{pred}_S(p') = p'$ and $E_{lr} \subseteq E$. Otherwise $p = \mathit{pred}_S(p')$ entails $p' = p0$ such that $p \in S$ and because of $S \subseteq E$ also $p \in E$ holds. As p occurs in E one has $E[n, p] = \{s \simeq t\}$ and—because of the node condition—$R[n, p] = C[n, p] = \varnothing$. Hence the KBtt inference step

$$(\mathcal{E} \uplus \{s \approx t\}, \mathcal{R}, \mathcal{C}) \vdash_{\mathsf{KBtt}} (\mathcal{E}, \mathcal{R} \cup \{s \to t\}, \mathcal{C} \cup \{s \to t\})$$

  is valid since $\mathcal{C} \cup \{s \to t\} = C[N, p] \cup \{s \to t\}$ terminates according to the side condition of orient in MKBtt.

  ii. If $p' \in R_{rl}$, similar reasoning as in the previous case shows that the simulated inference step is

$$(\mathcal{E} \uplus \{s \approx t\}, \mathcal{R}, \mathcal{C}) \vdash_{\mathsf{KBtt}} (\mathcal{E}, \mathcal{R} \cup \{t \to s\}, \mathcal{C} \cup \{t \to s\})$$

  iii. Finally, if $p' \notin R_{lr} \cup R_{rl}$ then process p' was not affected in this inference step, so $p = p'$ and we have $E[n, p] = E[n', p']$, $R[n, p] = R[n', p']$ and $C[n, p] = C[n', p']$. The projection of the considered MKBtt inference to process p' is thus an identity step.

In all remaining cases $p = p'$ holds as no process splitting occurs.

– If deduce adds a node $\langle s : t, \varnothing, \varnothing, R \cap R', \varnothing, \varnothing\rangle$ then for all $p \in R \cap R'$ both $\ell_1 \to r_1$ and $\ell_2 \to r_2$ occur in $R[N, p]$. Hence deduce can also be applied in KBtt, and indeed $E[N', p] = E[N, p] \cup \{s \approx t\}$. For all $p \notin R \cap R'$ the inference corresponds to an identity step.

– Whenever delete removes some node $\langle s : s, \varnothing, \varnothing, E, \varnothing, \varnothing\rangle$ then $s \approx s \in E[N, p]$ for all $p \in E$, and hence delete also applies in KBtt. For all $p \notin E$ an identity step is obtained.

– Next, assume rewrite was used. For every process $p \notin (R_0 \cup R_1 \cup E) \cap R$ an identity step is obtained. Otherwise, three cases can be distinguished which are distinct due to the node condition.

  i. If $p \in R_0 \cap R$ then $R[N, p]$ contains rules $s \to t$ and $\ell \to r$ such that $t \to_{\ell \to r} u$. Hence compose can be applied to replace $s \to t$ by $s \to u$, which is modelled in MKBtt by moving p from the rewrite label of a node with data $s : t$ to a node with data $s : u$.

ii. If $p \in E \cap R$ there is an equation $s \simeq t$ in $E[N, p]$ and a rule $\ell \to r$ in $R[N, p]$ such that $t \to_{\ell \to r} u$. Thus simplify can turn $s \simeq t$ into $s \simeq u$, and indeed $s \simeq u$ instead of $s \simeq t$ occurs in $E[N', p]$.

iii. If $p \in R_1 \cap R$ then there are rules $\ell \to r$ and $t \to s$ in $R[N, p]$ such that the latter can be collapsed into an equation $s \approx u$. Hence $s \approx u$ belongs to $E[N', p]$ and $t \to s$ is not in $R[N', p]$.

– If gc was applied the step obviously corresponds to an identity step on the level of KBtt for every process $p \in \mathcal{P}(N')$, and the same holds for subsume.

Finally, for every inference rule (besides gc and subsume) the non-emptiness requirement for the set of affected labels ensures that the strict part $\vdash$ holds for at least one $p' \in \mathcal{P}(N')$. □

Lemma 4.29. *Assume for a* KBtt *inference step* $(\mathcal{E}, \mathcal{R}, \mathcal{C}) \vdash (\mathcal{E}', \mathcal{R}', \mathcal{C}')$ *there exist a node set* N *and a process* p *such that* $\mathcal{E} = E[N, p]$, $\mathcal{R} = R[N, p]$ *and* $\mathcal{C} = C[N, p]$. *Then there is some inference step* $N \vdash N'$ *with split set* S *and a process* $p' \in \mathcal{P}(N')$ *such that* $p = pred_S(p')$, $\mathcal{E}' = E[N', p']$, $\mathcal{R}' = R[N', p']$ *and* $\mathcal{C}' = C[N', p']$.

Proof. In the following case analysis on the applied KBtt rule, $(*)$ refers to the proof obligations $\mathcal{E}' = E[N', p']$, $\mathcal{R}' = R[N', p']$, and $\mathcal{C}' = C[N', p']$.

– Assume orient was applied to replace some equation $s \simeq t \in \mathcal{E}$ by the rule $s \to t \in \mathcal{R}'$. Then there must be a node $n = \langle s : t, R_0, R_1, E, C_0, C_1 \rangle$ in N such that $p \in E$ and $\mathcal{C} \cup \{s \to t\}$ terminates. We distinguish two further cases. If $\mathcal{C} \cup \{t \to s\}$ terminates as well, we set $S = \{p\}$. For $R_{lr} = \{p0\}$ and $R_{rl} = \{p1\}$ an application of orient yields

$$\begin{aligned} N' = & split_{\{p\}}(N \setminus \{n\}) \cup \\ & \{\langle s : t, R_0 \cup \{p0\}, R_1 \cup \{p1\}, E \setminus \{p\}, C_0 \cup \{p0\}, C_1 \cup \{p1\} \rangle\} \end{aligned}$$

For $p' = p0$ we have $p = pred_S(p')$, and $(*)$ is satisfied. If $\mathcal{C}[N, p] \cup \{t \to s\}$ does not terminate, we apply orient with $S = \varnothing$ and $R_{lr} = \{p\}$ to obtain

$$N' = (N \setminus \{n\}) \cup \{\langle s : t, R_0 \cup \{p\}, R_1, E \setminus \{p\}, C_0 \cup \{p\}, C_1 \rangle\}$$

Thus we have $p' = p$ which trivially satisfies $p = pred_S(p')$, so $(*)$ holds.

In all remaining cases we can set $p' = p$ since no process splitting occurs.

– In the case where deduce generates $s \approx t$ from an overlap involving rules $\ell_1 \to r_1$ and $\ell_2 \to r_2$, there are nodes $\langle \ell_1 : r_1, R, \dots \rangle$ and $\langle \ell_2 : r_2, R', \dots \rangle$ in N such that $p \in R \cap R'$. Applying deduce in MKBtt thus yields

$$N' = N \cup \{\langle s : t, \varnothing, \varnothing, R \cap R', \varnothing, \varnothing \rangle\}$$

such that $(*)$ is satisfied.

– If delete removes some equation $s \approx s$ from $\mathcal{E}$ then N must contain a node $n = \langle s : s, R_0, R_1, E \uplus \{p\}, C_0, C_1\rangle$. Since the equation $s \approx s$ cannot be oriented into a terminating rule, the sets R_0, R_1, C_0 and C_1 must be empty. Thus n can be removed by delete in MKBtt.

– If simplify reduces an equation $s \simeq t$ to $s \simeq u$ using a rule $\ell \to r$, there are nodes $n = \langle s : t, R_0, R_1, E, C_0, C_1\rangle$ and $\langle \ell : r, R, \dots\rangle$ in N such that $p \in E \cap R$. We can therefore use rewrite to infer

$$N' = (N \setminus \{n\}) \cup \{\langle s : t, R_0 \setminus R, R_1 \setminus R, E \setminus R, C_0, C_1\rangle\} \cup \{\langle s : u, R_0 \cap R, \varnothing, (E \cup R_1) \cap R, \varnothing, \varnothing\rangle\}$$

Since $p \in E \cap R$, $(*)$ holds.

– If compose rewrites $s \to t$ to $s \to u$ using a rule $\ell \to r$ then N contains nodes $n = \langle s : t, R_0, R_1, E, C_0, C_1\rangle$ and $\langle \ell : r, R, \dots\rangle$ such that $p \in R_0 \cap R$. Thus rewrite applies and $(*)$ is satisfied since we obtain

$$N' = (N \setminus \{n\}) \cup \{\langle s : t, R_0 \setminus R, R_1 \setminus R, E \setminus R, C_0, C_1\rangle\} \cup \{\langle s : u, R_0 \cap R, \varnothing, (E \cup R_1) \cap R, \varnothing, \varnothing\rangle\}$$

– Finally, assume collapse is applied to turn a rule $t \to s$ into an equation $u \approx s$ using $\ell \to r$. Then N contains nodes $n = \langle s : t, R_0, R_1, E, C_0, C_1\rangle$ and $\langle \ell : r, R, \dots\rangle$ such that p occurs in $R_1 \cap R$. To satisfy $(*)$ we can thus apply rewrite to obtain

$$N' = (N \setminus \{n\}) \cup \{\langle s : t, R_0 \setminus R, R_1 \setminus R, E \setminus R, C_0, C_1\rangle\} \cup \{\langle s : u, R_0 \cap R, \varnothing, (E \cup R_1) \cap R, \varnothing, \varnothing\rangle\}$$ □

Definition 4.30. Consider an MKBtt run γ of the form $N_0 \vdash N_1 \vdash \cdots \vdash N_k$ where $N_0 = N_{\mathcal{E}}$ and let $p \in \mathcal{P}(N_k)$. We inductively define the sequence $p_0, \dots, p_k$ of *ancestors* of p by setting $p_k = p$ and $p_i = \mathit{pred}_{S_i}(p_{i+1})$ for $0 \leqslant i < k$, where S_i is the split set of the step $N_i \vdash N_{i+1}$.

We thus obtain the following corollary from Lemma 4.28:

Corollary 4.31. *Consider an* MKBtt *run* $\gamma\colon N_0 \vdash N_1 \vdash \cdots \vdash N_k$ *with* $p \in \mathcal{P}(N_k)$ *having ancestors* $p_0, \dots, p_k$. *Let* $\mathcal{E}_i$, $\mathcal{R}_i$ *and* $\mathcal{C}_i$ *denote* $E[N_i, p_i]$, $R[N_i, p_i]$ *and* $C[N_i, p_i]$, *respectively. Then the sequence*

$$\gamma_p\colon (\mathcal{E}_0, \mathcal{R}_0, \mathcal{C}_0) \vdash^{=} (\mathcal{E}_1, \mathcal{R}_1, \mathcal{C}_1) \vdash^{=} \cdots \vdash^{=} (\mathcal{E}_k, \mathcal{R}_k, \mathcal{C}_k)$$

is a valid KBtt *run, called the* projection *of* γ *to* p. □

Using projections, the definitions of success, failure and fairness given for KBtt can be naturally extended to MKBtt.

Definition 4.32. A finite MKBtt run γ of the form $N_0 \vdash^* N$

– is *fair* if γ_p is fair and nonfailing for some process $p \in \mathcal{P}(N)$,

– *succeeds* if $E[N,p] = \varnothing$ and $R[N,p]$ is convergent for some process $p \in \mathcal{P}(N)$, and

– *fails* if γ_p fails for all processes $p \in \mathcal{P}(N)$.

As the simulation of KBtt with MKBtt is sound (Lemma 4.28) and complete (Lemma 4.29), it is straightforward to establish correctness using the corresponding results for KBtt. We call an MKBtt run $\gamma\colon N_0 \vdash^* N$ *simplifying* if the resulting system $R[N,p]$ is reduced whenever γ succeeds for some process p.

Correctness Theorem 4.33. *Let $N_\mathcal{E}$ be the initial node set for a set of equations $\mathcal{E}$ and let γ be a finite* MKBtt *run of the form $N_\mathcal{E} \vdash^* N$ which is nonfailing and fair for some $p \in \mathcal{P}(N)$. Then $R[N,p]$ is convergent for $\mathcal{E}$.*

Proof. According to Corollary 4.31 there is a corresponding fair and nonfailing KBtt run $\gamma_p : (\mathcal{E}, \varnothing, \varnothing) \vdash^* (\varnothing, R[N,p], C[N,p])$. Since γ_p is finite, $R[N,p]$ is convergent for $\mathcal{E}$ by Theorem 4.16. □

Example 4.34. We illustrate an MKBtt run on the system CGE_2 (see Example 4.17). An MKBtt run starts with the initial node set

$$\langle \mathsf{e} \cdot x : x, \varnothing, \varnothing, \{\epsilon\}, \varnothing, \varnothing \rangle \tag{1}$$
$$\langle \mathsf{i}(x) \cdot x : \mathsf{e}, \varnothing, \varnothing, \{\epsilon\}, \varnothing, \varnothing \rangle \tag{2}$$
$$\langle x \cdot (y \cdot z) : (x \cdot y) \cdot z, \varnothing, \varnothing, \{\epsilon\}, \varnothing, \varnothing \rangle \tag{3}$$
$$\langle \mathsf{f}(x \cdot y) : \mathsf{f}(x) \cdot \mathsf{f}(y), \varnothing, \varnothing, \{\epsilon\}, \varnothing, \varnothing \rangle \tag{4}$$
$$\langle \mathsf{g}(x \cdot y) : \mathsf{g}(x) \cdot \mathsf{g}(y), \varnothing, \varnothing, \{\epsilon\}, \varnothing, \varnothing \rangle \tag{5}$$
$$\langle \mathsf{f}(x) \cdot \mathsf{g}(y) : \mathsf{g}(y) \cdot \mathsf{f}(x), \varnothing, \varnothing, \{\epsilon\}, \varnothing, \varnothing \rangle \tag{6}$$

When applying orient to nodes (1) and (2), only the direction from left to right yields valid and terminating rewrite rules. For node (3), both orientations are possible such that process ϵ is split into 0 and 1. These three nodes are thus modified as follows:

$$\langle \mathsf{e} \cdot x : x, \{0,1\}, \varnothing, \varnothing, \{0,1\}, \varnothing \rangle \tag{1}$$
$$\langle \mathsf{i}(x) \cdot x : \mathsf{e}, \{0,1\}, \varnothing, \varnothing, \{0,1\}, \varnothing \rangle \tag{2}$$
$$\langle x \cdot (y \cdot z) : (x \cdot y) \cdot z, \{0\}, \{1\}, \varnothing, \{0\}, \{1\} \rangle \tag{3}$$

Nodes (4) and (5) can be oriented in both directions, independent of the orientation of associativity. Now the current node set contains eight processes (constraint labels are omitted for the sake of readability; at this point they coincide with the respective rewrite labels):

$$\langle \mathsf{e} \cdot x : x, \{000, \ldots, 111\}, \varnothing, \varnothing, \ldots \rangle \tag{1}$$
$$\langle \mathsf{i}(x) \cdot x : \mathsf{e}, \{000, \ldots, 111\}, \varnothing, \varnothing, \ldots \rangle \tag{2}$$
$$\langle x \cdot (y \cdot z) : (x \cdot y) \cdot z, \{000, 001, 010, 011\}, \{100, 101, 110, 111\}, \varnothing, \ldots \rangle \tag{3}$$
$$\langle \mathsf{f}(x \cdot y) : \mathsf{f}(x) \cdot \mathsf{f}(y), \{000, 001, 100, 101\}, \{010, 011, 110, 111\}, \varnothing, \ldots \rangle \tag{4}$$
$$\langle \mathsf{g}(x \cdot y) : \mathsf{g}(x) \cdot \mathsf{g}(y), \{000, 010, 100, 110\}, \{001, 011, 101, 111\}, \varnothing, \ldots \rangle \tag{5}$$
$$\langle \mathsf{f}(x) \cdot \mathsf{g}(y) : \mathsf{g}(y) \cdot \mathsf{f}(x), \varnothing, \varnothing, \{000, \ldots, 111\}, \ldots \rangle \tag{6}$$

We abbreviate $\{000, 001, 010, 011\}$ to P_0 and $\{100, 101, 110, 111\}$ to P_1. The overlap $\mathsf{e} \cdot y \leftarrow (\mathsf{i}(x) \cdot x) \cdot y \rightarrow \mathsf{i}(x) \cdot (x \cdot y)$ between nodes (2) and (3) allows to deduce the additional node

$$\langle \mathsf{i}(x) \cdot (x \cdot y) : \mathsf{e} \cdot y, \varnothing, \varnothing, P_1, \varnothing, \varnothing \rangle \tag{7}$$

A rewrite step with node (1) simplifies this node to

$$\langle \mathsf{i}(x) \cdot (x \cdot y) : \mathsf{e} \cdot y, \varnothing, \varnothing, \varnothing, \varnothing, \varnothing \rangle \tag{7}$$

and adds

$$\langle \mathsf{i}(x) \cdot (x \cdot y) : y, \varnothing, \varnothing, P_1, \varnothing, \varnothing \rangle \tag{8}$$

The former is removed by gc and the latter is oriented to

$$\langle \mathsf{i}(x) \cdot (x \cdot y) : y, P_1, \varnothing, \varnothing, P_1, \varnothing \rangle \tag{8}$$

In a similar way, for processes in P_0 the overlap $(x \cdot \mathsf{i}(y)) \cdot y \leftarrow x \cdot (\mathsf{i}(y) \cdot y) \rightarrow x \cdot \mathsf{e}$ between (3) and (2) yields a node

$$\langle (x \cdot \mathsf{i}(y)) \cdot y : x \cdot \mathsf{e}, P_0, \varnothing, \varnothing, P_0, \varnothing \rangle \tag{9}$$

Additionally, there are critical peaks $x \cdot y \leftarrow x \cdot (\mathsf{e} \cdot y) \rightarrow (x \cdot \mathsf{e}) \cdot y$ between nodes (1) and (3), $\mathsf{e} \cdot x \leftarrow (\mathsf{i}(\mathsf{i}(x)) \cdot \mathsf{i}(x)) \cdot x \rightarrow \mathsf{i}(\mathsf{i}(x)) \cdot \mathsf{e}$ between nodes (2) and (9), and $\mathsf{i}(\mathsf{i}(x)) \cdot \mathsf{e} \leftarrow \mathsf{i}(\mathsf{i}(x)) \cdot (\mathsf{i}(x) \cdot x) \rightarrow x$ between nodes (2) and (8). Orienting the ensuing nodes yields

$$\langle (x \cdot \mathsf{e}) \cdot y : x \cdot y, P_0, \varnothing, \varnothing, P_0, \varnothing \rangle \tag{10}$$
$$\langle \mathsf{i}(\mathsf{i}(x)) \cdot \mathsf{e} : \mathsf{e} \cdot x, P_0, \varnothing, \varnothing, P_0, \varnothing \rangle \tag{11}$$
$$\langle \mathsf{i}(\mathsf{i}(x)) \cdot \mathsf{e} : x, P_1, \varnothing, \varnothing, P_1, \varnothing \rangle \tag{12}$$

Applying rewrite with (1) to node (11) creates a node with the same data as (12) also for processes in P_0, such that a subsume step results in the updated node

$$\langle \mathsf{i}(\mathsf{i}(x)) \cdot \mathsf{e} : x, P_1 \cup P_0, \varnothing, \varnothing, P_1, \varnothing \rangle \tag{12}$$

Now the peak $x \cdot y \leftarrow (\mathsf{i}(\mathsf{i}(x)) \cdot \mathsf{e}) \cdot y \rightarrow \mathsf{i}(\mathsf{i}(x)) \cdot y$ between (12) and (10) adds

$$\langle \mathsf{i}(\mathsf{i}(x)) \cdot y : x \cdot y, P_0, \varnothing, \varnothing, P_0, \varnothing \rangle \tag{13}$$

after a subsequent orient step. At this point overlaps between (13) and (12) and (13) and (2) trigger the creation of nodes that are oriented as

$$\langle x \cdot \mathsf{e} : x, P_0, \varnothing, \varnothing, P_0, \varnothing \rangle \tag{14}$$
$$\langle x \cdot \mathsf{i}(x) : \mathsf{e}, P_0, \varnothing, \varnothing, P_0, \varnothing \rangle \tag{15}$$

We obtain the modified node

$$\langle (x \cdot \mathsf{i}(y)) \cdot y : x \cdot \mathsf{e}, \varnothing, \varnothing, \varnothing, P_0, \varnothing \rangle \tag{9}$$

when using node (14) in a rewrite step, together with a new node with data $(x \cdot \mathsf{i}(y)) \cdot y : x$, which is oriented as

$$\langle (x \cdot \mathsf{i}(y)) \cdot y : x, P_0, \varnothing, \varnothing, P_0, \varnothing \rangle \tag{16}$$

Node (14) can also be used in rewrite steps to modify (10) and (12) to

$$\langle (x \cdot \mathsf{e}) \cdot y : x \cdot y, \varnothing, \varnothing, \varnothing, P_0, \varnothing \rangle \tag{10}$$

and

$$\langle \mathsf{i}(\mathsf{i}(x)) \cdot \mathsf{e} : x, P_1, \varnothing, \varnothing, P_0, \varnothing \rangle \tag{12}$$

while adding

$$\langle x \cdot y : x \cdot y, \varnothing, \varnothing, P_0, \varnothing, \varnothing \rangle \tag{17}$$

and

$$\langle \mathsf{i}(\mathsf{i}(x)) : x, \varnothing, \varnothing, P_0, \varnothing, \varnothing \rangle \tag{18}$$

to the current node set. The latter is oriented into

$$\langle \mathsf{i}(\mathsf{i}(x)) : x, P_0, \varnothing, \varnothing, P_0, \varnothing \rangle \tag{18}$$

while node (17) is subject to a delete inference. The overlaps $\mathsf{i}(\mathsf{e}) \leftarrow \mathsf{i}(\mathsf{e}) \cdot \mathsf{e} \rightarrow \mathsf{e}$ between (14) and (2) and $x \cdot \mathsf{e} \leftarrow x \cdot (y \cdot \mathsf{i}(y)) \rightarrow (x \cdot y) \cdot \mathsf{i}(y)$ between (15) and (3) add

$$\langle \mathsf{i}(\mathsf{e}) : \mathsf{e}, P_0, \varnothing, \varnothing, P_0, \varnothing \rangle \tag{19}$$

and

$$\langle (x \cdot y) \cdot \mathsf{i}(y) : x, P_0, \varnothing, \varnothing, P_0, \varnothing \rangle \tag{20}$$

to the node set (in the latter case, after rewrite using (14) simplifies $x \cdot \mathsf{e}$ to x).

To make a long story short, we will only sketch the remainder of the run. After some additional deduce steps, the last node concerning plain group theory

$$\langle \mathsf{i}(x \cdot y) : \mathsf{i}(y) \cdot \mathsf{i}(x), \varnothing, \varnothing, P_1 \cup P_0, \varnothing, \varnothing \rangle$$

is derived, and can again be oriented in both directions, resulting in a split of all current processes. To complete the theory of homomorphisms, nodes with data $\mathsf{f}(x) \cdot (\mathsf{f}(y) \cdot z) : \mathsf{f}(x \cdot y) \cdot z$, $\mathsf{f}(\mathsf{e}) : \mathsf{e}$, and $\mathsf{f}(\mathsf{i}(x)) : \mathsf{i}(\mathsf{f}(x))$ and similar ones for g are derived. The last kind of nodes gives again rise to process splits. It remains to orient node (6) and consider the critical pair $\mathsf{f}(x) \cdot (\mathsf{g}(y) \cdot z) \leftarrow \rtimes \rightarrow \mathsf{g}(y) \cdot (\mathsf{f}(x) \cdot z)$ before e.g. process 011110 succeeds after joining all remaining critical pairs. We obtain the same convergent TRS as in Example 4.17.

The sequence of orientations gives rise to a process tree, where every branching point corresponds to a process split in an orient step. Part of the process tree developed during the described completion run is sketched in Figure 4.5.

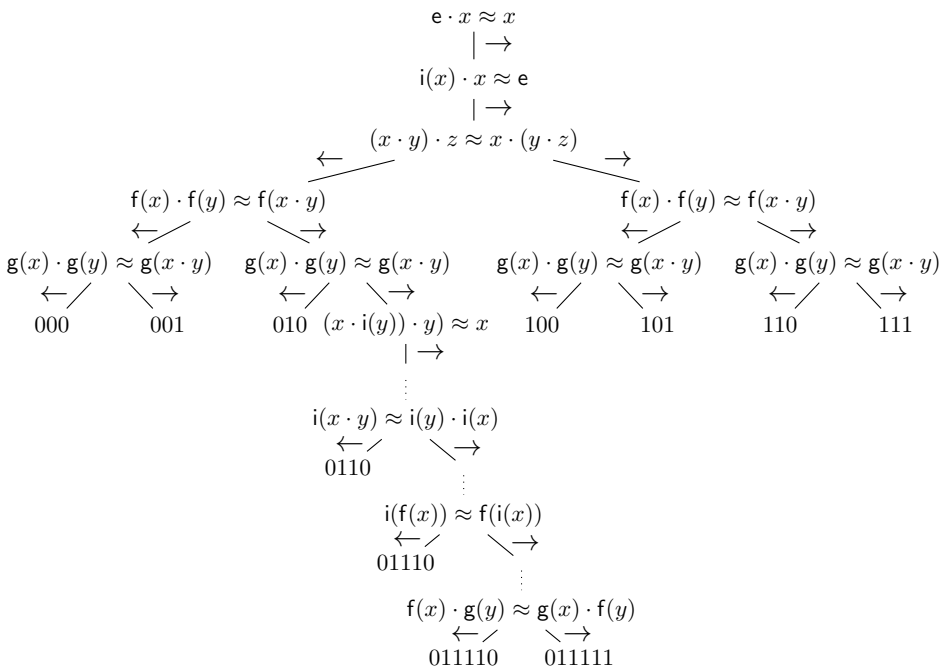


Figure 4.5: Part of a CGE_2 process tree with all branching points leading to process 011110.

More on Completeness

Theorem 3.13 states the completeness of KB in the following sense: If a set of equations $\mathcal{E}$ admits an equivalent finite convergent rewrite system $\mathcal{R}$, any fair KB run will produce an equivalent finite convergent system if a reduction order compatible with $\mathcal{R}$ is used, provided the run does not fail. The following example shows that MKBtt might even fail if one uses a termination tool $\mathbb{T}$ that can prove the termination of $\mathcal{R}$.

Example 4.35. The convergent rewrite system $\mathcal{R}$ consisting of the rules

$$\mathsf{f}(\mathsf{h}(x,y)) \to \mathsf{f}(\mathsf{i}(x,x)) \qquad \mathsf{h}(\mathsf{a},\mathsf{a}) \to \mathsf{c}$$
$$\mathsf{g}(\mathsf{i}(x,y)) \to \mathsf{g}(\mathsf{h}(x,x)) \qquad \mathsf{i}(\mathsf{a},\mathsf{a}) \to \mathsf{c}$$

is derived from the input equalities $\mathcal{E}$

$$\mathsf{f}(\mathsf{h}(x,y)) \approx \mathsf{f}(\mathsf{i}(x,x)) \qquad \mathsf{h}(\mathsf{a},\mathsf{a}) \approx \mathsf{c} \qquad \mathsf{h}(\mathsf{a},\mathsf{a}) \approx \mathsf{i}(\mathsf{a},\mathsf{a})$$
$$\mathsf{g}(\mathsf{i}(x,y)) \approx \mathsf{g}(\mathsf{h}(x,x)) \qquad \mathsf{i}(\mathsf{a},\mathsf{a}) \approx \mathsf{c}$$

in any fair run of standard completion that uses the reduction order $\to_{\mathcal{R}}^{+}$. The system $\mathcal{R}$ is easily shown to be terminating with a matrix interpretation of dimension 2; e.g. the termination tool $\mathsf{TTT2}$ using the strategy `matrix -ib 2 -d 2 -direct` immediately outputs a termination proof. However, if a KBtt run uses $\mathsf{TTT2}$ with this strategy and starts by orienting $\mathsf{h}(\mathsf{a},\mathsf{a}) \approx \mathsf{i}(\mathsf{a},\mathsf{a})$ then no matter which orientation is chosen, one of the equations in the leftmost

$$\textsf{orient}\quad \frac{N \uplus \{\langle s : t, R_0, R_1, E, C_0, C_1\rangle\}}{\mathit{split}'(E_{lr}, E_{rl}, N) \cup \{\langle s : t, R_0 \cup R_{lr}, R_1 \cup R_{rl}, E', C_0 \cup R_{lr}, C_1 \cup R_{rl}\rangle\}}$$

if $E_{lr}, E_{rl} \subseteq E$, $E' = E \setminus (E_{lr} \cup E_{rl}) \cup \{p- \mid p \in E_{lr} \cup E_{rl}\}$, E_{lr} is the set of all processes $p \in E$ such that $\mathbb{T} \vDash C_p(N) \cup \{s \to t\}$, E_{rl} is the set of all processes p such that $\mathbb{T} \vDash C_p(N) \cup \{t \to s\}$, $R_{lr} = \{p0 \mid p \in E_{lr}\}$ and $R_{rl} = \{p1 \mid p \in E_{rl}\}$, and $E_{lr} \cup E_{rl} \neq \varnothing$

Figure 4.6: The orient rule in $\mathsf{MKBtt_c}$.

column remains unorientable. Similarly, if MKBtt starts by applying orient to $\mathsf{h}(\mathsf{a},\mathsf{a}) \approx \mathsf{i}(\mathsf{a},\mathsf{a})$ then process ϵ gets split into 0 and 1. But in subsequent steps neither process can orient both of the equations in the leftmost column, so the run fails.

This example shows that the order in which nodes are processed has considerable influence: orienting nodes too early can prevent KBtt and MKBtt from producing a convergent system even if a successful run exists. Nevertheless, completeness in this sense can be partially obtained in a slightly modified version of MKBtt which we will refer to as $\mathsf{MKBtt_c}$. In contrast to the previous version, a process can now also keep an equation unoriented. For this purpose, processes are now viewed as strings in $(0 + 1 + -)^*$. We write $\mathbb{T} \vDash \mathcal{R}$ if the *termination tool* $\mathbb{T}$ can verify termination of the rewrite system $\mathcal{R}$. The orient rule in $\mathsf{MKBtt_c}$ is given in Figure 4.6. Here, $\mathit{split}'(E_{lr}, E_{rl}, N)$ replaces every occurrence of a process $p \in E_{lr} \cap E_{rl}$ in a node of N by $\{p-, p0, p1\}$, every occurrence of $p \in E_{lr} \setminus E_{rl}$ by $\{p-, p0\}$ and every occurrence of $p \in E_{rl} \setminus E_{lr}$ by $\{p-, p1\}$. The notion of a split set in MKBtt is replaced by *split tuple*, which refers to the pair of process sets (E_{lr}, E_{rl}). For all inference steps that use a different rule than orient, the split tuple is $(\varnothing, \varnothing)$.

Example 4.36. Assume $\mathsf{MKBtt_c}$ is run on the input equalities from Example 4.35 and starts by orienting $\mathsf{h}(\mathsf{a},\mathsf{a}) \approx \mathsf{i}(\mathsf{a},\mathsf{a})$. Then the resulting node is $\langle \mathsf{h}(\mathsf{a},\mathsf{a}) : \mathsf{i}(\mathsf{a},\mathsf{a}), \{0\}, \{1\}, \{-\}, \{0\}, \{1\}\rangle$. In contrast to MKBtt, a descendant of process $-$ can deliver a convergent system.

To obtain a completeness result for $\mathsf{MKBtt_c}$, we require a stronger notion of fairness which requires to equally advance all processes at some point.

Definition 4.37. Consider an equational proof P, a run $N_0 \vdash N_1 \vdash N_2 \vdash \cdots$ with $p \in \mathcal{P}(N_k)$, and let $\succ$ denote $\to^+_{C[N_k,p]}$. Then p *eventually simplifies* P *starting from* N_k if

- there is a proof Q in $(E[N_k, p], R[N_k, p])$ such that $P \Rightarrow^{\succ}_{\mathsf{KB}} Q$, or
- all direct successors $q \in \mathcal{P}(N_{k+1})$ of p eventually simplify P starting from N_{k+1}.

Thus a run γ with process $p \in \mathcal{P}(N_k)$ eventually simplifies a proof P if all successors of p in γ allow for a smaller proof at some point.

Definition 4.38 (Strong Fairness)**.** A run $\gamma\colon N_0 \vdash N_1 \vdash N_2 \vdash \cdots$ is *strongly fair* if for every $k \geqslant 0$, $p \in N_k$, and equational proof P in $(E[N_k, p], R[N_k, p])$ which is not in normal form, the following conditions hold:

(i) If N_k admits a step $N_k \vdash N$ such that $p \in \mathcal{P}(N)$ and there is an equational proof Q in $(E[N, p], R[N, p])$ satisfying $P \Rightarrow^{\succ}_{\mathsf{KB}} Q$, then p eventually simplifies P starting from N_k.

(ii) If there is an orient step $N_k \vdash N$ applied to node n such that N contains a successor p' of p and there is an equational proof Q in $(E[N, p'], R[N, p'])$ satisfying $P \Rightarrow^{\succ}_{\mathsf{KB}} Q$, then every successor q of p either performed an orient step on n and got extended by $-$ in this step, or eventually simplifies P from N_k.

Here $\succ$ denotes the reduction order $\rightarrow^{+}_{C[N_k,p]}$.

Intuitively, a strongly fair run requires all processes to simplify an equational proof if this simplification can be done without process splits (case (i)). Moreover, if an orient step on, say, a node with data $s : t$ allows for a simplification then all processes except the one that does not orient $s : t$ are required to perform this step (case (ii)). A sufficient condition for a run to be strongly fair is that all processes are advanced using a breadth-first strategy.

A termination tool $\mathbb{T}$ *covers* some reduction order $\succ$ if for any rewrite system $\mathcal{R}$ that is compatible with $\succ$, $\mathbb{T} \vDash \mathcal{R}$ holds.

Lemma 4.39. *Consider an* $\mathsf{MKBtt_c}$ *run* $\gamma\colon N_0 \vdash N_1 \vdash N_2 \vdash \cdots$ *which employs a termination tool* $\mathbb{T}$ *covering some reduction order* $\succ$*.*

i. For every node set N_k *there exists a process* p_k *such that* $C[N_k, p_k] \subseteq \succ$ *and the sequence* $(E[N_k, p_k], R[N_k, p_k])_{k\geqslant 0}$ *is a valid* KB *run* γ_p *using* $\succ$*.*

ii. If γ *is strongly fair then* γ_p *is fair.*

Proof.

i. We construct the process sequence $(p_k)_{k\geqslant 0}$ inductively such that

$$\begin{aligned}(E[N_k, p_k], R[N_k, p_k], C[N_k, p_k])\\ \vdash^{=} (E[N_{k+1}, p_{k+1}], R[N_{k+1}, p_{k+1}], C[N_{k+1}, p_{k+1}])\end{aligned} \tag{$*$}$$

is a valid KBtt inference step and $C[N_k, p_k] \subseteq \succ$.

We start by setting $p_0 = \epsilon$. Now consider an inference step $N_k \vdash N_{k+1}$ with split tuple (S_0, S_1). If $p_k \notin S_0 \cup S_1$ then we take $p_{k+1} = p_k$. By a straightforward adaptation of Lemma 4.28 to $\mathsf{MKBtt_c}$ a corresponding KBtt (or empty) step $(*)$ is possible, and $C[N_k, p_k] \subseteq \succ$ follows from the induction hypothesis. Otherwise, we must have $p_k \in E$ for an inference step orienting a term pair $s : t$ (adopting the notation used in Figure 4.6). If $s \succ t$ then $\mathbb{T} \vDash C[N_k, p_k] \cup \{s \to t\}$ as $\mathbb{T}$ covers $\succ$. In this case we set $p_{k+1} = p_k 0$. Due to the side condition of orient, $p_k \in S_0$ and hence $p_{k+1} \in \mathcal{P}(N_{k+1})$. Again $(*)$ is a KBtt step and by the choice of p_{k+1} also

$C[N_{k+1}, p_{k+1}] \subseteq \succ$ holds. The argument for the case $t \succ s$ is symmetric. If s and t are incomparable in $\succ$, we may choose $p_{k+1} = p_k-$. Then $(*)$ is an equality step and $C[N_{k+1}, p_{k+1}] \subseteq \succ$ follows from the induction hypothesis.

As the constructed sequence $(E[N_k, p_k], R[N_k, p_k], C[N_k, p_k])_{k\geqslant 0}$ constitutes a KBtt run which satisfies $C[N_k, p_k] \subseteq \succ$ for all $k \geqslant 0$, there is also a valid KB run $(E[N_k, p_k], R[N_k, p_k])_{k\geqslant 0}$ which uses $\succ$ as reduction order.

ii. Let $\mathcal{E}_\omega$ and $\mathcal{R}_\omega$ denote the persistent sets of γ_p. Suppose P is a proof in $(\mathcal{E}_\omega, \mathcal{R}_\omega)$ which is not a rewrite proof and there exists an inference $(\mathcal{E}_\omega, \mathcal{R}_\omega) \vdash (\mathcal{E}, \mathcal{R})$ such that $(\mathcal{E}, \mathcal{R})$ admits a proof Q satisfying $P \Rightarrow^{\succ}_{\mathsf{KB}} Q$. Then there must be a node set N_j in γ such that $(E[N_j, p_j], R[N_j, p_j])$ contains all equations and rules that are used in P together with those used when simplifying P to Q. By adapting Lemma 4.29 to $\mathsf{MKBtt_c}$, it follows that there is an inference step $N_j \vdash N$ such that $\mathcal{E}' = E[N, p']$, $\mathcal{R}' = R[N, p']$, and $C[N, p'] \subseteq \succ$ holds for some successor p' of p_j, and $(\mathcal{E}', \mathcal{R}')$ admits proof Q.

We distinguish two cases. If $(E[N_j, p_j], R[N_j, p_j]) \vdash (\mathcal{E}', \mathcal{R}')$ and thus $N_j \vdash N$ does not apply orient then no process splitting occurs and $p_j \in \mathcal{P}(N)$. By strong fairness, p_j eventually simplifies P. In particular, some successor p_m in the process sequence $(p_k)_{k\geqslant 0}$ with $m \geqslant j$ has to provide a proof Q' in $(E[N_m, p_m], R[N_m, p_m])$ such that $P \Rightarrow^{\succ}_{\mathsf{KB}} Q'$. Therefore also γ_p allows for this simplified proof.

Now suppose $(E[N_j, p_j], R[N_j, p_j]) \vdash (\mathcal{E}', \mathcal{R}')$ applied orient to some equation $s \approx t$ and $s \succ t$ holds. By construction of the sequence $(p_k)_{k\geqslant 0}$ no successor of p_j can have obtained $-$ as part of its label when orienting a node with data $s : t$. Hence, according to strong fairness all successors of p_j have to eventually simplify P. So some p_m in $(p_k)_{k\geqslant 0}$ with $m \geqslant j$ has to provide a proof Q' in $(E[N_m, p_m], R[N_m, p_m])$ with $P \Rightarrow^{\succ}_{\mathsf{KB}} Q'$. Again this proof is reflected in γ_p, which proves fairness of this KB run. □

The following completeness result shows that an $\mathsf{MKBtt_c}$ run employing a sufficiently powerful termination prover can produce any convergent system which is derivable in a KB run.

Theorem 4.40. *Consider a finite canonical rewrite system $\mathcal{R}$ which can be constructed from $\mathcal{E}$ in a fair KB run using $\succ$. If $\mathbb{T}$ covers $\succ$ then any strongly fair and simplifying $\mathsf{MKBtt_c}$ run $N_0 \vdash N_1 \vdash N_2 \vdash \cdots$ which uses $\mathbb{T}$ and does not have a failing process develops some process $p \in \mathcal{P}(N_n)$ which satisfies $E[N_n, p] = \varnothing$ and $R[N_n, p] = \mathcal{R}$ (up to renaming variables).*

Proof. According to Lemma 4.39 there is a sequence of processes $(p_k)_{k\geqslant 0}$ such that $(E[N_k, p_k], R[N_k, p_k])_{k\geqslant 0}$ is a fair KB run using $\succ$. By repeating the following argument of [11, Theorem 3.9], we will see that this run succeeds with system $\mathcal{R}$. Each rule $\ell \to r$ in $\mathcal{R}$ is a theorem in $\mathcal{E}$ and therefore will have a persisting rewrite proof after a finite number of steps in every fair and unfailing run. Let $\mathcal{R}' \subseteq \bigcup_i R[N_i, p_i]$ be the set of rules required for proofs of all rules in

deduce	$\dfrac{N}{N \cup \{\langle s : t, \varnothing, \varnothing, E, \varnothing, \varnothing\rangle\}}$
	if there exist nodes $\langle \ell_1 : r_1, R, \dots\rangle, \langle \ell_2 : r_2, R', \dots\rangle \in N$ and an overlap o involving rules $\ell_1 \to r_1$ and $\ell_2 \to r_2$ that gives rise to a critical pair $s \leftarrow \rtimes \rightarrow t$ such that $E = \mathrm{CPC}_m(o, R \cap R', N) \neq \varnothing$

Figure 4.7: The deduce inference rule using a critical pair criterion.

$\mathcal{R}$. Both $\mathcal{R}$ and $\mathcal{R}'$ are contained in $\succ$. Hence all these proofs must be of the form $\ell \to^*_{\mathcal{R}'} r$: Suppose r was reducible in $\mathcal{R}'$ to a term r' such that $r \succ r'$. Then there must also be a proof $r \leftrightarrow^*_{\mathcal{R}} r'$ as $\mathcal{R}$ is a convergent presentation of the theory. But $r \succ r'$ implies that r is reducible in $\mathcal{R}$, contradicting the assumption that $\mathcal{R}$ is canonical.

Thus $\to_{\mathcal{R}} \subseteq \to^+_{\mathcal{R}'}$ holds. As $\mathcal{R}$ and $\mathcal{R}'$ have the same equational theory, $\mathcal{R}'$ must be convergent and hence canonical since it was constructed by a simplifying run. Thus $\mathcal{R}$ and $\mathcal{R}'$ have to be equal, because the canonical rewrite system compatible with a given reduction order is unique (up to variable renaming) [80]. □

Note that even if $\mathbb{T}$ covers $\succ$, an MKBtt$_c$ run might still fail if equations are selected in an unfortunate order (cf. Example 3.18).

4.3.1 Critical Pair Criteria

By filtering out equational consequences that can be ignored without compromising completeness, critical pair criteria allow to reduce the number of nodes in an MKBtt run. It is not difficult to incorporate the well-known criteria for standard completion (see Section 3.2.2) to the setting of MKBtt. All those criteria require to check whether an overlapped term can be reduced in a certain way other than indicated by the overlap itself. Since MKBtt allows to check reducibility for multiple processes at once, the redundancy checks required for the respective multi-completion criteria can even be shared among multiple processes.

Definition 4.41. Given a KB critical pair criterion CPC, the corresponding MKBtt *critical pair criterion* CPC_m maps an overlap o with associated critical pair $s \leftarrow \times \rightarrow t$, a set of processes E and a node set N to a process set $\mathrm{CPC}_m(o, E, N) = E'$ such that $E' \subseteq E$ and $s \approx t \in E[N, p] \setminus \mathrm{CPC}(E[N, p], R[N, p])$ for all $p \in E'$.

Intuitively, the set E' contains all processes in E for which the critical pair derived from o is not superfluous. Thus, in the deduce rule for MKBtt the equation label of the new node is filtered by the criterion as shown in Figure 4.7.

Consider a finite MKBtt run γ of the form $N_0 \vdash^* N_k$ and a process $p \in N_k$. Let p_i denote the ancestor of p in N_i and let $\succ$ denote the reduction order $\to^+_{C[N_k,p]}$. Then we call γ *fair with respect to* CPC_m *and* p if the following condition holds: Whenever a node set N_i gives rise to an overlap o with critical pair $s \leftarrow \rtimes \rightarrow t$

as described in Figure 4.7 and $p_i \in E \setminus \mathrm{CPC}_m(o, E, N_i)$ then there exists a proof Q in some $(E[N_j, p_j], R[N_j, p_j])$ for $j > 0$ such that $s \approx t \Rightarrow^{\succ}_{\mathsf{KB}} Q$ holds. The run γ is *fair with respect to* CPC_m if it is fair with respect to CPC_m and some process $p \in \mathcal{P}(N_k)$. An MKBtt *critical pair criterion* CPC_m is *correct* if every finite nonfailing run γ that is fair with respect to CPC_m is also fair in the sense of Definition 4.32.

Lemma 4.42. *Every* MKBtt critical pair criterion CPC_m *obtained from a correct criterion* CPC *is correct.*

Proof. Let γ be a finite nonfailing run of the form $N_0 \vdash^* N_k$ which is fair with respect to CPC_m and some process $p \in \mathcal{P}(N_k)$, and let p_i denote the ancestor of p in N_i. Assume a critical overlap o where $p_i \in E \setminus \mathrm{CPC}_m(o, E, N_i)$ for some ancestor p_i of p. By definition there exists a proof Q in some $(E[N_j, p_j], R[N_j, p_j])$ such that $s \approx t \Rightarrow^{\succ}_{\mathsf{KB}} Q$. Hence the projected run γ_p is fair with respect to CPC and by correctness of CPC it is also fair. Thus γ is fair as well. □

The following example illustrates the use of the *primality* criterion PCP_m and the *connectedness* criterion WCP_m in MKBtt.

Example 4.43. An MKBtt run on CGE_2 encounters the nodes

$$\langle \mathsf{e} \cdot x : x, P_0 \cup P_1, \ldots \rangle \tag{1}$$
$$\langle \mathsf{i}(x) \cdot x : \mathsf{e}, P_0 \cup P_1, \ldots \rangle \tag{2}$$
$$\langle x \cdot \mathsf{e} : x, P_0, \ldots \rangle \tag{14}$$
$$\langle \mathsf{i}(\mathsf{e}) : \mathsf{e}, P_0, \ldots \rangle \tag{19}$$
$$\langle y \cdot \mathsf{i}(x \cdot y) : \mathsf{i}(x), P_0 \cup P_1, \ldots \rangle \tag{21}$$

The overlap $\langle y \cdot \mathsf{i}(x \cdot y) \to \mathsf{i}(x), \epsilon, \mathsf{e} \cdot x \to x \rangle$ creates the critical pair $\mathsf{i}(x) \leftarrow \rtimes \rightarrow \mathsf{i}(x \cdot \mathsf{e})$ for the process set $P_0 \cup P_1$. When PCP_m is applied, it is checked whether there exists a node which allows to reduce the term $u = \mathsf{e} \cdot \mathsf{i}(x \cdot \mathsf{e})$ at some position below the root. Since node (14) can reduce u at position 21, the critical pair is recognized as redundant for all processes in P_0 such that the deduced node is $\langle \mathsf{i}(x) : \mathsf{i}(x \cdot \mathsf{e}), \varnothing, \varnothing, P_1, \varnothing, \varnothing \rangle$.

Furthermore, the overlap $\langle \mathsf{i}(\mathsf{e}) \to \mathsf{e}, 1, \mathsf{i}(x) \cdot x \to \mathsf{e} \rangle$ between nodes (19) and (2) gives rise to the critical pair $\mathsf{e} \cdot \mathsf{e} \leftarrow \rtimes \rightarrow \mathsf{e}$ for the process set P_0. To reduce the term $\mathsf{i}(\mathsf{e}) \cdot \mathsf{e}$ also node (14) can be applied at the root position. While PCP_m is not applicable since the overlap position is below ϵ, WCP_m requires to check the critical pairs involved in the decomposition. Indeed, the critical peak $\mathsf{e} \leftarrow \mathsf{i}(\mathsf{e}) \cdot \mathsf{e} \rightarrow \mathsf{i}(\mathsf{e})$ between nodes (2) and (14) is already covered by node (19) and the peak involving nodes (14) and (19) can be ignored since it is just a variable overlap. Hence this critical pair can be ignored.

4.3.2 Isomorphisms

In an implementation of MKBtt, performance is significantly affected by the number of simulated processes. On some input problems, runs exhibit similar process pairs which have the same probability of success.

Example 4.44. A run on CGE_2 may generate a node set N with process p where $E[N,p]$ consists of the equations

$$(x \cdot y) \cdot z \approx x \cdot (y \cdot z) \qquad \mathsf{f}(\mathsf{e}) \approx \mathsf{e}$$
$$\mathsf{g}(x) \cdot \mathsf{f}(y) \approx \mathsf{f}(y) \cdot \mathsf{g}(x) \qquad \mathsf{g}(\mathsf{e}) \approx \mathsf{e}$$

and $R[N,p] = C[N,p]$ consists of the rewrite rules

$$\mathsf{e} \cdot x \to x \qquad \mathsf{f}(x \cdot y) \to \mathsf{f}(x) \cdot \mathsf{f}(y)$$
$$\mathsf{i}(x) \cdot x \to \mathsf{e} \qquad \mathsf{g}(x \cdot y) \to \mathsf{g}(x) \cdot \mathsf{g}(y)$$

If an inference step $N \vdash N'$ applies orient to the equation $\mathsf{g}(x) \cdot \mathsf{f}(y) \approx \mathsf{f}(y) \cdot \mathsf{g}(x)$, the process p is split as both orientations are possible. But the states of the emerging child processes $p0$ and $p1$ are the same up to interchanging f and g. Hence further deductions of these processes will be symmetric.

Such similarities between processes are captured by the more general concept of *isomorphisms*.

Definition 4.45. A bijection $\theta\colon \mathcal{T}(\mathcal{F},\mathcal{V}) \to \mathcal{T}(\mathcal{F},\mathcal{V})$ extends to an *isomorphism* between rewrite systems $\mathcal{R}$ and $\mathcal{R}'$ if $\mathcal{R}' = \{\theta(\ell) \to \theta(r) \mid \ell \to r \in \mathcal{R}\}$ such that $s \to_{\mathcal{R}} t$ if and only if $\theta(s) \to_{\mathcal{R}'} \theta(t)$ for all terms s and t. Two sets of equations $\mathcal{E}$ and $\mathcal{E}'$ are isomorphic with respect to θ if $\mathcal{E}' = \{\theta(u) \approx \theta(v) \mid u \approx v \in \mathcal{E}\}$ and for all terms s and t, $s \leftrightarrow_{\mathcal{E}} t$ if and only if $\theta(s) \leftrightarrow_{\mathcal{E}'} \theta(t)$. These concepts are expressed by writing $\mathcal{R} \cong_\theta \mathcal{R}'$ and $\mathcal{E} \cong_\theta \mathcal{E}'$, respectively. Two MKBtt processes p and q are isomorphic in a node set N if there exists some isomorphism θ such that $E[N,p] \cong_\theta E[N,q]$, $R[N,p] \cong_\theta R[N,q]$ and $C[N,p] \cong_\theta C[N,q]$.

Lemma 4.46. *Let N_p and N_q be node sets containing processes p and q such that*

$$(E[N_p,p], R[N_p,p], C[N_p,p]) \cong_\theta (E[N_q,q], R[N_q,q], C[N_q,q])$$

If there is a step $N_p \vdash N_p'$ such that p is the predecessor of $p' \in \mathcal{P}(N_p')$ then there is also an inference step $N_q \vdash^{=} N_q'$ and a process $q' \in \mathcal{P}(N_q')$ such that q is the predecessor of q' and

$$(E[N_p',p'], R[N_p',p'], C[N_p',p']) \cong_\theta (E[N_q',q'], R[N_q',q'], C[N_q',q'])$$

Proof. If $(E[N_p,p], R[N_p,p], C[N_p,p]) = (E[N_p',p], R[N_p',p], C[N_p',p])$ and $p \in \mathcal{P}(N_p')$ then p was not affected by the step $N_p \vdash N_p'$ so we can set $N_q' = N_q$. Otherwise a *mirror step* $N_q \vdash N_q'$ using the same inference rule can model the step for q. More precisely, the mirror step is defined by case distinction on the rule applied in $N_p \vdash N_p'$.

- Assume an orient step turned a node $n = \langle s : t, R_0, R_1, E, C_0, C_1\rangle$ into $\langle s : t, R_0 \cup R_{lr}, R_1 \cup R_{rl}, E', C_0 \cup R_{lr}, C_1 \cup R_{rl}\rangle$ using split set S. Then a node n' with data $\theta(s) : \theta(t)$ has to occur in N_q since $E[N_q,q] \cong E[N_p,p]$ by assumption. Three further possibilities can be distinguished: Assume $p = p'$ and $p \in E_{lr} \setminus S$ because $C[N_p,p] \cup \{s \to t\}$ terminates, but

$C[N_p, p] \cup \{t \to s\}$ does not. Thus also $C[N_q, q] \cup \{\theta(s) \to \theta(t)\}$ terminates, but $C[N_q, q] \cup \{\theta(t) \to \theta(s)\}$ does not. So the mirror step $N_q \vdash N'_q$ can apply orient to node n' with $E_{lr} = R_{rl} = \{q\}$ and $E_{rl} = R_{rl} = \varnothing$. In the second case where $p = p'$ and $p \in E_{rl} \setminus S$, we reason symmetrically to the preceding case. For the final case, let $p \in S$. We orient n' to obtain the mirror step $N_q \vdash N'_q$. Because $C[N_p, p] \cong C[N_q, q]$ holds, both orientations terminate, so q gets split into $q0$ and $q1$ which are then isomorphic to $p0$ and $p1$, respectively.

All remaining inference rules do not split processes so $p = p'$ and thus $q = q'$.

- Assume delete was applied to a node $\langle s : s, \varnothing, \varnothing, E, \varnothing, \varnothing\rangle$ where $p \in E$. Then there must be a node n' of the form $\langle \theta(s) : \theta(s), \varnothing, \varnothing, E', \varnothing, \varnothing\rangle$ in N_q such that $q \in E'$, and $N_q \vdash N'_q$ will be a delete step removing n'.
- If deduce created a node with data $\langle s : t, \varnothing, \varnothing, R \cap R', \varnothing, \varnothing\rangle$ originating from an overlap involving nodes with terms $\ell_1 : r_1$ and $\ell_2 : r_2$, due to $R[N_p, p] \cong R[N_q, q]$, there must be nodes with data $\langle \theta(\ell_1) : \theta(r_1), R, \ldots\rangle$ and $\langle \theta(\ell_2) : \theta(r_2), R', \ldots\rangle$ in N_q which allow in a mirror step $N_q \vdash N'_q$ to deduce a node $\langle \theta(s) : \theta(t), \varnothing, \varnothing, R \cap R', \varnothing, \varnothing\rangle$.
- If rewrite was applied to a node $\langle s : t, R_0, R_1, E, C_0, C_1\rangle$ using a rule node $\langle \ell : r, R, \ldots\rangle$ to create $\langle s : u, R'_0, \varnothing, E', \varnothing, \varnothing\rangle$, then the mirror step $N_q \vdash N'_q$ can apply the same rule to nodes with data $\theta(s) : \theta(t)$ and $\theta(\ell) : \theta(r)$, which exist by assumption. Then q is removed from the rewrite or equation label of the node with data $\theta(s) : \theta(t)$, and occurs now in a node with data $\theta(s) : \theta(u)$. □

Theorem 4.47. *Let N_i be a set of nodes containing isomorphic processes $p_i, q_i \in \mathcal{P}(N_i)$. Assume there exists an* MKBtt *completion run γ of the form $N_i \vdash^* N_k$ and a process $p_k \in \mathcal{P}(N_k)$ such that p_i is the ancestor of p_k in N_i, and the projected run γ_{p_k} is fair and successful. Then there is also a fair deduction γ' of the form $N_i \vdash^* N'_m$ with a process $q_m \in \mathcal{P}(N'_m)$ such that q_i is an ancestor of q_m, and also γ'_{q_m} is fair and successful.*

Proof. Induction on the length of the run $\gamma\colon N_i \vdash^* N_k$ shows that there also exists a run $\gamma'\colon N_i = N'_i \vdash^* N'_m$ with a process $q_m \in N'_m$ such that the tuples $(E[N_k, p_k], R[N_k, p_k], C[N_k, p_k])$ and $(E[N'_m, q_m], R[N'_m, q_m], C[N'_m, q_m])$ are isomorphic. For the case where $k = i$ the processes p_i and q_i are by assumption isomorphic via some mapping θ, so we set $m = i$. For the induction step we consider a run $N_i \vdash^* N_k \vdash N_{k+1}$ where $p_k \in \mathcal{P}(N_k)$ is the predecessor of $p_{k+1} \in \mathcal{P}(N_{k+1})$. By the induction hypothesis there is a sequence $N_i = N'_i \vdash^* N'_m$ with $q_m \in N'_m$ such that $(E[N_k, p_k], R[N_k, p_k], C[N_k, p_k])$ and $(E[N'_m, q_m], R[N'_m, q_m], C[N'_m, q_m])$ are isomorphic with respect to θ. By Lemma 4.46, the last step $N_k \vdash N_{k+1}$ can be mirrored by $N'_m \vdash^= N'$ such that some process q' in N' is isomorphic to p_{k+1} in N_{k+1}. If $N'_m = N'$ and thus $q' = q_m$ then the claim clearly holds. Otherwise we set $N'_{m+1} = N'$, and by Lemma 4.46 there is some process $q_{m+1} \in \mathcal{P}(N'_{m+1})$ such that $(E[N_{k+1}, p_{k+1}], R[N_{k+1}, p_{k+1}], C[N_{k+1}, p_{k+1}])$ is again with respect to θ isomorphic to $(E[N'_{m+1}, q_{m+1}], R[N'_{m+1}, q_{m+1}], C[N'_{m+1}, q_{m+1}])$.

Hence given $\gamma\colon N_i \vdash^* N_k$ with $p_k \in \mathcal{P}(N_k)$ there is a run $\gamma'\colon N_i \vdash^* N'_m$ with $q_m \in \mathcal{P}(N'_m)$ such that $(E[N_k, p_k], R[N_k, p_k], C[N_k, p_k])$ is isomorphic to $(E[N'_m, q_m], R[N'_m, q_m], C[N'_m, q_m])$. Success of p_k in N_k implies $E[N_k, p_k] = \varnothing$ and thus also $E[N'_m, q_m] = \varnothing$. By the definition of isomorphisms γ'_{q_m} also inherits fairness from γ_k. □

According to Theorem 4.47, if two isomorphic processes are detected in the current node set $\mathcal{N}$ then one of them can be deleted from all nodes in $\mathcal{N}$ without compromising completeness. The following two concrete shapes of symmetries are examples for isomorphisms. *Renamings* swap function symbols as in Example 4.44, where $p0$ and $p1$ are isomorphic under the mapping θ that exchanges f and g. *Argument permutations* associate with every function symbol f of arity n a permutation π_f of the set $\{1, \ldots, n\}$. Then the mapping on terms defined by $\theta(x) = x$ and $\theta(f(t_1, \ldots, t_n)) = f(\theta(t_{\pi_f(1)}), \ldots, \theta(t_{\pi_f(n)}))$ also induces an isomorphism. For example, when completing SK90-3.02 [95] a process with state

$$(x + y) + z \approx x + (y + z) \qquad \begin{array}{c} \mathsf{f}(\mathsf{f}(x)) \to x \\ \mathsf{f}(x + y) \to \mathsf{f}(x) + \mathsf{f}(y) \end{array}$$

has to orient the associativity axiom. Both orientations preserve termination, but the two child processes emerging from a process split are isomorphic under the argument permutation $\pi_+ = (1\,2)$.

Chapter 5

Ordered Completion Systems

As already emphasized, the reduction order used plays a crucial role with respect to the success of a completion run: a wrong choice can easily lead to failure. Multi-completion and completion with termination tools constitute two approaches to tackle this problem. However, for many equational theories no convergent system exists, and for some theories a convergent TRS exists but cannot be constructed by a completion procedure. The following example illustrates the latter case.

Example 5.1 ([6,11])**.** When standard completion is run on the set of equations

$$1 \cdot (-x + x) \approx 0 \qquad 1 \cdot (x + -x) \approx x + -x \qquad -x + x \approx y + -y$$

the first two equations can be oriented, but this does not trigger any critical pairs and independent of the employed order the run will fail as the last equality is unorientable and persistent. Nevertheless, there exists a convergent TRS for this theory:

$$-x + x \to 0 \qquad x + -x \to 0 \qquad 1 \cdot 0 \to 0$$

This shortcoming is addressed by *unfailing completion*, first introduced by Bachmair, Dershowitz and Plaisted [6, 13]. Given a set of equations, it generally constructs only a *ground convergent* rewrite system, but it always succeeds (although this may require an infinite derivation). A ground convergent system does not allow to decide the equational theory, but is sufficient for many practical applications such as validity checks of ground equalities. Underlying completion-based theorem proving systems such as Waldmeister [72], it has become a well-established calculus in automated reasoning.

Section 5.1 describes ordered completion (as unfailing completion is more commonly called nowadays) in its original setting. Section 5.2 outlines ordered multi-completion while Section 5.3 describes ordered completion with termination tools. These two approaches are combined to ordered multi-completion with termination tools in Section 5.4.

5.1 Ordered Completion

We first require some additional definitions connected to ground convergence and ordered completion. A reduction order $\succ$ is *complete* for a theory $\mathcal{E}_0$ if for every pair of ground terms u, v such that $u \leftrightarrow^*_{\mathcal{E}_0} v$ either $u \succ v$ or

$v \succ u$ holds, and it is *completable* if it can be extended to a complete order for the theory. Note that a reduction order that is total on ground terms is complete for any theory, as is for example the case for KBO and LPO with total precedences. Also, any reduction order induced by a polynomial interpretation can be extended to a total order, e.g. by lexicographically combining it with an LPO. For an ES $\mathcal{E}$ and a reduction order $\succ$, the TRS $\mathcal{E}_\succ$ consists of all *orientable instances* of $\mathcal{E}$, i.e., all rules $s\sigma \to t\sigma$ such that $s \simeq t \in \mathcal{E}$ and $s\sigma \succ t\sigma$. In the sequel we assume $\succ$ to be a completable reduction order for the theory associated with the initial equalities $\mathcal{E}_0$. Ordered completion aims at constructing a ground convergent system for the theory. Here a pair $(\mathcal{E}, \mathcal{R})$ of a set of equations $\mathcal{E}$ and a set of rewrite rules $\mathcal{R}$ is *ground convergent* with respect to $\succ$ if $\mathcal{R} \cup \mathcal{E}_\succ$ is ground-convergent.

In contrast to standard completion, ordered completion considers *extended* critical pairs among equations in $\mathcal{E}$ and rewrite rules in $\mathcal{R}$.

Definition 5.2. Let $\mathcal{E}$ be a set of equations and $\mathcal{R}$ be a set of rewrite rules. An *extended overlap* with respect to $\succ$ is a triple $\langle \ell_1 \approx r_1, p, \ell_2 \approx r_2 \rangle_\sigma$ such that $\ell_1 \simeq r_1, \ell_2 \simeq r_2 \in \mathcal{E} \cup \mathcal{R}$ are equations without common variables, $p \in \mathcal{P}\mathrm{os}_\mathcal{F}(\ell_2)$, the terms ℓ_1 and $\ell_2|_p$ are unifiable with most general unifier σ, and $r_i\sigma \not\succ \ell_i\sigma$ for both $i \in \{1, 2\}$. Moreover, $\ell_2\sigma[r_1\sigma]_p \leftarrow \rtimes \rightarrow r_2\sigma$ constitutes an *extended critical pair*. Again we write $s \leftarrow \times \rightarrow t$ to denote $s \leftarrow \rtimes \rightarrow t$ or $t \leftarrow \rtimes \rightarrow s$. The set of all extended critical pairs among $\mathcal{E} \cup \mathcal{R}$ with respect to $\succ$ is denoted by $\mathrm{CP}_\succ(\mathcal{E} \cup \mathcal{R})$.

Note that in contrast to standard critical pairs, a root overlap of two equations that are variants of each other need not be trivial. For example, overlapping $\mathsf{f}(x) \approx \mathsf{g}(y)$ on itself yields the non-trivial extended critical pair $\mathsf{g}(x) \leftarrow \rtimes \rightarrow \mathsf{g}(y)$.

As a second difference to standard completion, ordered completion also allows the use of equation instances for rewriting. For a set of equations $\mathcal{E}$, by $\mathcal{E}_\succ$ we denote its (possibly infinite) set of *orientable instances*, i.e., all rules $u\sigma \to v\sigma$ such that $u \simeq v$ in $\mathcal{E}$ and $u\sigma \succ v\sigma$.

By oKB we will denote all inference rules KB of standard completion except for deduce, together with the additional rules given in Figure 5.1 (although the deduce rule from KB is subsumed by deduce_2). As for KB, an inference sequence $(\mathcal{E}_0, \mathcal{R}_0) \vdash (\mathcal{E}_1, \mathcal{R}_1) \vdash (\mathcal{E}_2, \mathcal{R}_2) \vdash \cdots$ is called a *run* with *persistent* equalities $\mathcal{E}_\omega = \bigcup_i \bigcap_{j>i} \mathcal{E}_j$ and rules $\mathcal{R}_\omega = \bigcup_i \bigcap_{j>i} \mathcal{R}_j$. For the sake of readability we assume $\mathcal{R}_0 = \varnothing$ although all of the following results generalize to the case where $\mathcal{R}_0$ is non-empty, provided that $\mathcal{R}_0$ is contained in the reduction order $\succ$. The proof reduction relation for standard completion is adapted to cover rewrite steps using rules in $\mathcal{E}_\succ$.

Definition 5.3 ([13])**.** Let $\mathcal{E}$ be a set of equations, $\mathcal{R}$ a rewrite system and $\succ$ a reduction order containing $\mathcal{R}$. The *cost* of an equational proof step is then

deduce_2	$\dfrac{\mathcal{E},\mathcal{R}}{\mathcal{E}\cup\{s\approx t\},\mathcal{R}}$	if $s \leftarrow\!\!\rtimes\!\!\rightarrow t \in \mathrm{CP}_\succ(\mathcal{E}\cup\mathcal{R})$
$\mathsf{simplify}_2$	$\dfrac{\mathcal{E}\uplus\{s\simeq t\},\mathcal{R}}{\mathcal{E}\cup\{s\simeq u\},\mathcal{R}}$	if $t \to_{\mathcal{E}_\succ} u$ using $\ell\sigma\to r\sigma$ for $\ell\simeq r\in\mathcal{E}$ such that $t \rhd \ell$
$\mathsf{compose}_2$	$\dfrac{\mathcal{E},\mathcal{R}\uplus\{s\to t\}}{\mathcal{E},\mathcal{R}\cup\{s\to u\}}$	if $t \to_{\mathcal{E}_\succ} u$
$\mathsf{collapse}_2$	$\dfrac{\mathcal{E},\mathcal{R}\uplus\{t\to s\}}{\mathcal{E}\cup\{u\approx s\},\mathcal{R}}$	if $t \to_{\mathcal{E}_\succ} u$ using $\ell\sigma\to r\sigma$ for $\ell\simeq r\in\mathcal{E}$ such that $t \rhd \ell$

Figure 5.1: Some inference rules of ordered completion (oKB).

defined as follows:

$$c(s \xleftrightarrow[u\approx v]{} t) = (\{s,t\},\bot,\bot,\bot) \qquad \text{if } u\simeq v\in\mathcal{E},\ u\not\succ v \text{ and } v\not\succ u$$
$$c(s \xleftrightarrow[\ell\to r]{p} t) = c(t \xleftrightarrow[r\leftarrow\ell]{p} s) = (\{s\}, s|_p, \ell, \{t\}) \qquad \text{if } \ell\to r\in\mathcal{R}$$
$$c(s \xleftrightarrow[\ell\to r]{p} t) = c(t \xleftrightarrow[r\leftarrow\ell]{p} s) = (\{s\}, s|_p, \ell, \{t,\top\}) \qquad \text{if } \ell\to r\in\mathcal{E}_\succ$$

To compare cost tuples we use the lexicographic combination of $\succ_{\mathsf{mul}}$, $\rhd$, $\trianglerighteq$ and $\succ_{\mathsf{mul}}$, where $\bot$ is minimal and $\top$ is maximal in the latter three orderings. Again the cost of an equational proof is the multiset consisting of the costs of its steps, and the proof order $\succcurlyeq_{\mathsf{oKB}}$ on equational proofs is the multiset extension of the order on proof step costs. Finally, let $P \Rightarrow^{\succ}_{\mathsf{oKB}} Q$ hold if and only if $P \succcurlyeq_{\mathsf{oKB}} Q$ and P and Q prove the same equation.

Lemma 5.4 ([13])**.** $\Rightarrow^{\succ}_{\mathsf{oKB}}$ *is a proof reduction relation.* □

We show that oKB runs constitute equational inference sequences according to Definition 3.1 (with the theory $\mathcal{T}$ being empty). For the sake of readability $\succcurlyeq$ abbreviates $\succcurlyeq_{\mathsf{oKB}}$ if the intended definition is clear from the context.

Lemma 5.5. *Any* oKB *run is an equational inference sequence with respect to* $\Rightarrow^{\succ}_{\mathsf{oKB}}$.

Proof. We have to show that every inference step can be modeled by expand and $\mathsf{contract}$ steps according to Definition 3.1. Every deduce and deduce_2 step is clearly an instance of expand. A delete inference obviously constitutes a $\mathsf{contract}$ step as $s \leftrightarrow^{\epsilon}_{s\approx s} s \Rightarrow^{\succ}_{\mathsf{oKB}} s$. Every other inference step can be viewed as an expansion adding the respective rule or equation, followed by a contraction. An orient inference adds $s\to t$ and subsequently removes $s\simeq t$, which is valid as $s \leftrightarrow_{s\approx t} t \Rightarrow^{\succ}_{\mathsf{oKB}} s \leftrightarrow_{s\to t} t$ because $\{(\{s\},s,s,\{t,\top\})\} \succcurlyeq \{(\{s\},s,s,\{t\})\}$.

An application of $\mathsf{simplify}$ can be viewed as an expansion adding $s\simeq u$ followed by a contraction step removing $s\simeq t$. The latter transforms the proof $P\colon s \leftrightarrow_{s\approx t} t$ into $Q\colon s \leftrightarrow_{s\approx u} u \,{}_{\ell\to r}\!\leftarrow t$, where $\ell\to r\in\mathcal{R}$. A case distinction

shows that this transformation always results in a smaller proof. If $s \succ t$ then $c(P) = \{(\{s\}, s, s, \{t, \top\})\}$. As $s \succ t$ and $t \succ u$ imply $s \succ u$ we have $c(Q) = \{(\{s\}, s, s, \{u, \top\}), (\{t\}, \ldots)\}$, and hence $P \Rightarrow^{\succ}_{\mathsf{oKB}} Q$. Otherwise, if $t \succ s$ then $c(P) = \{(\{t\}, t, s, \{s, \top\})\}$ and if neither $s \succ t$ nor $t \succ s$ then $c(P) = \{(\{s, t\}, \ldots)\}$. In both cases $c(Q)$ involves $(\{t\}, t, \ell, \{u\})$ and one of $(\{s, u\}, \ldots)$, $(\{s\}, \ldots)$, or $(\{u\}, \ldots)$. Because of $t \succ u$ and $t \trianglerighteq \ell$ this results in a decrease in both cases. In case of a $\mathsf{simplify}_2$ step the reasoning is very similar, except that the cost of the last proof step amounts to $(\{t\}, t, \ell, \{u, \top\})$ and we have $t \triangleright \ell$ by the side condition of $\mathsf{simplify}_2$.

A $\mathsf{compose}$ step adds $s \to u$ and performs a contraction removing $s \to t$, which is justified because $s \leftrightarrow_{s \to t} t$ has cost $\{(\{s\}, s, s, \{t\})\}$ while the cost of $s \leftrightarrow_{s \to u} u \mathrel{{}_{\mathcal{R}}\!\leftarrow} t$ only amounts to $\{(\{s\}, s, s, \{u\}), (\{t\}, \ldots)\}$ and both $s \succ t$ and $t \succ u$ hold. In case of $\mathsf{compose}_2$ all affected components of proof costs are the same, so the argument is similar.

Finally, a $\mathsf{collapse}$ step constitutes an expansion adding $u \approx s$ followed by a contraction removing $t \to s$. The latter yields a decrease because $t \leftrightarrow_{t \to s} s$ with cost $\{(\{t\}, t, t, \ldots)\}$ is replaced by $t \to^{p}_{\ell \to r} u \leftrightarrow_{u \approx s} s$ for some $\ell \to r$ and $p \in \mathcal{P}\mathrm{os}(t)$ in $\mathcal{R}$ such that $t \triangleright \ell$, so that the latter proof's cost consists of $(\{t\}, t, \ell, \ldots)$ and one of $(\{s, u\}, \ldots)$, $(\{s\}, \ldots)$ or $(\{u\}, \ldots)$. The decrease results from $t \triangleright \ell$ and $t \succ s, u$. In case of $\mathsf{collapse}_2$ all affected components of proof costs are the same, so the argument is similar. □

As in the case for standard completion, the correctness proof of ordered completion crucially relies on a critical pair lemma. We recall the original result [13] and add a statement about proof simplification with respect to $\Rightarrow^{\succ}_{\mathsf{oKB}}$.

Extended Critical Pair Lemma 5.6. *Let $\succ$ be a complete reduction order for $\mathcal{E} \cup \mathcal{R}$ such that $\mathcal{R} \subseteq \succ$, and consider a peak $s \mathrel{{}_{\mathcal{E}_\succ \cup \mathcal{R}}\!\leftarrow} u \to_{\mathcal{E}_\succ \cup \mathcal{R}} t$ of ground terms s, t, and u. Then $s \downarrow_{\mathcal{E}_\succ \cup \mathcal{R}} t$ unless $s = C[s'\sigma]$ and $t = C[t'\sigma]$ for some extended critical pair $s' \leftarrow \times \to t' \in \mathrm{CP}_\succ(\mathcal{E} \cup \mathcal{R})$, context C and substitution σ.*

Proof. We consider a peak $s \mathrel{{}^{p,\tau}_{r_1 \approx \ell_1}\!\leftarrow} u \to^{q,\tau}_{\ell_2 \approx r_2} t$ such that $\ell_i \to r_i \in \mathcal{R}$ or $\ell_i\tau \to r_i\tau \in \mathcal{E}_\succ$ for both $i \in \{1, 2\}$. Assume there is an extended overlap $\langle \ell_1 \approx r_1, p, \ell_2 \approx r_2 \rangle_\rho$ and some substitution σ such that $s = C[s'\sigma]$ and $t = C[t'\sigma]$ for $s' = \ell_2\rho[r_1\rho]$ and $t' = r_2\rho$. Because of $\ell_i \to r_i \in \mathcal{R}$ or $\ell_i\tau \simeq r_i\tau \in \mathcal{E}_\succ$ we must have $r_i\rho \not\succ \ell_i\rho$ as $\succ$ is closed under substitutions, and therefore $s' \simeq t' \in \mathrm{CP}_\succ(\mathcal{E} \cup \mathcal{R})$. If the peak does not contain a proper overlap then the two rewrite steps are either performed in parallel, or they form a variable overlap. Both cases are handled in exactly the same way as in the Critical Pair Lemma 3.8. □

Corollary 5.7. *If $\succ$ is complete for $\mathcal{E}$ and P: $s \mathrel{{}_{\mathcal{E}_\succ \cup \mathcal{R}}\!\leftarrow} u \to_{\mathcal{E}_\succ \cup \mathcal{R}} t$ is a ground non-overlap then there is a proof Q such that $P \Rightarrow^{\succ}_{\mathsf{oKB}} Q$.*

Proof. By Lemma 5.6 there is a rewrite proof $Q\colon s \to^*_{\mathcal{E}_\succ \cup \mathcal{R}} \cdot \mathrel{{}^*_{\mathcal{E}_\succ \cup \mathcal{R}}\!\leftarrow} t$. While $c(P) = \{(\{u\}, \ldots), (\{u\}, \ldots)\}$ all tuples $(\{v\}, \ldots)$ in $c(Q)$ satisfy $s \succeq v$ or $t \succeq v$. Since $u \succ s, t$ we have $P \Rightarrow^{\succ}_{\mathsf{oKB}} Q$. □

In order to characterize runs that perform all required expansions, we use the following notion of fairness.

Definition 5.8 ([6])**.** Let $\succ$ be a reduction order that is extensible to a complete reduction order $>$ for $\mathcal{E}_0$. A run $(\mathcal{E}_0, \varnothing) \vdash (\mathcal{E}_1, \mathcal{R}_1) \vdash (\mathcal{E}_2, \mathcal{R}_2) \vdash \cdots$ using $\succ$ is *fair* if for any ground proof P in $\mathcal{E}_\omega \cup \mathcal{R}_\omega$ which is not a rewrite proof in $(\mathcal{E}_\omega)_> \cup \mathcal{R}$ there is a proof Q in $\bigcup_i \mathcal{E}_i \cup \mathcal{R}_i$ such that $P \Rightarrow^{\succ}_{\mathsf{oKB}} Q$.

In practice, fairness can be ensured by a simpler condition, which is also the notion originally used in [13].

Lemma 5.9 ([6])**.** *Let $\succ$ be completable for $\mathcal{E}_0$. If an* oKB *run $(\mathcal{E}_0, \varnothing) \vdash (\mathcal{E}_1, \mathcal{R}_1) \vdash (\mathcal{E}_2, \mathcal{R}_2) \vdash \cdots$ satisfies $\mathrm{CP}_\succ(\mathcal{E}_\omega \cup \mathcal{R}_\omega) \subseteq \bigcup_i \mathcal{E}_i$ then it is fair.*

Proof. Let $> \supseteq \succ$ be complete for $\mathcal{E}_0$. Assume P is a ground non-rewrite proof in $(\mathcal{E}_\omega)_> \cup \mathcal{R}_\omega$, that is, a proof containing a peak $P' \colon s \leftarrow u \rightarrow t$. According to the Extended Critical Pair Lemma 5.6, if this peak does not constitute a proper overlap then there exists a smaller proof. Otherwise, $s = C[\ell\sigma]$ and $t = C[r\sigma]$ for some extended critical pair $\ell \leftarrow \times \rightarrow r$ in $\mathrm{CP}_>(\mathcal{E}_\omega \cup \mathcal{R}_\omega)$. Note that we have $\mathrm{CP}_>(\mathcal{E}_\omega \cup \mathcal{R}_\omega) \subseteq \mathrm{CP}_\succ(\mathcal{E}_\omega \cup \mathcal{R}_\omega)$, so by assumption $\ell \simeq r$ occurs in some set $\mathcal{E}_i$. We have $c(P') = \{(\{u\}, \ldots), (\{u\}, \ldots)\}$, while the cost $c(s \leftrightarrow_{\ell \approx r} t)$ amounts to $\{(\{s\}, \ldots)\}$ or $\{(\{t\}, \ldots)\}$, so $u \succ s, t$ implies $P' \Rightarrow^{\succ}_{\mathsf{oKB}} s \leftrightarrow_{\ell \approx r} t$. By the Persistence Lemma 3.4 there is a proof Q' in the limit $(\mathcal{E}_\omega)_> \cup \mathcal{R}_\omega$ which satisfies $s \leftrightarrow_{\ell \approx r} t$ $(\Rightarrow^{\succ}_{\mathsf{oKB}})^= Q'$. Consequently, $P = P[P'] \Rightarrow^{\succ}_{\mathsf{oKB}} P[Q']$ holds as $\Rightarrow^{\succ}_{\mathsf{oKB}}$ is a proof reduction relation. □

It can be shown that fair runs always produce a desired outcome [6, 13]:

Correctness Theorem 5.10. *Let $\succ$ be completable for $\mathcal{E}_0$, and $(\mathcal{E}_0, \varnothing) \vdash (\mathcal{E}_1, \mathcal{R}_1) \vdash (\mathcal{E}_2, \mathcal{R}_2) \vdash \cdots$ be a fair* oKB *run using $\succ$. Then $(\mathcal{E}_\omega, \mathcal{R}_\omega)$ is ground convergent for $\mathcal{E}_0$ with respect to any complete extension of $\succ$.*

Proof. Let $> \supseteq \succ$ be a complete reduction order for the theory $\mathcal{E}_0$. We show that for every pair of ground terms $s \leftrightarrow^*_{\mathcal{E}_0} t$ there is a persisting rewrite proof in the limit. According to Lemma 5.5 any oKB run is an equational inference sequence. Hence by the Persistence Lemma 3.4 there exists a proof of $s \approx t$ in $(\mathcal{E}_\omega)_> \cup \mathcal{R}_\omega$. Let P be such a proof which is minimal with respect to $\Rightarrow^{\succ}_{\mathsf{oKB}}$. If P were not a rewrite proof then fairness would imply the existence of a smaller proof, contradicting minimality of P. Since oKB is an equational inference system by Lemma 5.5 the system $(\mathcal{E}_\omega, \mathcal{R}_\omega)$ has the same equational theory as $\mathcal{E}_0$, so by Lemma 3.2 it is ground convergent for $\mathcal{E}_0$. □

If a run considers all extended critical pairs and no persistent equations remain then the final rewrite system is even convergent. This is easily proven in the same way as for Correctness Theorem 3.12.

Theorem 5.11. *Let $(\mathcal{E}_0, \varnothing) \vdash (\mathcal{E}_1, \mathcal{R}_1) \vdash (\mathcal{E}_2, \mathcal{R}_2) \vdash \cdots$ be an* oKB *run such that $\mathrm{CP}(\mathcal{R}_\omega) \subseteq \bigcup_i \mathcal{E}_i$ and $\mathcal{E}_\omega = \varnothing$. Then $\mathcal{R}_\omega$ is convergent for $\mathcal{E}_0$.*

Proof. We show that for all terms s, t such that $s \leftrightarrow^*_{\mathcal{E}_0} t$ there is a persisting rewrite proof in the limit. By the Persistence Lemma 3.4 the TRS $\mathcal{R}_\omega$ admits a proof of any such equation $s \approx t$. Let P be a minimal such proof, and assume it is not a rewrite proof. Hence it must contain a peak $P' \colon s \leftarrow u \rightarrow t$. If

$s \downarrow_{\mathcal{R}_\omega} t$ would hold then a smaller proof would be possible by a similar argument as used in the proof of Corollary 3.9. This contradicts minimality of P. Therefore, by the Critical Pair Lemma 3.8 the proof P' must contain an instance of a critical pair, so $s = C[\ell\sigma]$ and $t = C[r\sigma]$ for some context C, substitution σ, and $\ell \leftarrow \times \rightarrow r$ in $\mathrm{CP}(\mathcal{R}_\omega)$. We have $c(P) = \{(\{u\}, \ldots), (\{u\}, \ldots)\}$ but $Q\colon s \leftrightarrow_{\ell \approx l} t$ amounts to $c(Q) = \{(\{s, t\}, \ldots)\}$, so $P' \Rightarrow^{\succ}_{\mathsf{oKB}} Q$. By the Persistence Lemma 3.4 there is a proof Q' in $\mathcal{R}_\omega$ such that $Q\ (\Rightarrow^{\succ}_{\mathsf{oKB}})^{=}\ Q'$. This entails $P[P'] \Rightarrow^{\succ}_{\mathsf{oKB}} P[Q']$, which again contradicts P being minimal. Since $\mathcal{R}_\omega$ has the same equational theory as $\mathcal{E}_0$ by Lemma 3.2 it is convergent for $\mathcal{E}_0$. □

An oKB completion procedure is *simplifying* if for all inputs $\mathcal{E}_0$ and $\succ$ the rewrite system $\mathcal{R}_\omega$ is reduced and all equations $u \simeq v$ in $\mathcal{E}_\omega$ are both unorientable with respect to $\succ$ and irreducible in $\mathcal{R}_\omega$.

Theorem 5.12 ([6,13])**.** *Let $\succ$ be completable for $\mathcal{E}$, and assume $\mathcal{R}$ is a canonical rewrite system for $\mathcal{E}$ such that $\mathcal{R} \subseteq \succ$. Then any fair and simplifying* oKB *run $(\mathcal{E}_0, \varnothing) \vdash (\mathcal{E}_1, \mathcal{R}_1) \vdash (\mathcal{E}_2, \mathcal{R}_2) \vdash \cdots$ using $\succ$ yields $\mathcal{E}_\omega = \varnothing$ and $\mathcal{R}_\omega = \mathcal{R}$ (modulo variable renaming).* □

Example 5.13 ([6,11])**.** Assume we run ordered completion using an LPO with precedence $\cdot > + > - > 1 > 0$ on the set of equations from Example 5.1:

$$1 \cdot (-x + x) \approx 0 \tag{1}$$
$$1 \cdot (x + -x) \approx x + -x \tag{2}$$
$$-x + x \approx y + -y \tag{3}$$

The first two equations can be oriented.

$$1 \cdot (-x + x) \to 0 \tag{1}$$
$$1 \cdot (x + -x) \to x + -x \tag{2}$$

The extended overlap $\langle (3), 2, (1) \rangle$ adds an equation which is oriented as

$$1 \cdot (y + -y) \to 0 \tag{4}$$

Now the extended overlap $\langle (2), \epsilon, (4) \rangle$ adds an equation which is oriented as

$$x + -x \to 0 \tag{5}$$

Rule (5) now simplifies the unorientable equation (3), and uses compose and collapse to reduce terms in (2) and (4). This adds equations that can be oriented as follows:

$$-x + x \to 0 \tag{6}$$
$$1 \cdot 0 \to 0 \tag{7}$$

Rule (6) collapses also rule (1). Since there are no equations left and no critical pairs among (5), (6), and (7), the run succeeds with the convergent TRS mentioned in Example 5.1:

$$-x + x \to 0 \qquad x + -x \to 0 \qquad 1 \cdot 0 \to 0$$

Example 5.14 ([44])**.** Consider the following set of equations describing an *entropic groupoid*, and choose LPO with empty precedence as reduction order.

$$(x \cdot y) \cdot (z \cdot w) \approx (x \cdot z) \cdot (y \cdot w) \tag{1}$$
$$(x \cdot y) \cdot x \approx x \tag{2}$$

Although an orient inference step can turn equation (2) into the rewrite rule

$$(x \cdot y) \cdot x \to x \tag{2}$$

standard completion fails as the permutative equation (1) remains unorientable. When applying ordered completion instead, a deduce_2 step from the extended overlap $\langle 1, \epsilon, 2\rangle$ yields the equation $((x \cdot y) \cdot x) \cdot (z \cdot y) \approx x \cdot y$, which can be simplified with rule (2) such that we obtain an equation that is oriented as

$$x \cdot (z \cdot y) \to x \cdot y \tag{3}$$

This rule can in turn be used to simplify (1) which results in the new equation

$$(x \cdot y) \cdot z \approx (x \cdot w) \cdot z \tag{4}$$

This equation gives rise to the overlap $\langle 2, 1, 4\rangle$. The resulting extended critical pair $((x \cdot y) \cdot z) \cdot w \leftarrow \rtimes \to x \cdot w$ can be oriented as an additional rewrite rule:

$$((x \cdot y) \cdot z) \cdot w \to x \cdot w \tag{5}$$

All further critical pairs among the rules (2), (3), (5) and equation (4) are either joinable or are simplified to instances of (4). Thus we derive the ground convergent system

$$\begin{array}{ll} (x \cdot y) \cdot z \approx (x \cdot w) \cdot z & (x \cdot y) \cdot x \to x \\ x \cdot (y \cdot z) \to x \cdot z & ((x \cdot y) \cdot z) \cdot w \to x \cdot w \end{array}$$

Note that in contrast to convergent TRSs, ground convergent systems compatible with a given reduction order need not be unique.

Example 5.15. Given a signature consisting of a unary symbol f and a constant a, both TRSs $\mathcal{R} = \{\mathsf{f}(x) \to x\}$ and $\mathcal{R}' = \{\mathsf{f}(\mathsf{f}(x)) \to \mathsf{f}(x), \mathsf{f}(\mathsf{a}) \to \mathsf{a}\}$ are ground convergent, reduced, compatible with LPO, and they induce the same conversion on ground terms.

5.1.1 Refutational Theorem Proving

Ordered completion can also be employed for equational theorem proving. For ground goals this was already observed in [13]. The case of non-ground goals basically relies on completeness of narrowing with (ground-)convergent rewrite systems [38]. In the context of completion-based theorem proving, respective results are e.g. discussed in [29, 84].

Let a *goal* be an equation $s \approx t$ in which all variables are existentially quantified (note that universally quantified variables can be replaced by Skolem constants [29]). Consider a set of initial equations $\mathcal{E}$ and a goal $s \approx t$, and let the binary symbol equal and the constants true and false be fresh. We write $\mathcal{E}^{s \approx t}$ for the set $\mathcal{E} \cup \{\mathsf{equal}(s,t) \approx \mathsf{false},\ \mathsf{equal}(x,x) \approx \mathsf{true}\}$.

Note that a reduction order $\succ$ which is complete for $\mathcal{E} \cup \{s \approx t\}$ can be extended to a complete reduction order $\succ_{\mathsf{c}}$ for $\mathcal{E}^{s \approx t}$: Let $\mathcal{F}$ denote the set of function symbols occurring in $\mathcal{E} \cup \{s \approx t\}$. Then we define $\succ_{\mathsf{c}}$ as an extension of $\succ$ such that (i) $\mathsf{equal}(s,t) \succ_{\mathsf{c}} u \succ_{\mathsf{c}} \mathsf{true} \succ_{\mathsf{c}} \mathsf{false}$ and (ii) $\mathsf{equal}(s,t) \succ_{\mathsf{c}} \mathsf{equal}(u,v)$ if and only if $\{s,t\} \succ_{\mathsf{mul}} \{u,v\}$, for all terms $s,t,u,v \in \mathcal{T}(\mathcal{F},\mathcal{V})$ (cf. [13]).[1]

In order to derive a correctness result we first state two auxiliary lemmas.

Lemma 5.16. *Consider a set of equations $\mathcal{E}$, a goal $s \approx t$, and an* oKB *run $\gamma\colon (\mathcal{E}^{s \approx t}, \varnothing) \vdash (\mathcal{E}_1, \mathcal{R}_1) \vdash (\mathcal{E}_2, \mathcal{R}_2) \vdash \cdots$ using $\succ_{\mathsf{c}}$ such that an equation $\mathsf{equal}(u,v) \simeq \mathsf{false}$ occurs in $\mathcal{E}_\alpha \cup \mathcal{R}_\alpha$ for some $\alpha > 0$. Then there is a substitution σ such that $s\sigma \leftrightarrow^*_{\mathcal{E}} u$ and $t\sigma \leftrightarrow^*_{\mathcal{E}} v$.*

Proof. A straightforward induction argument shows that all equations which occur in γ and involve the symbol equal need to have one of the forms $\mathsf{equal}(x,x) \simeq \mathsf{false}$ or $\mathsf{equal}(u,v) \simeq \mathsf{false}$ for $u,v \in \mathcal{T}(\mathcal{F},\mathcal{V})$, since equal is a fresh symbol. Moreover, all rewrite rules involving the symbol equal need to have the form $\mathsf{equal}(u,v) \to \mathsf{false}$ for $u,v \in \mathcal{T}(\mathcal{F},\mathcal{V})$ because false is minimal in the order.

We now apply transfinite induction on α to prove the claim. In the base case only $\mathsf{equal}(s,t) \approx \mathsf{false}$ occurs in $\mathcal{E}_0$, and the claim trivially holds for an empty substitution σ.

Next, assume $\mathsf{equal}(u,v) \simeq \mathsf{false} \in \mathcal{E}_{n+1} \cup \mathcal{R}_{n+1}$. If $\mathsf{equal}(u,v) \simeq \mathsf{false}$ also occurs in $\mathcal{E}_n \cup \mathcal{R}_n$ then the claim holds by the induction hypothesis. Otherwise, one of the following cases must apply.

- Suppose $\mathsf{equal}(u,v) \simeq \mathsf{false}$ is added by a $\mathsf{simplify}$ or $\mathsf{simplify}_2$ step from an equation $\mathsf{equal}(u',v') \simeq \mathsf{false} \in \mathcal{E}_n$, or by a $\mathsf{collapse}$ or $\mathsf{collapse}_2$ step from a rule $\mathsf{equal}(u',v') \to \mathsf{false} \in \mathcal{R}_n$. As equal is fresh the rewrite step must either take place in u' or in v'. Without loss of generality, assume that $u' \to_{\ell \to r} u$ using a rule $\ell \to r \in \mathcal{R}_n \cup \mathcal{E}_n^{\succ}$, and $v = v'$. By the induction hypothesis there is a substitution σ such that $s\sigma \leftrightarrow^*_{\mathcal{E}} u'$ and $t\sigma \leftrightarrow^*_{\mathcal{E}} v$. According to the Soundness Lemma 3.2 we have $\ell \leftrightarrow^*_{\mathcal{E}} r$, and hence $s\sigma \leftrightarrow^*_{\mathcal{E}} u' \leftrightarrow^*_{\mathcal{E}} u$.

- Otherwise $\mathsf{equal}(u,v) \simeq \mathsf{false}$ must have been added by a deduce_2 step. As noted above, the symbol equal can only occur in equations $\mathsf{equal}(x,x) \simeq \mathsf{false}$ or $\mathsf{equal}(u,v) \simeq \mathsf{false}$ for $u,v \in \mathcal{T}(\mathcal{F},\mathcal{V})$. Hence the respective overlap must have the form $\langle \ell \approx r, p, \mathsf{equal}(u',v') \approx \mathsf{false} \rangle_\tau$ such that $\ell \simeq r$, $\mathsf{equal}(u',v') \simeq \mathsf{false} \in \mathcal{E}_n \cup \mathcal{R}_n$ and $p > \epsilon$. Without loss of generality we assume that $p = 1q$, so the overlap corresponds to a peak

$$\mathsf{equal}(u'\tau[r\tau]_q, v'\tau) \leftrightarrow_{r \approx \ell} \mathsf{equal}(u'\tau, v'\tau) \leftrightarrow_{\mathsf{equal}(u',v') \approx \mathsf{false}} \mathsf{false}$$

1 Note that ill-sorted terms such as $\mathsf{equal}(\mathsf{equal}(\mathsf{a},\mathsf{a}),\mathsf{a})$ do not belong to the equational theory, hence they need not be taken into account by $\succ_{\mathsf{c}}$.

such that $u = u'\tau[r\tau]_q$ and $v = v'\tau$. By the induction hypothesis there is a substitution σ such that $s\sigma \leftrightarrow^*_{\mathcal{E}} u'$ and $t\sigma \leftrightarrow^*_{\mathcal{E}} v'$. According to the Soundness Lemma 3.2 we have $\ell \leftrightarrow^*_{\mathcal{E}} r$, and hence both $s\sigma\tau \leftrightarrow^*_{\mathcal{E}} u'\tau \leftrightarrow^*_{\mathcal{E}} u$ and $t\sigma\tau \leftrightarrow^*_{\mathcal{E}} v$ hold.

Finally, if $\mathsf{equal}(u, v) \simeq \mathsf{false} \in \mathcal{E}_\omega \cup \mathcal{R}_\omega$ then $\mathsf{equal}(u, v) \simeq \mathsf{false} \in \mathcal{E}_n \cup \mathcal{R}_n$ for some $n < \omega$, so the claim holds by the induction hypothesis. □

Theorem 5.17. *Let $\mathcal{E}$ be a set of equations and $s \approx t$ be a goal such that $\succ$ is completable for $\mathcal{E} \cup \{s \approx t\}$, and consider a fair* oKB *run $(\mathcal{E}^{s\approx t}, \varnothing) \vdash (\mathcal{E}_1, \mathcal{R}_1) \vdash (\mathcal{E}_2, \mathcal{R}_2) \vdash \cdots$ using $\succ_{\mathsf{c}}$. Then $\mathsf{true} \simeq \mathsf{false} \in \bigcup_i \mathcal{E}_i \cup \mathcal{R}_i$ if and only if there exists a substitution σ such that $s\sigma \leftrightarrow^*_{\mathcal{E}} t\sigma$.*

Proof. Assume there is a substitution σ such that $s\sigma \leftrightarrow^*_{\mathcal{E}} t\sigma$. Let σ' be ground such that $\sigma' = \sigma\tau$ for some substitution τ. According to the Correctness Theorem 5.10, the run under consideration produces a system $(\mathcal{E}_\omega, \mathcal{R}_\omega)$ which is ground convergent with respect to $\succ_{\mathsf{c}}$. The equation $\mathsf{true} \approx \mathsf{false}$ has the ground proof $\mathsf{true} \leftrightarrow \mathsf{equal}(s\sigma, s\sigma) \leftrightarrow^*_{\mathcal{E}} \mathsf{equal}(s\sigma, t\sigma) \leftrightarrow \mathsf{false}$ in $\mathcal{E}^{s\approx t}$. By the Persistence Lemma 3.4, $\mathsf{true} \approx \mathsf{false}$ also has a persistent minimal rewrite proof in some $(\mathcal{E}_n, \mathcal{R}_n)$. By the definition of $\succ_{\mathsf{c}}$ we must have $\mathsf{true} \to \mathsf{false} \in \mathcal{R}_n$.

For the reverse direction, let n be minimal such that $\mathsf{true} \simeq \mathsf{false} \in \mathcal{E}_n \cup \mathcal{R}_n$. Suppose this is the case because simplify applied $\mathsf{equal}(x, x) \to \mathsf{true}$ or $\mathsf{simplify}_2$ applied $\mathsf{equal}(x, x) \approx \mathsf{true}$ to some $\mathsf{equal}(u, u) \simeq \mathsf{false} \in \mathcal{E}_{n-1}$, or because collapse applied $\mathsf{equal}(x, x) \to \mathsf{true}$ or $\mathsf{collapse}_2$ applied $\mathsf{equal}(x, x) \approx \mathsf{true}$ to some $\mathsf{equal}(u, u) \to \mathsf{false} \in \mathcal{R}_{n-1}$. In either case, by Lemma 5.16 we have $s\sigma \leftrightarrow^*_{\mathcal{E}} u$ and $t\sigma \leftrightarrow^*_{\mathcal{E}} u$ for some substitution σ. Otherwise, if a deduce_2 step added the equation $\mathsf{true} \simeq \mathsf{false}$ because two equations or rules $\mathsf{equal}(x, x) \simeq \mathsf{true}$, $\mathsf{equal}(u, v) \simeq \mathsf{false} \in \mathcal{E}_n \cup \mathcal{R}_n$ admit an overlap at the root position then we must have $u\tau = v\tau$ for some substitution τ. By Lemma 5.16 there is some σ such that $s\sigma \leftrightarrow^*_{\mathcal{E}} u$ and $t\sigma \leftrightarrow^*_{\mathcal{E}} v$, hence $s\sigma\tau \leftrightarrow^*_{\mathcal{E}} t\sigma\tau$. Note that no other inference rule can have added $\mathsf{true} \simeq \mathsf{false}$. Hence there exists a substitution σ such that $s\sigma \leftrightarrow^*_{\mathcal{E}} t\sigma$. □

Example 5.18. In example SYN080-1 from TPTP 3.6.0 [100] the ground goal $\mathsf{f}(\mathsf{f}(\mathsf{a})) \approx \mathsf{f}(\mathsf{g}(\mathsf{b}))$ is to be proven from the singleton axiom set $\mathcal{E} = \{\mathsf{f}(x) \approx \mathsf{g}(y)\}$. Let $\succ$ be an LPO with precedence $\mathsf{f} \succ \mathsf{g} \succ \mathsf{a} \succ \mathsf{b}$. We can run oKB using $\succ_{\mathsf{c}}$ on the equations

$$\mathsf{f}(x) \approx \mathsf{g}(y) \tag{1}$$

$$\mathsf{equal}(x, x) \approx \mathsf{true} \tag{2}$$

$$\mathsf{equal}(\mathsf{f}(\mathsf{f}(\mathsf{a})), \mathsf{f}(\mathsf{g}(\mathsf{b}))) \approx \mathsf{false} \tag{3}$$

An orient step can turn equation (2) into the rule $\mathsf{equal}(x, x) \to \mathsf{true}$. A $\mathsf{simplify}_2$ step applying the instance $\mathsf{f}(\mathsf{a}) \to \mathsf{g}(\mathsf{b})$ of equation (1) to (3) yields

$$\mathsf{equal}(\mathsf{f}(\mathsf{g}(\mathsf{b})), \mathsf{f}(\mathsf{g}(\mathsf{b}))) \approx \mathsf{false} \tag{4}$$

Now simplify can use (2) to rewrite equation (4) to $\mathsf{true} \approx \mathsf{false}$, so by Theorem 5.17 we have $\mathsf{f}(\mathsf{f}(\mathsf{a})) \leftrightarrow^*_{\mathcal{E}} \mathsf{f}(\mathsf{g}(\mathsf{b}))$.

deduce	$\dfrac{\mathcal{E},\mathcal{R}}{\mathcal{E}\cup\{s\approx t\},\mathcal{R}}$	if $s \leftarrow\!\rtimes\!\rightarrow t \in \mathrm{CP}_\succ(\mathcal{E}\cup\mathcal{R})$
orient	$\dfrac{\mathcal{E}\uplus\{s\simeq t\},\mathcal{R}}{\mathcal{E},\mathcal{R}\cup\{s\to t\}}$	if $s \succ t$
delete	$\dfrac{\mathcal{E}\uplus\{s\approx s\},\mathcal{R}}{\mathcal{E},\mathcal{R}}$	
simplify	$\dfrac{\mathcal{E}\uplus\{s\simeq t\},\mathcal{R}}{\mathcal{E}\cup\{s\simeq u\},\mathcal{R}}$	if $t \to_{\mathcal{R}\cup\mathcal{E}_\succ} u$
compose	$\dfrac{\mathcal{E},\mathcal{R}\uplus\{s\to t\}}{\mathcal{E},\mathcal{R}\cup\{s\to u\}}$	if $t \to_{\mathcal{R}\cup\mathcal{E}_\succ} u$
collapse	$\dfrac{\mathcal{E},\mathcal{R}\uplus\{t\to s\}}{\mathcal{E}\cup\{u\approx s\},\mathcal{R}}$	if $t \to_{\mathcal{R}\cup\mathcal{E}_\succ} u$

Figure 5.2: The inference system oKB′ of finite ordered completion.

5.1.2 Finite Runs

As for standard completion, *finite* ordered completion runs can be performed using slightly simpler inference rules. We define the inference system oKB′ by the rules displayed in Figure 5.2. Note that the two simplify, collapse and compose rules are merged now, and the former two enjoy relaxed side conditions. Just like for finite standard completion, in the definition of the proof order we need to uniquely identify rewrite rules, i.e., we need to distinguish between (variants of) the same rewrite rule developed in the course of a run. For this purpose we assume that all equations and rewrite rules occurring during a run are variable-disjoint.

In the remainder of Section 5.1 finite oKB′ runs of the form

$$(\mathcal{E}_0,\varnothing) \vdash (\mathcal{E}_1,\mathcal{R}_1) \vdash (\mathcal{E}_2,\mathcal{R}_2) \vdash \cdots \vdash (\mathcal{E}_n,\mathcal{R}_n) \tag{5.1}$$

are considered. We define a modified proof order $\Rightarrow^{\succ,n}_{\mathsf{oKB}}$ as follows.

Definition 5.19. Consider a run of the form (5.1) which has length $n \in \mathbb{N}$ and uses the reduction order $\succ$. The *cost* c_n of an equational proof step in $\bigcup_i \mathcal{E}_i \cup \mathcal{R}_i$ is defined as follows:

$$c(s \underset{u\approx v}{\longleftrightarrow} t) = (\{s,t\},\bot,n-k,\bot)$$
where k is maximal such that $u \simeq v \in \mathcal{E}_k$

$$c(s \underset{\ell\to r}{\overset{p}{\longleftrightarrow}} t) = c(t \underset{r\leftarrow\ell}{\overset{p}{\longleftrightarrow}} s) = (\{s\}, s|_p, n-k, \bot)$$
where k is maximal such that $\ell \to r \in \mathcal{R}_k$

$$c(s \underset{\ell\to r}{\overset{p}{\longleftrightarrow}} t) = c(t \underset{r\leftarrow\ell}{\overset{p}{\longleftrightarrow}} s) = (\{s\}, s|_p, n-k, \top)$$
where k is maximal such that $\ell \to r \in (\mathcal{E}_k)_\succ$

To compare cost triples we use the lexicographic combination of $\succ_{\mathsf{mul}}$, $\rhd$, the standard order $>$ on $\mathbb{N}$, and the order where $\bot < \top$ (and $\bot$ is considered minimal with respect to $\rhd$). Again the cost of an equational proof is the multiset consisting of the costs of its steps, and the proof order $\gg^{n}_{\mathsf{oKB}}$ on equational proofs is the multiset extension of the order on proof step costs. Finally, we write $P \Rightarrow^{\succ,n}_{\mathsf{oKB}} Q$ if and only if $P \gg^{n}_{\mathsf{oKB}} Q$, and P and Q prove the same equation.

As a lexicographic combination of well-founded orders $\gg$ is terminating, so the following result is easily established.

Lemma 5.20. *The relation $\Rightarrow^{\succ,n}_{\mathsf{oKB}}$ is a proof reduction relation.* □

We next show that oKB' runs constitute equational inference sequences with respect to $\Rightarrow^{\succ,n}_{\mathsf{oKB}}$. We will drop the subscript and write $\gg$ instead of $\gg^{n}_{\mathsf{oKB}}$ if the intended definition is clear from the context.

Lemma 5.21. *A finite oKB' run of length n is an equational inference sequence with respect to $\Rightarrow^{\succ,n}_{\mathsf{oKB}}$.* □

Proof. We argue that every inference step $(\mathcal{E}_i, \mathcal{R}_i) \vdash_{\mathsf{oKB}'} (\mathcal{E}_{i+1}, \mathcal{R}_{i+1})$ can be simulated by expand and $\mathsf{contract}$ steps according to Definition 3.1. To enhance readability, we will simply write $\Rightarrow$ instead of $\Rightarrow^{\succ,n}_{\mathsf{oKB}}$.

Every deduce step clearly constitutes an expansion. A delete inference obviously constitutes a $\mathsf{contract}$ step as $s \leftrightarrow_{s \approx s} s \Rightarrow s$. Every other inference step can be viewed as an expansion adding the respective rule or equation, followed by a contraction.

An orient inference expands by adding $s \to t$ and subsequently removes $s \simeq t$. Note that i is maximal such that $s \simeq t \in \mathcal{E}_i$ while for the maximal j such that $s \to t \in \mathcal{R}_j$ we have $j > i$. Hence the contraction is valid as $s \leftrightarrow^{\epsilon}_{s \approx t} t \Rightarrow s \leftrightarrow^{\epsilon}_{s \to t} t$ because $\{(\{s\}, s, n-i, \top)\} \gg \{(\{s\}, s, n-j, \bot)\}$.

An application of $\mathsf{simplify}$ can be viewed as an expansion inference adding $s \simeq u$ followed by a contraction step removing $s \simeq t$ since $t \to_{\ell \to r} u$ for some rule $\ell \to r$. While i is maximal such that $s \simeq t \in \mathcal{E}_i$, the maximal j such that $s \simeq u \in \mathcal{E}_j$ is greater than i. Also the maximal k such that $\ell \to r$ occurs in $\mathcal{R}_k$ or $(\mathcal{E}_k)_{\succ}$ must be greater than i. The contraction step transforms the proof $P\colon s \leftrightarrow_{s \approx t} t$ into $Q\colon s \leftrightarrow_{s \approx u} u \; {}_{r \leftarrow \ell}\!\leftarrow t$. A case distinction shows that the contraction always results in a smaller proof. If $c(P) = \{(\{s\}, s, n-i, \ldots)\}$ because $s \succ t$ then we have $s \succ u$ and thus $c(Q) = \{(\{s\}, s, n-j, \ldots), (\{t\}, \ldots)\}$, which yields a decrease as $n-i > n-j$. Otherwise, $c(P)$ is $\{(\{s,t\}, \ldots)\}$ or $\{(\{t\}, t, n-i, \ldots)\}$, and $c(Q)$ involves $\{(\{s,u\}, \ldots)$ (or $\{(\{s\}, \ldots)\}$ or $\{(\{u\}, \ldots)\})$ and $(\{t\}, t, n-k, \ldots)$. Because of $t \succ u$, $n-i > n-j$, and $n-i > n-k$ this results in a decrease in both cases.

A $\mathsf{compose}$ step adds $s \to u$ and performs a contraction removing $s \to t$. Here i is maximal such that $s \to t \in \mathcal{R}_i$ but the maximal j such that $s \to u \in \mathcal{R}_j$ is greater than i. The contraction is justified because $s \leftrightarrow_{s \to t} t$ has cost $\{(\{s\}, s, n-i, \ldots)\}$ while the cost of $s \leftrightarrow_{s \to u} u \leftarrow t$ only amounts to $\{(\{s\}, s, n-j, \ldots), (\{t\}, \ldots)\}$, and both $s \succ t$ and $t \succ u$ hold.

Finally, a $\mathsf{collapse}$ step constitutes an expansion adding $u \approx s$ followed by a contraction removing $t \to s$ because $t \to^{p}_{\ell \to r} u$ for some rule $\ell \to r$ and

$p \in \mathcal{P}\mathrm{os}(t)$. Here i is maximal such that $t \to s \in \mathcal{R}_i$ but the maximal k such that $\ell \to r$ occurs in $\mathcal{R}_k$ or $(\mathcal{E}_k)_\succ$ must be greater than i. On the proof level, this contraction replaces $t \leftrightarrow_{t\to s} s$ with cost $\{(\{t\}, t, n-i, \ldots)\}$ by $t \to u \leftrightarrow_{u\approx s} s$, where the cost of the latter consists of $(\{t\}, t|_p, n-k, \ldots)$ and one of $(\{s,u\}, \ldots)$, $(\{s\}, \ldots)$, or $(\{u\}, \ldots)$. This results in a smaller proof as $t \succ s, u$, $t \trianglerighteq t|_p$, and $n - i > n - k$. □

Fairness is defined exactly as in Definition 5.8 except that the modified proof order $\Rightarrow^{\succ,n}_{\mathsf{oKB}}$ is used. The following results on fairness, correctness, and completeness can be proven in exactly the same way as for oKB since all comparisons of proof steps involve only the first component where the proof orders $\Rightarrow^{\succ}_{\mathsf{oKB}}$ and $\Rightarrow^{\succ,n}_{\mathsf{oKB}}$ coincide.

Lemma 5.22. *Let $\succ$ be completable for $\mathcal{E}_0$. If an oKB' run $(\mathcal{E}_0, \varnothing) \vdash (\mathcal{E}_1, \mathcal{R}_1) \vdash (\mathcal{E}_2, \mathcal{R}_2) \vdash \cdots \vdash (\mathcal{E}_n, \mathcal{R}_n)$ satisfies $\mathrm{CP}_\succ(\mathcal{E}_n \cup \mathcal{R}_n) \subseteq \bigcup_i \mathcal{E}_i$ then it is fair.* □

Correctness Theorem 5.23. *Let $\succ$ be completable for $\mathcal{E}_0$, and $(\mathcal{E}_0, \varnothing) \vdash (\mathcal{E}_1, \mathcal{R}_1) \vdash \cdots \vdash (\mathcal{E}_n, \mathcal{R}_n)$ be a fair oKB' run using $\succ$. Then $(\mathcal{E}_n, \mathcal{R}_n)$ is ground convergent for $\mathcal{E}_0$ with respect to any complete extension of $\succ$.* □

Lemma 5.24. *Let $(\mathcal{E}_0, \varnothing) \vdash (\mathcal{E}_1, \mathcal{R}_1) \vdash \cdots \vdash (\varnothing, \mathcal{R}_n)$ be an oKB' run such that $\mathrm{CP}(\mathcal{R}_n) \subseteq \bigcup_i \mathcal{E}_i$. Then $\mathcal{R}_n$ is convergent for $\mathcal{E}_0$.* □

Completeness Theorem 5.25. *Let $\succ$ be completable for $\mathcal{E}$, and assume $\mathcal{R}$ is a finite canonical rewrite system for $\mathcal{E}$ such that $\mathcal{R} \subseteq \succ$. Then any fair and simplifying oKB' run $(\mathcal{E}_0, \varnothing) \vdash (\mathcal{E}_1, \mathcal{R}_1) \vdash \cdots \vdash (\mathcal{E}_n, \mathcal{R}_n)$ using $\succ$ yields $\mathcal{E}_n = \varnothing$ and $\mathcal{R}_n = \mathcal{R}$ (modulo variable renaming).* □

Theorem 5.26. *Let $\mathcal{E}$ be a set of equations and $s \approx t$ be a goal such that $\succ$ is completable for $\mathcal{E} \cup \{s \approx t\}$, and consider a fair oKB' run $((\mathcal{E}^{s\approx t}, \varnothing) \vdash (\mathcal{E}_1, \mathcal{R}_1) \vdash \cdots \vdash (\mathcal{E}_n, \mathcal{R}_n)$ using $\succ_{\mathsf{c}}$. Then $\mathsf{true} \simeq \mathsf{false} \in \bigcup_i \mathcal{E}_i \cup \mathcal{R}_i$ if and only if there exists a substitution σ such that $s\sigma \leftrightarrow^*_{\mathcal{E}} t\sigma$.* □

5.1.3 Critical Pair Criteria

As for standard completion, critical pair criteria can be used as a means to filter out critical pairs that can be ignored without compromising completeness. Again the *compositeness criterion* serves as a general condition.

Let $\mathcal{E}$ be a set of equations, $\mathcal{R}$ be a set of rewrite rules, and $\succ$ be a reduction order. A critical pair criterion $\mathrm{CPC}_\succ$ maps $(\mathcal{E}, \mathcal{R})$ to a set of equations such that $\mathrm{CPC}_\succ(\mathcal{E}, \mathcal{R})$ is a subset of $\mathrm{CP}_\succ(\mathcal{E} \cup \mathcal{R})$. Intuitively, $\mathrm{CPC}_\succ(\mathcal{E}, \mathcal{R})$ captures superfluous critical pairs. We use the same notion of compositeness as for standard completion, except that a different proof order is used.

Definition 5.27. Let $\mathcal{E}$ be a set of equations, $\mathcal{R}$ a set of rewrite rules, and $\succcurlyeq$ a proof order using reduction order $\succ$. An equational proof $P\colon s \leftarrow u \to t$ is *composite* with respect to $(\mathcal{E}, \mathcal{R})$ and $\succcurlyeq$ if there exist terms $u_0, \ldots, u_{n+1}$ such that $s = u_0$, $t = u_{n+1}$, and $u \succ u_i$ for all $1 \leqslant i \leqslant n$, together with proofs $P_0, \ldots, P_n$ in $(\mathcal{E}, \mathcal{R})$ such that P_i proves $u_i \approx u_{i+1}$ and $P \succcurlyeq P_i$ holds for all $1 \leqslant i \leqslant n$.

The *compositeness criterion* $\mathrm{CCP}_\succ(\mathcal{E},\mathcal{R})$ returns all extended critical pairs among equations in $\mathcal{E}$ and rules in $\mathcal{R}$ for which the associated overlaps are composite with respect to $(\mathcal{E},\mathcal{R})$ and $\succcurlyeq$. Ignoring composite critical pairs does not affect fairness, as the following result shows, thereby relaxing Lemma 5.9.

Lemma 5.28. *Let $\succ$ be complete for $\mathcal{E}_0$. Consider an* oKB *run $\gamma\colon (\mathcal{E}_0,\varnothing) \vdash (\mathcal{E}_1,\mathcal{R}_1) \vdash (\mathcal{E}_2,\mathcal{R}_2) \vdash \cdots$ using $\succ$ and let $\mathcal{C}$ be a subset of $\bigcup_i \mathrm{CCP}_\succ(\mathcal{E}_i \cup \mathcal{R}_i)$, where* CCP *is computed with respect to $\succcurlyeq_{\mathsf{oKB}}$. If $\mathrm{CP}_\succ(\mathcal{E}_\omega \cup \mathcal{R}_\omega) \setminus \mathcal{C} \subseteq \bigcup_i \mathcal{E}_i$ then γ is fair.*

Proof. We argue by induction on $\succcurlyeq_{\mathsf{oKB}}$ that any ground non-rewrite proof in $(\mathcal{E}_\omega)_\succ \cup \mathcal{R}_\omega$ can be transformed into a rewrite proof. Such a non-rewrite proof must contain a peak $P\colon s \leftarrow u \rightarrow t$. If this peak does not constitute a proper overlap then according to Corollary 5.7 there exists a proof P' such that $P \Rightarrow^{\succ}_{\mathsf{oKB}} P'$, and by the induction hypothesis P' can be transformed into a rewrite proof in $(\mathcal{E}_\omega)_\succ \cup \mathcal{R}_\omega$. Otherwise, $s = C[\ell\sigma]$ and $t = C[r\sigma]$ for some extended critical pair $\ell \leftarrow\!\times\!\rightarrow r$ in $\mathrm{CP}_\succ(\mathcal{E}_\omega \cup \mathcal{R}_\omega)$, and $P \Rightarrow^{\succ}_{\mathsf{oKB}} s \leftrightarrow^{\epsilon}_{\ell\approx r} t$. If $\ell \simeq r$ occurs in some set $\mathcal{E}_i$ then by the Persistence Lemma 3.4 there is a proof P' in $(\mathcal{E}_\omega)_\succ \cup \mathcal{R}_\omega$ such that $s \leftrightarrow^{\epsilon}_{\ell\approx r} t$ $(\Rightarrow^{\succ}_{\mathsf{oKB}})^{=}$ P'. By the induction hypothesis P' can be transformed into a rewrite proof and hence this also holds for P. Otherwise, $\ell \simeq r \in \mathrm{CCP}_\succ(\mathcal{E}_i,\mathcal{R}_i)$ for some i. Let v be the term at the peak of the corresponding overlap $P'\colon \ell \leftarrow v \rightarrow r$. By definition, there are terms $v_0,\ldots,v_{n+1}$ such that $s = v_0$, $t = v_{n+1}$ and $v \succ v_i$, and $(\mathcal{E}_i,\mathcal{R}_i)$ admits proofs P_i of $v_i \approx v_{i+1}$ for which $P' \succcurlyeq_{\mathsf{oKB}} P_i$. Thus the proofs $C[P_i\sigma]$ prove $C[v_i\sigma] \approx C[v_{i+1}\sigma]$ for all $1 \leqslant i \leqslant n$, and by the Persistence Lemma 3.4 there are respective proofs P_i' in $(\mathcal{E}_\omega)_\succ \cup \mathcal{R}_\omega$ such that $C[P_i\sigma]$ $(\Rightarrow^{\succ}_{\mathsf{oKB}})^{=}$ P_i'. According to the induction hypothesis all these proofs can be transformed into rewrite proofs Q_i of $C[v_i\sigma] \approx C[v_{i+1}\sigma]$. Thus all terms occurring in the proofs Q_i are smaller than $u = C[v\sigma]$ such that $P \Rightarrow^{\succ}_{\mathsf{oKB}} Q_1 \cdots Q_n$ holds. By the induction hypothesis $Q_1 \cdots Q_n$ can be transformed into a rewrite proof, hence this also holds for P. □

A completely analogous proof can be used to show a corresponding result for oKB', thereby relaxing Lemma 5.22.

Lemma 5.29. *Let $\succ$ be complete for $\mathcal{E}_0$. Consider a finite* oKB' *run $\gamma\colon (\mathcal{E}_0,\varnothing) \vdash (\mathcal{E}_1,\mathcal{R}_1) \vdash \cdots \vdash (\mathcal{E}_n,\mathcal{R}_n)$ using $\succ$ and let $\mathcal{C}$ be a subset of $\bigcup_i \mathrm{CCP}_\succ(\mathcal{E}_i \cup \mathcal{R}_i)$, where* CCP *is computed with respect to $\succcurlyeq^n_{\mathsf{oKB}}$. If $\mathrm{CP}_\succ(\mathcal{E}_\omega \cup \mathcal{R}_\omega) \setminus \mathcal{C} \subseteq \bigcup_i \mathcal{E}_i$ then γ is fair.* □

The compositeness criterion only applies to overlaps $s \leftarrow u \rightarrow t$ such that $u \succ s$ and $u \succ t$. In contrast to standard critical pairs this need not hold for extended critical pairs. We therefore restrict to *oriented* extended overlaps $\langle \ell_1 \approx r_1, p, \ell_2 \approx r_2\rangle_\sigma$ which satisfy $\ell_2[\sigma] \succ \ell_2\sigma[r_1\sigma]_p$ and $\ell_2[\sigma] \succ r_2\sigma$. Note that this is equivalent to $\ell_i \rightarrow r_i \in \mathcal{R}$ or $\ell_i\sigma \rightarrow r_i\sigma \in \mathcal{E}_\succ$ for both $i \in \{1,2\}$. As concrete instances of compositeness we outline the primality criterion [51] and the connectedness criterion [61] in the setting of (finite) ordered completion.

Primality

Let $\mathcal{E}$ be a set of equations and $\mathcal{R}$ be a set of rewrite rules such that $\succ$ can be extended to a complete reduction order for $\mathcal{E} \cup \mathcal{R}$. An oriented overlap is *prime* if $\ell_2\sigma$ is not reducible by $\mathcal{E}_\succ \cup \mathcal{R}$ at any position strictly below p. The primality criterion $\mathrm{PCP}_\succ(\mathcal{E}, \mathcal{R})$ returns all oriented critical pairs among rules in $\mathcal{R}$ for which the associated overlaps are not prime.

Lemma 5.30. *Every oriented extended critical pair which is not prime is composite with respect to* $\succcurlyeq_{\mathsf{oKB}}$ *and* $\succcurlyeq^n_{\mathsf{oKB}}$.

Proof. Consider an oriented overlap $\langle \ell_1 \approx r_1, p, \ell_2 \approx r_2 \rangle_\sigma$ with corresponding peak $P\colon s \;{}_{r_1\approx\ell_1}{}^{p}\!\leftarrow u \rightarrow^{\epsilon}_{\ell_2\approx r_2} t$. Suppose the extended critical pair $s \leftarrow \rtimes \rightarrow t$ is not prime since $u \rightarrow^q_{\mathcal{E}_\succ \cup \mathcal{R}} v$ for $p < q$. We have $u \succ s, t, v$ by assumption. Let proofs P_1 and P_2 be defined by $P_1\colon s \;{}_{r_1\approx\ell_1}{}^{p}\!\leftarrow u \rightarrow^q v$ and $P_2\colon v \;{}^{q}\!\leftarrow u \rightarrow^{\epsilon}_{\ell_2\approx r_2} t$. Note that P_1P_2 proves $s \approx t$. For both $\succcurlyeq_{\mathsf{oKB}}$ and $\succcurlyeq^n_{\mathsf{oKB}}$, the cost of P is of the form $\{(\{u\}, u|_p, \ldots), (\{u\}, u, \ldots)\}$ whereas $c(P_1) = \{(\{u\}, u|_p, \ldots), (\{u\}, u|_q, \ldots)\}$, and $c(P_2) = \{(\{u\}, u|_q, \ldots), (\{u\}, u, \ldots)\}$. Since $u \rhd u|_p \rhd u|_q$ both $P \succcurlyeq P_1$ and $P \succcurlyeq P_2$ hold, independent of whether $\succcurlyeq$ is $\succcurlyeq_{\mathsf{oKB}}$ or $\succcurlyeq^n_{\mathsf{oKB}}$. Thus P is composite. □

Again a special case of $\mathrm{PCP}_\succ$ is captured by the *unblockedness criterion* $\mathrm{BCP}_\succ$ [9] (see Section 3.2).

Connectedness

Also the *connectedness* criterion extends to the setting of ordered completion. Given a set of equations $\mathcal{E}$ and a set of rewrite rules $\mathcal{R}$, an extended critical pair $s \leftarrow \times \rightarrow t$ originating from an oriented overlap $s \leftarrow u \rightarrow t$ is *connected below* u if there exists an equational proof $s = u_0 \leftrightarrow u_1 \leftrightarrow \cdots \leftrightarrow u_{n+1} = t$ in $(\mathcal{E}, \mathcal{R})$ such that $u \succ u_i$ for all $1 \leqslant i \leqslant n$.

Lemma 5.31. *If an extended critical pair stemming from an oriented overlap* $s \leftarrow u \rightarrow t$ *is connected below* u *then it is composite with respect to both* $\succcurlyeq_{\mathsf{oKB}}$ *and* $\succcurlyeq^n_{\mathsf{oKB}}$.

Proof. Let P_i denote the single-step proof $u_i \leftrightarrow u_{i+1}$ in $(\mathcal{E}, \mathcal{R})$. For both proof orders under consideration $c(P) = \{(\{u\}, \ldots), (\{u\}, \ldots)\}$ while $c(P_i)$ consists of a single tuple of the form $(\{u_i\}, \ldots)$, $(\{u_{i+1}\}, \ldots)$, or $(\{u_i, u_{i+1}\}, \ldots)$. Therefore $u \succ u_i, u_{i+1}$ implies $P \succcurlyeq P_i$ for all $0 \leqslant i \leqslant n$. Since obviously $P_0 \cdots P_n$ proves $s \approx t$, the extended critical pair is composite. □

In practice this rather abstract criterion can be approximated for instance by the weak connectivity test described in Section 3.2, and the resulting critical pair criterion is denoted $\mathrm{WCP}_\succ$. As in the case of standard completion, the criteria $\mathrm{WCP}_\succ$ and $\mathrm{PCP}_\succ$ can be combined since they both constitute special cases of compositeness. This *mixed* criterion will again be referred to as $\mathrm{MCP}_\succ$.

5.2 Ordered Multi-Completion

Kondo and Kurihara [62] proposed the use of multiple reduction orders also for the setting of ordered completion. To this end all reduction orders in the set $\mathcal{O}$ are assumed to be completable for the theory under consideration. In the inference system MKB the rewrite and deduce rules are replaced by the rules in Figure 5.3. We refer to the combined set of inference rules by oMKB.

odeduce	$\dfrac{N}{N \cup \{\langle s : t, \varnothing, \varnothing, L\rangle\}}$
	if $\langle \ell_1 : r_1, R, \ldots, E\rangle$, $\langle \ell_2 : r_2, R', \ldots, E'\rangle \in N$ and $L \subseteq (R \cup E) \cap (R' \cup E')$ such that $s \leftarrow \rtimes \rightarrow t \in \mathrm{CP}_\succ(\ell_1 \rightarrow r_1, \ell_2 \rightarrow r_2)$ for all $\succ \in L$ and $L \neq \varnothing$
$\mathsf{orewrite}_1$	$\dfrac{N \uplus \{\langle s : t, R_0, R_1, E\rangle\}}{N \cup \{\langle s : t, R_0 \setminus L, R_1, E \setminus L\rangle,\ \langle s : u, R_0 \cap L, \varnothing, E \cap L\rangle\}}$
	if $\langle \ell : r, R, \ldots, E \uplus E'\rangle \in N$, $t \rightarrow_{\ell \rightarrow r} u$ such that $t \doteq \ell$, $L = R \cup E$ where $t \succ u$ for all $\succ \in E$, and $(R_0 \cup E) \cap L \neq \varnothing$
$\mathsf{orewrite}_2$	$\dfrac{N \uplus \{\langle s : t, R_0, R_1, E\rangle\}}{N \cup \{\langle s : t, R_0 \setminus L, R_1 \setminus L, E \setminus L\rangle,\ \langle s : u, R_0 \cap L, \varnothing, (E \cup R_1) \cap L\rangle\}}$
	if $\langle \ell : r, R, \ldots, E \uplus E'\rangle \in N$, $t \rightarrow_{\ell \rightarrow r} u$ such that $t \rhd \ell$, $L = R \cup E$ where $t \succ u$ for all $\succ \in E$, and $(R_0 \cup R_1 \cup E) \cap L \neq \varnothing$

Figure 5.3: Rules in the inference system for ordered multi-completion (oMKB).

Runs and projections functions are defined exactly as for MKB. Adapting Lemmas 4.3 and 4.4 to the setting of ordered completion yields the following simulation results.

Simulation Soundness Lemma 5.32. *If $N_0 \vdash^\alpha_{\mathsf{oMKB}} N_\alpha$ then for all $\succ_i \in \mathcal{O}$ there is some $\beta \leqslant \alpha$ such that $(E[N_0, i], R[N_0, i]) \vdash^\beta_{\mathsf{oKB}} (E[N_\alpha, i], R[N_\alpha, i])$.* □

An oMKB run γ can thus be projected to oKB runs γ_i for all $\succ_i \in \mathcal{O}$.

Simulation Completeness Lemma 5.33. *Let N_0 be a node set such that $\mathcal{E}_0 = E[N_0, i]$ and $\mathcal{R}_0 = R[N_0, i]$ for some $\succ_i \in \mathcal{O}$ and there is a run $(\mathcal{E}_0, \mathcal{R}_0) \vdash^\alpha_{\mathsf{oKB}} (\mathcal{E}_\alpha, \mathcal{R}_\alpha)$. Then there is a node set N_α such that $N_0 \vdash^\alpha_{\mathsf{oMKB}} N_\alpha$, $\mathcal{E}_\alpha = E[N_\alpha, i]$, and $\mathcal{R}_\alpha = R[N_\alpha, i]$ hold.* □

A run γ is *fair* if some γ_i is fair. Using this definition and the simulation properties, the following correctness result is straightforward.

Correctness Theorem 5.34. *Let $\mathcal{E}$ be a set of equations and $N_\mathcal{E}$ be an initial node set whose reduction orders $\mathcal{O}$ are completable for $\mathcal{E}$. For any fair run $N_\mathcal{E} \vdash^\alpha_{\mathsf{MKB}} N$ there is then some $\succ_i \in \mathcal{O}$ such that $(E[N, i], R[N, i])$ is ground convergent with respect to any complete extension of $\succ_i$.*

Proof. By Lemma 5.32 and fairness there is some $\succ_i \in \mathcal{O}$ such that the oKB run γ_i is fair. Hence the claim follows from Theorem 5.10. □

5.3 Ordered Completion with Termination Tools

Although ordered completion never fails, the reduction order still has significant impact on the success of a run. For instance, an unfortunate choice of the input order may lead to an infinite ground convergent system, while another reduction order could admit a finite result. Both ordered completion and completion-based theorem proving can thus benefit from extending the class of applicable reduction orders as facilitated by the use of termination tools. We will first present ordered completion with termination tools by adopting an approach very similar to KBtt. In the next step this method will be combined with multi-completion to efficiently pursue multiple processes in parallel.

Ordered completion with termination tools (oKBtt) is described by the rules depicted in Figure 5.4 together with orient, delete, simplify, compose and collapse from KBtt.

deduce_2	$\dfrac{\mathcal{E},\mathcal{R},\mathcal{C}}{\mathcal{E}\cup\{s\approx t\},\mathcal{R},\mathcal{C}}$	if $s \leftarrow_{\mathcal{E}\cup\mathcal{R}} u \rightarrow_{\mathcal{E}\cup\mathcal{R}} t$
$\mathsf{simplify}_2$	$\dfrac{\mathcal{E}\uplus\{s\simeq t\},\mathcal{R},\mathcal{C}}{\mathcal{E}\cup\{s\simeq u\},\mathcal{R},\mathcal{C}\cup\{\ell\sigma\to r\sigma\}}$	if $t \xrightarrow{\ell\sigma\to r\sigma} u$ using $\ell\simeq r\in\mathcal{E}$ such that $\mathcal{C}\cup\{\ell\sigma\to r\sigma\}$ terminates
$\mathsf{compose}_2$	$\dfrac{\mathcal{E},\mathcal{R}\uplus\{s\to t\},\mathcal{C}}{\mathcal{E},\mathcal{R}\cup\{s\to u\},\mathcal{C}\cup\{\ell\sigma\to r\sigma\}}$	if $t \xrightarrow{\ell\sigma\to r\sigma} u$ using $\ell\simeq r\in\mathcal{E}$ such that $\mathcal{C}\cup\{\ell\sigma\to r\sigma\}$ terminates
$\mathsf{collapse}_2$	$\dfrac{\mathcal{E},\mathcal{R}\uplus\{t\to s\},\mathcal{C}}{\mathcal{E}\cup\{u\approx s\},\mathcal{R},\mathcal{C}\cup\{\ell\sigma\to r\sigma\}}$	if $t \xrightarrow{\ell\sigma\to r\sigma} u$ using $\ell\simeq r\in\mathcal{E}$ such that $\mathcal{C}\cup\{\ell\sigma\to r\sigma\}$ terminates

Figure 5.4: Some inference rules of oKBtt.

An inference sequence $\gamma : (\mathcal{E}_0,\mathcal{R}_0,\mathcal{C}_0) \vdash (\mathcal{E}_1,\mathcal{R}_1,\mathcal{C}_1) \vdash \cdots \vdash (\mathcal{E}_n,\mathcal{R}_n,\mathcal{C}_n)$ with respect to oKBtt will always supposed to be finite and is called a *run*. We will in the sequel assume that $\mathcal{R}_0 = \mathcal{C}_0 = \varnothing$. Below we show that oKBtt can be simulated by oKB′ and vice versa.

Simulation Soundness Lemma 5.35. *Any run* $(\mathcal{E}_0,\varnothing,\varnothing) \vdash^n_{\mathsf{oKBtt}} (\mathcal{E}_n,\mathcal{R}_n,\mathcal{C}_n)$ *admits an* oKB′ *run* $(\mathcal{E}_0,\varnothing) \vdash^n_{\mathsf{oKB}'} (\mathcal{E}_n,\mathcal{R}_n)$ *using reduction order* $\rightarrow^+_{\mathcal{C}_n}$.

Proof. Let $\succ_n$ denote $\rightarrow^+_{\mathcal{C}_n}$. We use induction on n. The claim is trivially true for $n = 0$. For a run of the form $(\mathcal{E}_0,\varnothing,\varnothing) \vdash^* (\mathcal{E}_n,\mathcal{R}_n,\mathcal{C}_n) \vdash (\mathcal{E}_{n+1},\mathcal{R}_{n+1},\mathcal{C}_{n+1})$, the induction hypothesis yields a corresponding oKB′ run $(\mathcal{E}_0,\varnothing) \vdash^* (\mathcal{E}_n,\mathcal{R}_n)$ using the reduction order $\succ_n$. Since constraint rules are never removed we have $\mathcal{C}_k \subseteq \mathcal{C}_{n+1}$ for all $k \leqslant n$, so the same run can be obtained with $\succ_{n+1}$. A case distinction on the applied oKBtt rule shows that an oKB′ step $(\mathcal{E}_n,\mathcal{R}_n) \vdash (\mathcal{E}_{n+1},\mathcal{R}_{n+1})$ using $\succ_{n+1}$ is possible.

If orient added the rule $s \to t$ then $s \succ_{n+1} t$ holds by definition, so oKB′ can apply orient as well. In case $\mathsf{simplify}_2$, $\mathsf{compose}_2$ or $\mathsf{collapse}_2$ was applied using an

instance $\ell\sigma \to r\sigma$ of an equation in $\mathcal{E}_n$, we have $\ell\sigma \succ_{n+1} r\sigma$ by definition of the inference rules, hence the respective oKB' step can be applied as well. Clearly, in the remaining cases the inference step can be simulated by the corresponding oKB' rule since no conditions on the order are involved. □

Simulation Completeness Lemma 5.36. *If* $(\mathcal{E}_0, \varnothing) \vdash^n_{\mathsf{oKB}'} (\mathcal{E}_n, \mathcal{R}_n)$ *is a valid* oKB' *run using* $\succ$ *then there is also a valid* oKBtt *run* $(\mathcal{E}_0, \varnothing, \varnothing) \vdash^n_{\mathsf{oKBtt}} (\mathcal{E}_n, \mathcal{R}_n, \mathcal{C}_n)$ *such that* $\mathcal{C}_n \subseteq \succ$.

Proof. By induction on n. For $n = 0$ the claim is trivially satisfied. For a run $(\mathcal{E}_0, \varnothing) \vdash^* (\mathcal{E}_n, \mathcal{R}_n) \vdash (\mathcal{E}_{n+1}, \mathcal{R}_{n+1})$ the induction hypothesis yields the existence of an oKBtt run $(\mathcal{E}_0, \mathcal{R}_0, \mathcal{C}_0) \vdash^* (\mathcal{E}_n, \mathcal{R}_n, \mathcal{C}_n)$ such that $\mathcal{C}_n \subseteq \succ$. An easy case distinction on the oKB' step $(\mathcal{E}_n, \mathcal{R}_n) \vdash (\mathcal{E}_{n+1}, \mathcal{R}_{n+1})$ shows that using $\succ$ for termination checks allows for a corresponding oKBtt step.

If the applied inference rule is orient then $\mathcal{E}_n = \mathcal{E}_{n+1} \uplus \{s \simeq t\}$, $\mathcal{R}_{n+1} = \mathcal{R}_n \uplus \{s \to t\}$ and $s \succ t$. Thus also $\mathcal{C}_n \cup \{s \to t\} \subseteq \succ$, ensuring termination of the extended constraint system. Hence the oKBtt inference orient can be applied to obtain a step $(\mathcal{E}_n, \mathcal{R}_n, \mathcal{C}_n) \vdash (\mathcal{E}_n \setminus \{s \approx t\}, \mathcal{R}_n \cup \{s \to t\}, \mathcal{C}_n \cup \{s \to t\})$. If the inference step $(\mathcal{E}_n, \mathcal{R}_n) \vdash (\mathcal{E}_{n+1}, \mathcal{R}_{n+1})$ applies $\mathsf{compose}_2$, $\mathsf{simplify}_2$ or $\mathsf{collapse}_2$ then an equation $\ell \simeq r$ is used with a substitution σ such that $\ell\sigma \succ r\sigma$. Thus also $\mathcal{C}_n \cup \{\ell\sigma \to r\sigma\} \subseteq \succ$, ensuring termination of the extended constraint system such that the respective oKBtt rule is applicable with $\mathcal{C}_{n+1} = \mathcal{C}_n \cup \{\ell\sigma \to r\sigma\}$. In the remaining cases one can set $\mathcal{C}_{n+1} = \mathcal{C}_n$ and replace the applied rule by its oKBtt counterpart since no conditions on the order are involved. □

Totalizability

Lemma 5.35 shows that an oKBtt run resulting in the final constraint system $\mathcal{C}$ can be simulated by oKB' using the reduction order $\to^+_{\mathcal{C}}$. If this order should play the role of $\succ$ in the Correctness Theorem 5.23 then it has to be contained in a reduction order $>$ which is complete for the theory. Unfortunately, such an order does not always exist. In the proof of the Extended Critical Pair Lemma 5.6, totalizability of the reduction order is needed to guarantee joinability of variable overlaps. Assume an oKBtt run has a final state $(\mathcal{E}, \mathcal{R}, \mathcal{C})$ such that $\to^+_{\mathcal{C}}$ cannot be extended to a complete order for the theory. If $\mathcal{E} = \varnothing$ then $\mathcal{R}$ is nevertheless convergent (cf. Theorem 5.11). Otherwise $(\mathcal{E}, \mathcal{R})$ need not constitute a ground convergent system, as illustrated by the following example.

Example 5.37. A fair oKBtt run starting from

$$\mathsf{f}(\mathsf{a}+\mathsf{c}) \approx \mathsf{f}(\mathsf{c}+\mathsf{a}) \qquad \mathsf{a} \approx \mathsf{b} \qquad \mathsf{g}(\mathsf{c}+\mathsf{b}) \approx \mathsf{g}(\mathsf{b}+\mathsf{c}) \qquad x+y \approx y+x$$

might produce the following result $(\mathcal{E}, \mathcal{R})$:

$$x+y \approx y+x \qquad\qquad \begin{aligned} \mathsf{a} &\to \mathsf{b} \\ \mathsf{f}(\mathsf{b}+\mathsf{c}) &\to \mathsf{f}(\mathsf{c}+\mathsf{b}) \\ \mathsf{g}(\mathsf{c}+\mathsf{b}) &\to \mathsf{g}(\mathsf{b}+\mathsf{c}) \end{aligned}$$

with $\mathcal{C} = \mathcal{R} \cup \{\mathsf{f}(\mathsf{a}+\mathsf{c}) \to \mathsf{f}(\mathsf{c}+\mathsf{a})\}$. It is easy to show that $(\mathcal{E}, \mathcal{R})$ cannot constitute a ground convergent system with respect to any reduction order $>$ that is complete for the theory: Consider the instance $\mathsf{c}+\mathsf{a} \approx \mathsf{a}+\mathsf{c}$ of commutativity. Note that no reduction order $>$ extending $\to_{\mathcal{C}}^{+}$ can orient this equation from left to right. Hence $\mathsf{a}+\mathsf{c} > \mathsf{c}+\mathsf{a}$ must hold, which admits the variable overlap $\mathsf{b}+\mathsf{c} \leftarrow \mathsf{a}+\mathsf{c} \to \mathsf{c}+\mathsf{a} \to \mathsf{c}+\mathsf{b}$. Due to the rules in $\mathcal{C}$ the terms $\mathsf{b}+\mathsf{c}$ and $\mathsf{c}+\mathsf{b}$ have to be incomparable in $>$, so the overlap is not joinable.

To solve this problem we restrict the termination checks in oKBtt inferences.

Definition 5.38. An *oKBtt$_{\mathbb{T}}$ procedure* refers to any program which implements the inference rules of oKBtt and employs a termination tool $\mathbb{T}$ for termination checks in orient, simplify$_2$, compose$_2$ and collapse$_2$ inferences. An oKBtt$_{\text{total}}$ procedure is an oKBtt$_{\mathbb{T}}$ procedure where $\mathbb{T}$ ensures *total* termination of the checked system.

Thus, for any constraint system $\mathcal{C}$ derived in an oKBtt$_{\text{total}}$ run the relation $\to_{\mathcal{C}}^{+}$ can be extended to a ground-total reduction order.

Correctness

The Correctness Theorem 5.10 states that an ordered completion run produces a ground convergent system—provided it is fair. In implementations fairness is typically ensured by considering extended critical pairs with respect to the employed reduction order. However, in oKBtt runs the role of the reduction order is played by the relation $\to_{\mathcal{C}}^{+}$ depending on the final constraint system $\mathcal{C}$, which is not known in advance. Thus the set of extended critical pairs cannot be computed during a run, though it can be overapproximated: Note that $\mathrm{CP}_{\succ}(\mathcal{E}) \subseteq \mathrm{CP}_{>}(\mathcal{E})$ holds for all reduction orders $> \subseteq \succ$ and ES $\mathcal{E}$. For instance, we have $\mathrm{CP}_{\succ}(\mathcal{E}) \subseteq \mathrm{CP}_{\varnothing}(\mathcal{E})$—but considering all critical pairs in $\mathrm{CP}_{\varnothing}(\mathcal{E})$ is rather inefficient in an implementation. Since any ground-total reduction order $\succ$ is a simplification order (cf. Section 2.2) we also have $\rhd \subseteq \succ$, so $\mathrm{CP}_{\succ}(\mathcal{E} \cup \mathcal{R})$ can be overapproximated by $\mathrm{CP}_{\rhd}(\mathcal{E} \cup \mathcal{R})$. This motivates the following definition.

Definition 5.39. An oKBtt$_{\text{total}}$ run $(\mathcal{E}_0, \varnothing) \vdash^{*} (\mathcal{E}_n, \mathcal{R}_n)$ is *sufficiently fair* if $\mathrm{CP}_{\rhd}(\mathcal{E}_n \cup \mathcal{R}_n) \subseteq \bigcup_i \mathcal{E}_i$.

Lemma 5.40. *Any sufficiently fair oKBtt$_{\text{total}}$ run is fair.* □

With the above considerations, we can carry over the correctness result of ordered completion to the oKBtt$_{\text{total}}$ setting.

Correctness Theorem 5.41. *Let $(\mathcal{E}_0, \varnothing, \varnothing) \vdash^{*} (\mathcal{E}_n, \mathcal{R}_n, \mathcal{C}_n)$ be a sufficiently fair oKBtt$_{\text{total}}$ run. Then there exists a ground-total extension $>$ of $\to_{\mathcal{C}_n}^{+}$ such that $(\mathcal{E}_n, \mathcal{R}_n)$ is ground convergent for $\mathcal{E}_0$ with respect to $>$.*

Proof. By Lemma 5.35, there exists a corresponding oKB′ run γ using the reduction order $\to_{\mathcal{C}_n}^{+}$. Since $\mathcal{C}_n$ is totally terminating there exists a reduction order $>$ which contains $\to_{\mathcal{C}_n}^{+}$ and is total on $\mathcal{T}(\mathcal{F})$. Any reduction order $>$

which is total on ground terms contains the strict subterm relation $\triangleright$. Hence we have $\mathrm{CP}_{>}(\mathcal{E}_n \cup \mathcal{R}_n) \subseteq \mathrm{CP}_{\triangleright}(\mathcal{E}_n \cup \mathcal{R}_n)$ and as a consequence the sufficiently fair run γ is fair. By correctness of finite ordered completion (Theorem 5.23), $(\mathcal{E}_n, \mathcal{R}_n)$ is ground convergent for $\mathcal{E}_0$ with respect to $>$. □

Note that if an oKBtt run has no equations left in the end then the resulting TRS is convergent, independent of the termination techniques used:

Lemma 5.42. *Let* $(\mathcal{E}_0, \varnothing, \varnothing) \vdash^* (\varnothing, \mathcal{R}_n, \mathcal{C}_n)$ *be an* oKBtt *run such that* $\mathrm{CP}(\mathcal{R}_n) \subseteq \bigcup_i \mathcal{E}_i$ *and* $\succ \subseteq \to^+_{\mathcal{C}_n}$. *Then* $\mathcal{R}_n$ *is convergent for* $\mathcal{E}_0$.

Proof. According to Lemma 5.35, there exists a corresponding oKB′ run γ using the reduction order $\to^+_{\mathcal{C}_n}$. By Lemma 5.24 the TRS $\mathcal{R}_n$ is convergent for $\mathcal{E}_0$. □

Theorem 5.43. *Let* $\succ$ *be ground-total, and assume* $\mathcal{R}$ *is a finite, canonical rewrite system for* $\mathcal{E}$ *such that* $\mathcal{R} \subseteq \succ$. *Then there exists a sufficiently fair and simplifying* oKBtt$_{\mathsf{total}}$ *run* $(\mathcal{E}_0, \varnothing, \varnothing) \vdash^n (\varnothing, \mathcal{R}_n, \mathcal{C}_n)$ *such that* $\mathcal{R}_n = \mathcal{R}$ *(modulo variable renaming).*

Proof. According to Theorem 5.25, there exists an oKB′ run γ producing $\mathcal{R}_\omega = \mathcal{R}$ and $\mathcal{E}_\omega = \varnothing$. By Lemma 5.36 there is a corresponding oKBtt run $(\mathcal{E}, \varnothing, \varnothing) \vdash^* (\varnothing, \mathcal{R}, \mathcal{C})$. This run can be extended to $(\mathcal{E}, \varnothing, \varnothing) \vdash^* (\varnothing, \mathcal{R}, \mathcal{C}) \vdash^* (\mathcal{E}', \mathcal{R}, \mathcal{C})$ by deducing the remaining equations in $\mathcal{E}' = \mathrm{CP}_{\triangleright}(\mathcal{E}_\omega \cup \mathcal{R}_\omega) \setminus \mathrm{CP}_{\succ}(\mathcal{E}_\omega \cup \mathcal{R}_\omega)$ in order to make it sufficiently fair. Since $\mathcal{R}$ is convergent for $\mathcal{E}$, all equations in $\mathcal{E}'$ can be simplified to trivial ones which allows to derive the result $(\varnothing, \mathcal{R}, \mathcal{C})$. □

Example 5.44. Consider the input equations

$$\mathsf{g}(\mathsf{f}(x, \mathsf{b})) \approx \mathsf{a} \tag{1}$$
$$\mathsf{f}(\mathsf{g}(x), y) \approx \mathsf{f}(x, \mathsf{g}(y)) \tag{2}$$

and the monotone algebra $(\mathcal{A}, \succ_{\mathcal{A}})$ with carrier $\mathbb{N}$ and polynomial interpretations $\mathsf{f}_{\mathcal{A}}(x, y) = x + 2y + 1$, $\mathsf{g}_{\mathcal{A}}(x) = x + 1$, and $\mathsf{a}_{\mathcal{A}} = \mathsf{b}_{\mathcal{A}} = 0$. An oKBtt run using $\succ_{\mathcal{A}}$ may derive the following ground convergent system:

$$\begin{aligned}
\mathsf{f}(\mathsf{f}(x, \mathsf{b}), \mathsf{a}) &\approx \mathsf{f}(\mathsf{a}, \mathsf{f}(y, \mathsf{b})) & \mathsf{g}(\mathsf{f}(x, \mathsf{b})) &\to \mathsf{a} \\
\mathsf{f}(\mathsf{f}(x, \mathsf{b}), \mathsf{a}) &\approx \mathsf{f}(\mathsf{f}(y, \mathsf{b}), \mathsf{a}) & \mathsf{f}(x, \mathsf{g}(y)) &\to \mathsf{f}(\mathsf{g}(x), y) \\
\mathsf{f}(\mathsf{a}, \mathsf{f}(x, \mathsf{b})) &\approx \mathsf{f}(\mathsf{a}, \mathsf{f}(y, \mathsf{b})) & \mathsf{f}(\mathsf{g}(x), \mathsf{f}(y, \mathsf{b})) &\to \mathsf{f}(x, \mathsf{a})
\end{aligned}$$

In fact, if equation (2) would be oriented from left to right then the oKBtt run would diverge. Since $\mathsf{f}(x, \mathsf{g}(y)) \to \mathsf{f}(\mathsf{g}(x), y)$ cannot be oriented by any KBO or LPO which compares lists of subterms only from left to right, ordered completion tools that do not support other termination methods (e.g. Waldmeister) cannot derive a ground convergent system.

We finally consider theorem proving with oKBtt. The next theorem states that oKBtt constitutes a sound method for theorem proving, independent of the termination techniques in use.

$$\mathsf{rewrite}_2 \quad \frac{\mathcal{N} \uplus \{\langle s : t, R_0, R_1, E, C_0, C_1\rangle\}}{\begin{array}{l}\mathcal{N} \;\cup \{\langle s : t, R_0 \setminus R, R_1 \setminus R, E \setminus R, C_0, C_1\rangle\} \\ \quad\cup \{\langle s : u, R_0 \cap R, \varnothing, (E \cup R_1) \cap R, \varnothing, \varnothing\rangle\} \\ \quad\cup \{\langle \ell\sigma : r\sigma, \varnothing, \varnothing, \varnothing, R, \varnothing\rangle\}\end{array}}$$

if
- $\langle \ell : r, \ldots, E', \ldots\rangle \in \mathcal{N}$ and $t \xrightarrow{\ell\sigma \to r\sigma} u$
- $R \subseteq E' \cap (R_0 \cup R_1 \cup E)$ such that $C_p(\mathcal{N}) \cup \{\ell\sigma \to r\sigma\}$ terminates for all $p \in R$, and
- $R \neq \varnothing$

$$\mathsf{deduce}_2 \quad \frac{\mathcal{N}}{\mathcal{N} \cup \{\langle s : t, \varnothing, \varnothing, (R \cup E) \cap (R' \cup E'), \varnothing, \varnothing\rangle\}}$$

if
- $\langle \ell_1 : r_1, R, \ldots, E, \ldots\rangle, \langle \ell_2 : r_2, R', \ldots, E', \ldots\rangle \in \mathcal{N}$
- $s \xleftarrow{r_1 \leftarrow \ell_1} u \xrightarrow{\ell_2 \to r_2} t$ and $(R \cup E) \cap (R' \cup E') \neq \varnothing$

Figure 5.5: The $\mathsf{rewrite}_2$ and deduce_2 inference rules in oMKBtt.

Theorem 5.45. *Consider a set of equations $\mathcal{E}$, a goal $s \approx t$, and an* oKBtt *run $(\mathcal{E}^{s\approx t}, \varnothing) \vdash (\mathcal{E}_1, \mathcal{R}_1) \vdash (\mathcal{E}_2, \mathcal{R}_2) \vdash \cdots$. If $\mathsf{true} \simeq \mathsf{false} \in \bigcup_i \mathcal{E}_i \cup \mathcal{R}_i$ then there exists a substitution σ such that $s\sigma \leftrightarrow^*_{\mathcal{E}} t\sigma$.* □

We conclude this section with a completeness result for theorem proving with termination techniques that ensure total termination. Just like Theorem 5.45, it is not hard to prove this result in a similar fashion as Theorem 5.17.

Theorem 5.46. *Consider a set of equations $\mathcal{E}$, a goal $s \approx t$, and a sufficiently fair* $\mathsf{oKBtt}_{\mathsf{total}}$ *run $(\mathcal{E}^{s\approx t}, \varnothing) \vdash (\mathcal{E}_1, \mathcal{R}_1) \vdash (\mathcal{E}_2, \mathcal{R}_2) \vdash \cdots$. If $\mathsf{true} \simeq \mathsf{false} \notin \bigcup_i \mathcal{E}_i \cup \mathcal{R}_i$ then there is no substitution σ such that $s\sigma \leftrightarrow^*_{\mathcal{E}} t\sigma$.* □

5.4 Ordered Multi-Completion with Termination Tools

Ordered multi-completion with termination tools (oMKBtt) simulates multiple oKBtt processes in parallel, in a very similar way as MKBtt emulates KBtt runs. The inference system oMKBtt is described by the inference system in Figure 5.5, together with orient, delete, rewrite and (optionally) subsume and gc as defined for MKBtt in Figure 4.4. Note that the system oMKBtt considered here differs from the inference system presented in [110] in that no process splitting happens in $\mathsf{rewrite}_2$. Although the respective rule in [110] is correct we present a simpler variant here for two reasons: Firstly, all correctness and completeness results can be obtained with this simpler rule, too. Secondly, extensive experiments (described in Chapter 8) reveal that rewriting with equation instances hardly ever occurs in practice.

An oMKBtt inference sequence $N_0 \vdash N_1 \vdash N_2 \vdash \cdots \vdash N_n$ where $N_0 = N_{\mathcal{E}}$ for some set of equations $\mathcal{E}$ is called a *run*. As in the case of MKBtt, in the first place we need to establish some simple properties of runs.

Lemma 5.47. *In an* oMKBtt *run* $N_0 \vdash N_1 \vdash N_2 \vdash \cdots$ *the rewrite system* $C[N_k, p]$ *is terminating and* $R[N_k, p] \subseteq \to^+_{C[N_k,p]}$ *for all node sets* N_k *and* $p \in \mathcal{P}(N_k)$.

Proof. We apply induction on k. The claim holds for N_0 since for the single process ϵ occurring in N_0 we have $R[N_0, \epsilon] = C[N_0, \epsilon] = \varnothing$. In the induction step, assuming that the claim holds for N_k, a case distinction on the rule applied in $N_k \vdash N_{k+1}$ shows that it is also true for N_{k+1}. The cases of orient, rewrite, delete, subsume, and gc can be proven as for MKBtt (see Lemma 4.24). For the two additional rules we argue as follows, using the notation from Figure 5.5.

- Assume $\mathsf{rewrite}_2$ was applied. For all processes $p \in \mathcal{P}(N) \setminus R$ we have $R[N_k, p] = R[N_{k+1}, p]$ such that the proof obligations follow from the induction hypothesis. Let $\succ_i$ denote the relation $\to^+_{C[N_i,p]}$ for all $i \leqslant k+1$. By construction of the inference rule the rewrite system $C[N_{k+1}, p]$ terminates and contains $C[N_k, p]$, so $\succ_k \subseteq \succ_{k+1}$. If $p \in R_0 \cap R$ then $R[N_{k+1}, p] = R[N_k, p] \setminus \{s \to t\} \cup \{s \to u\}$. The relation $s \succ_k t$ holds by induction hypothesis and implies $s \succ_{k+1} t$, and $\ell\sigma \succ_{k+1} r\sigma$ holds by construction. From $s \to t \to_{\ell\sigma \to r\sigma} u$ we thus obtain $s \succ_{k+1} u$. For $p \in R \cap R_1$ and $p \in R \cap E$ we have $R[N_{k+1}, p] \subseteq R[N_k, p]$, so the claim follows from the induction hypothesis.

- If deduce_2 was applied then $R[N_k, p] = R[N_{k+1}, p]$ for all $p \in \mathcal{P}(N_{k+1})$, so the proof obligations follow from the induction hypothesis. □

Lemma 5.47 obviously implies that rewrite projections applied to node sets in oMKBtt runs yield terminating rewrite systems:

Corollary 5.48. *In an* oMKBtt *run* $N_0 \vdash N_1 \vdash N_2 \vdash \cdots$ *the rewrite system* $R[N_k, p]$ *is terminating for all node sets* N_k *with* $k \geqslant 0$ *and every process* $p \in \mathcal{P}(N_k)$. □

Lemma 5.49. *In an* oMKBtt *run* $N_0 \vdash N_1 \vdash N_2 \vdash \cdots$ *every node set* N_k *is well-encoded and satisfies the node condition.*

Proof. By induction on k. The claim clearly holds for N_0. Otherwise, we assume the well-encoded set N_k satisfies the node condition. A case distinction on the applied oMKBtt rule shows that this also holds for N_{k+1}. If orient, rewrite, delete, subsume, or gc was applied we argue as for MKBtt (see the proof of Lemma 4.26). If $\mathsf{rewrite}_2$ or deduce_2 was applied then $\mathcal{P}(N_{k+1}) = \mathcal{P}(N_k)$, so N_{k+1} is clearly well-encoded. The reasoning that N_{k+1} satisfies the node condition is similar as for $\mathsf{rewrite}_2$ and deduce_2, except for the node with data $\ell\sigma : r\sigma$ added in $\mathsf{rewrite}_2$. But this node obviously satisfies the node condition too as it has only one non-empty label. □

The *split set* as well as the *predecessor* of a process associated with an oMKBtt inference step is defined as for MKBtt (see Definition 4.27). Lemmas 5.50 and 5.51 state simulation properties relating oMKBtt runs to oKBtt runs.

Lemma 5.50. *Let N and N' be well-encoded node sets which satisfy the node condition. For an* oMKBtt *step $N \vdash N'$ with split set S the* oKBtt *step*

$$(E[N,p], R[N,p], C[N,p]) \vdash^{=} (E[N',p'], R[N',p'], C[N',p']) \tag{5.2}$$

is valid for all $p' \in \mathcal{P}(N')$ such that $p = pred_S(p')$. Moreover, there exists at least one process $p' \in \mathcal{P}(N')$ for which the step is not an equality step if the rule applied in $N \vdash N'$ is not gc or subsume.

Proof. By a case distinction on the inference step $N \vdash N'$. If an MKBtt rule was applied we can argue as in the proof of Lemma 4.28, since the inference rules in the simulated KBtt step are also present in oKBtt. If one of the additional oMKBtt rules was applied then $p = p'$ holds as no process splitting occurs. In the following reasoning for the remaining cases we use the notation from Figure 5.5.

- If deduce_2 adds a node $\langle s : t, \varnothing, \varnothing, P, \varnothing, \varnothing\rangle$ then for all $p \in P$ we have $\ell_1 \simeq r_1$, $\ell_2 \simeq r_2 \in R[N,p] \cup E[N,p]$. Hence deduce_2 can also add the equation $s \approx t$ in oKBtt. As $E[N',p] = E[N,p] \cup \{s \approx t\}$ the step (5.2) is valid. For all $p \notin R \cap R'$ the inference corresponds to an identity step.
- Assume $\mathsf{rewrite}_2$ was used. For every process $p \notin (R_0 \cup R_1 \cup E) \cap R$ an identity step is obtained. Otherwise, three cases can be distinguished which are distinct due to the node condition.
  i. If $p \in R_0 \cap R$ then $s \to t \in R[N,p]$, $\ell \simeq r \in E[N,p]$, and $C[N,p] \cup \{\ell\sigma \to r\sigma\}$ terminates. Hence $\mathsf{compose}_2$ in KBtt can replace $s \to t$ by $s \to u$ and add a constraint rule $\ell\sigma \to r\sigma$. As $R[N',p] = R[N,p] \setminus \{s \to t\} \cup \{s \to u\}$ and $\ell\sigma \to r\sigma \in C[N',p]$ the step (5.2) is valid.
  ii. If $p \in E \cap R$ then $s \simeq t$, $\ell \simeq r \in E[N,p]$ and $C[N,p] \cup \{\ell\sigma \to r\sigma\}$ terminates. Thus $\mathsf{simplify}_2$ can turn $s \simeq t$ into $s \simeq u$ and add a constraint rule $\ell\sigma \to r\sigma$. As $E[N',p] = E[N,p] \setminus \{s \simeq t\} \cup \{s \simeq u\}$ and $\ell\sigma \to r\sigma \in C[N',p]$ the step (5.2) is valid.
  iii. If $p \in R_1 \cap R$ then $t \to s \in R[N,p]$, $\ell \simeq r \in E[N,p]$, and $C[N,p] \cup \{\ell\sigma \to r\sigma\}$ terminates. A $\mathsf{collapse}_2$ step can remove $t \to s$ and add the equation $u \approx s$. We have $R[N',p] = R[N,p] \setminus \{s \to t\}$, $E[N',p] = E[N,p] \cup \{u \approx s\}$, and $\ell\sigma \to r\sigma \in C[N',p]$ so (5.2) holds.

For any inference rule besides gc and subsume, the non-emptiness requirement for the set of affected labels ensures that (5.2) is a strict step for some $p' \in \mathcal{P}(N')$. □

Lemma 5.51. *Assume for an* oKBtt *inference step $(\mathcal{E}, \mathcal{R}, \mathcal{C}) \vdash (\mathcal{E}', \mathcal{R}', \mathcal{C}')$ there exist a node set N and a process p such that $\mathcal{E} = E[N,p]$, $\mathcal{R} = R[N,p]$ and $\mathcal{C} = C[N,p]$. Then there is some* oMKBtt *inference step $N \vdash N'$ with split set S and a process $p' \in \mathcal{P}(N')$ such that $p = pred_S(p')$, $\mathcal{E}' = E[N',p']$, $\mathcal{R}' = R[N',p']$ and $\mathcal{C}' = C[N',p']$.*

Proof. By case analysis on the step $(\mathcal{E},\mathcal{R},\mathcal{C}) \vdash (\mathcal{E}',\mathcal{R}',\mathcal{C}')$, where we refer to the proof obligations $\mathcal{E}' = E[N',p']$, $\mathcal{R}' = R[N',p']$, and $\mathcal{C}' = C[N',p']$ by $(*)$. If orient, delete, compose, simplify, or collapse was applied then we argue as in the case for MKBtt (see the proof of Lemma 5.51). In all remaining cases we can set $p' = p$ since no process splitting occurs.

– Assume deduce_2 generates an equation $s \approx t$ from an overlap involving equations or rules $\ell_1 \simeq r_1$, $\ell_2 \to r_2 \in \mathcal{E} \cup \mathcal{R}$ then there must be nodes $\langle \ell_1 : r_1, R, \dots, E, \dots \rangle$ and $\langle \ell_2 : r_2, R', \dots, E', \dots \rangle$ in N such that $p \in (R \cup E) \cap (R' \cup E')$. A deduce_2 step in oMKBtt can thus result in

$$N' = N \cup \{\langle s : t, \varnothing, \varnothing, R \cap R', \varnothing, \varnothing\rangle\}$$

– If $\mathsf{simplify}_2$ rewrites $s \simeq t$ to $s \simeq u$ using a rule $\ell\sigma \to r\sigma$, the system $\mathcal{C} \cup \{\ell\sigma \to r\sigma\}$ must terminate. The set N thus contains nodes $n = \langle s : t, R_0, R_1, E, C_0, C_1\rangle$ and $\langle \ell : r, \dots, E', \dots \rangle$ such that $p \in E \cap E'$. Thus $\mathsf{rewrite}_2$ applies and $(*)$ is satisfied as we obtain

$$\begin{aligned} N' = (N \setminus \{n\}) \cup \{&\langle s : t, R_0, R_1, E \setminus \{p\}, C_0, C_1\rangle\} \\ &\cup \{\langle s : u, \varnothing, \varnothing, \{p\}, \varnothing, \varnothing\rangle,\ \langle \ell\sigma : r\sigma, \varnothing, \varnothing, \varnothing, \{p\}, \varnothing\rangle\} \end{aligned}$$

– If $\mathsf{compose}_2$ rewrites $s \to t$ to $s \to u$ using a rule $\ell\sigma \to r\sigma$ then $\mathcal{C} \cup \{\ell\sigma \to r\sigma\}$ terminates and N contains nodes $n = \langle s : t, R_0, R_1, E, C_0, C_1\rangle$ and $\langle \ell : r, \dots, E', \dots \rangle$ such that $p \in R_0 \cap E'$. Thus $\mathsf{rewrite}_2$ applies and $(*)$ is satisfied since we obtain

$$\begin{aligned} N' = (N \setminus \{n\}) \cup \{&\langle s : t, R_0 \setminus \{p\}, R_1, E, C_0, C_1\rangle\} \\ &\cup \{\langle s : u, \{p\}, \varnothing, \varnothing, \varnothing, \varnothing\rangle,\ \langle \ell\sigma : r\sigma, \varnothing, \varnothing, \varnothing, \{p\}, \varnothing\rangle\} \end{aligned}$$

– Finally, if $\mathsf{collapse}_2$ replaces $t \to s$ by an equation $u \approx s$ as $t \to_{\ell\sigma\to r\sigma} u$ then $\mathcal{C} \cup \{\ell\sigma \to r\sigma\}$ must terminate and the set N has to contain nodes $n = \langle s : t, R_0, R_1, E, C_0, C_1\rangle$ and $\langle \ell : r, \dots, E', \dots \rangle$ such that $p \in R_1 \cap E'$. Again $\mathsf{rewrite}_2$ applies and $(*)$ holds as we obtain

$$\begin{aligned} N' = (N \setminus \{n\}) \cup \{&\langle s : t, R_0 \setminus \{p\}, R_1, E, C_0, C_1\rangle\} \\ &\cup \{\langle s : u, \varnothing, \varnothing, \{p\}, \varnothing, \varnothing\rangle,\ \langle \ell\sigma : r\sigma, \varnothing, \varnothing, \varnothing, \{p\}, \varnothing\rangle\} \end{aligned}$$

□

Given an oMKBtt run $\gamma\colon N_0 \vdash^n N$, we define the sequence of *ancestors* $p_0, \dots, p_n$ of a process $p \in \mathcal{P}(N)$ and the *projected run* γ_p as in Definition 4.30. We then obtain the following corollary from Lemma 5.50:

Corollary 5.52. *Suppose an* oMKBtt *run* $\gamma\colon N_0 \vdash N_1 \vdash \cdots \vdash N_k$ *gives rise to a process* $p \in \mathcal{P}(N_k)$ *having ancestors* $p_0, \dots, p_k$*. Let* $\mathcal{E}_i$*,* $\mathcal{R}_i$ *and* $\mathcal{C}_i$ *denote* $E[N_i, p_i]$*,* $R[N_i, p_i]$ *and* $C[N_i, p_i]$*, respectively. Then the sequence*

$$\gamma_p\colon (\mathcal{E}_0,\mathcal{R}_0,\mathcal{C}_0) \vdash^= (\mathcal{E}_1,\mathcal{R}_1,\mathcal{C}_1) \vdash^= \cdots \vdash^= (\mathcal{E}_n,\mathcal{R}_n,\mathcal{C}_n)$$

is a valid oKBtt *run called the* projection *of* γ *to* p*.* □

Definition 5.53. A finite oMKBtt run $\gamma\colon N_0 \vdash^* N$ is *sufficiently fair for a process* $p \in \mathcal{P}(N)$ if γ_p is sufficiently fair, and it is *sufficiently fair* if it is sufficiently fair for some process p.

An $\mathsf{oMKBtt}_{\mathbb{T}}$ run refers to an oMKBtt run employing a termination tool $\mathbb{T}$ for all termination checks in orient and $\mathsf{rewrite}_2$ steps. Let an $\mathsf{oMKBtt}_{\mathsf{total}}$ run be any run where $\mathbb{T}$ ensures *total* termination of the checked system. From Lemmas 5.50 and 5.51 and the Correctness Theorem 5.41 for $\mathsf{oKBtt}_{\mathsf{total}}$ it is then not difficult to establish a correctness result for $\mathsf{oMKBtt}_{\mathsf{total}}$.

Correctness Theorem 5.54. *Let $N_{\mathcal{E}}$ be the initial node set for a set of equations $\mathcal{E}$ and $\gamma\colon N_{\mathcal{E}} \vdash^* N$ be a finite $\mathsf{oMKBtt}_{\mathsf{total}}$ run which is sufficiently fair for some $p \in \mathcal{P}(N)$. Then there exists a ground-total extension $>$ of $\to^+_{C[N,p]}$ such that $(E[N,p], R[N,p])$ is ground convergent for $\mathcal{E}$ with respect to $>$.*

Proof. Corollary 5.52 states that $\gamma_p : (\mathcal{E}, \varnothing, \varnothing) \vdash^* (E[N,p], R[N,p], C[N,p])$ is a valid $\mathsf{oKBtt}_{\mathsf{total}}$ run which is by assumption sufficiently fair. According to Correctness Theorem 5.41 there exists a ground-total extension $>$ of $\to^+_{C[N,p]}$ such that $(E[N,p], R[N,p])$ is ground convergent for $\mathcal{E}$ with respect to $>$. □

Chapter 6

Normalized Completion Systems

Natural input problems for completion procedures often contain axiomatizations of common algebraic structures, such as associative and commutative symbols, groups, or rings. Specialized completion procedures have been proposed to deal efficiently with some customary algebraic theories, most notably with AC [65–67, 83]. Completion procedures for left-linear theories were proposed in [6, 10, 45]. For more general theories $\mathcal{T}$ where $\mathcal{T}$-unification is finitary and the subterm ordering modulo $\mathcal{T}$ is well-founded, extensions have been presented in [10, 48]. However, this requirement excludes common theories such as AC with a unit element (ACU) or AC with idempotency (ACI). For the case of ACU, a specialized completion procedure based on constrained rewriting [18, 54] has been presented in [49], but this approach does not directly generalize to other theories such as Abelian groups.

Normalized completion [73–76] constitutes the last result in this line of research. It has three advantages over earlier proposals. (1) It allows completion modulo any theory $\mathcal{T}$ that can be represented as an AC-convergent rewrite system $\mathcal{S}$. (2) Critical pairs need not be computed with respect to the theory $\mathcal{T}$, which may not have a decidable unification problem. Instead, any theory between AC and $\mathcal{T}$ can be used. (3) The reduction order used to establish termination need not be compatible with $\mathcal{T}$, it only needs to be AC compatible. This is beneficial for theories like ACI where no $\mathcal{T}$-compatible reduction order can possibly exist, as the following example shows.

Example 6.1. Consider the theory ACI of an AC operator $\cdot$ which is idempotent, i.e., it satisfies $x \cdot x \leftrightarrow^*_{\mathrm{ACI}} x$. Assume there exists an ACI-compatible reduction order $\succ$. Suppose $s \succ t$ holds for some terms s and t, such that $s \cdot s \succ t$ holds by ACI compatibility. Then the following infinitely decreasing sequence contradicts wellfoundedness:

$$s \leftrightarrow^*_{\mathrm{ACI}} s \cdot s \succ t \cdot s \leftrightarrow^*_{\mathrm{ACI}} t \cdot s \cdot s \succ t \cdot t \cdot s \leftrightarrow^*_{\mathrm{ACI}} t \cdot t \cdot s \cdot s \succ \ldots$$

For the above mentioned reasons, normalized completion is applicable to many common theories such as AC augmented with axioms for unit elements, idempotency or nilpotency, but also to groups and rings. It moreover generalizes Buchberger's algorithm to compute Gröbner bases [76]. Compared to standard completion (Section 3.2) or AC completion, it also improves efficiency if the input theory includes a subtheory for which an (AC-)convergent presentation is already known. In computing less critical pairs by focusing on a particular theory, the approach shares advantages with specialized theorem proving techniques having built-in equational theories (e.g., [43, 82, 105]).

6.1 Normalized Completion

We first recall *normalized rewriting*, mainly based on the presentation in [75] but also taking into account results in [73, 74, 76].

Throughout this chapter, we will consider an equational theory $\mathcal{T}$ over a signature $\mathcal{F}$ which is represented by a fixed AC-convergent TRS $\mathcal{S}$, so

$$\overset{*}{\underset{\mathcal{T}}{\leftrightarrow}} \;=\; \xrightarrow[\mathcal{S}/\mathrm{AC}]{!} \cdot \overset{*}{\underset{\mathrm{AC}}{\leftrightarrow}} \cdot \xleftarrow[\mathcal{S}/\mathrm{AC}]{!}$$

For example, for the theory ACU of an AC operator $+$ with unit $\mathsf{0}$, we have $\mathcal{T} = \{x{+}(y{+}z) \approx (x{+}y){+}z, x{+}y \approx y{+}x, x{+}\mathsf{0} \approx x\}$ and $\mathcal{S} = \{x{+}\mathsf{0} \to x\}$. Note that the representation $\mathcal{S}$ need not be unique, for example if $\mathcal{T}$ is the theory of Abelian groups.[1] The associativity and commutativity equations in $\mathcal{T}$ for all associative-commutative symbols in $\mathcal{F}$ are denoted by AC.

Given a set of equations $\mathcal{E}$ and a set of rewrite rules $\mathcal{R}$, an *equational proof step* $s \leftrightarrow_e^{p,\sigma} t$ in $(\mathcal{T}, \mathcal{E}, \mathcal{R})$ is either an equality step using an equation $e \in \mathcal{E}$, a rewrite step using a rule $e \in \mathcal{R}$, an ($\mathcal{S}$-) rewrite step using a rule $e \in \mathcal{S}$, or an equality step using an equation e in AC.

We define normalized rewriting as in [75] but use a different notation to distinguish it from the by now established notation for rewriting modulo.

Definition 6.2 ([75])**.** Two terms s and t admit an *$\mathcal{S}$-normalized $\mathcal{R}$-rewrite step* if

$$s \xrightarrow[\mathcal{S}/\mathrm{AC}]{!} s' \overset{*}{\underset{\mathrm{AC}}{\leftrightarrow}} \cdot \xrightarrow[\ell \to r]{p} \cdot \overset{*}{\underset{\mathrm{AC}}{\leftrightarrow}} t \tag{6.1}$$

for some rule $\ell \to r$ in $\mathcal{R}$ and position p in s'. We write

$$s \xrightarrow[\ell \to r \backslash \mathcal{S}]{p} t$$

for (6.1), and write $s \to_{\mathcal{R}\backslash\mathcal{S}} t$ if $s \xrightarrow[\ell \to r \backslash \mathcal{S}]{p} t$ for a rule $\ell \to r$ in $\mathcal{R}$ and position p.

Let $\succ$ be an AC-compatible reduction order such that $\mathcal{S} \subseteq \succ$ and let $\succeq$ be the smallest AC-compatible preorder containing $\succ$. For any set of rewrite rules $\mathcal{R}$ satisfying $\mathcal{R} \subseteq \succ$ the normalized rewrite relation $\to_{\mathcal{R}\backslash\mathcal{S}}$ is well-founded [74, 75], so we can consider equational proofs of the form

$$s \xrightarrow[\mathcal{R}\backslash\mathcal{S}]{!} \cdot \overset{*}{\underset{\mathcal{T}}{\leftrightarrow}} \cdot \xleftarrow[\mathcal{R}\backslash\mathcal{S}]{!} t$$

These normal form proofs will play a special role and are called *normalized rewrite proofs*.

By AC-convergence of $\mathcal{S}$ for $\mathcal{T}$, any such proof can be transformed into a proof $s \downarrow\!\downarrow_{\mathcal{R}\backslash\mathcal{S}} t$, where the relation $\downarrow\!\downarrow_{\mathcal{R}\backslash\mathcal{S}}$ abbreviates

$$\xrightarrow[\mathcal{R}\backslash\mathcal{S}]{!} \cdot \xrightarrow[\mathcal{S}/\mathrm{AC}]{!} \cdot \overset{*}{\underset{\mathrm{AC}}{\leftrightarrow}} \cdot \xleftarrow[\mathcal{S}/\mathrm{AC}]{!} \cdot \xleftarrow[\mathcal{R}\backslash\mathcal{S}]{!}$$

1 To avoid confusion we differentiate between the theory $\mathcal{T}$ and its AC-convergent representation $\mathcal{S}$, although both are denoted by S in [75].

A TRS $\mathcal{R}$ is called $\mathcal{S}$-*convergent for* a set of equations $\mathcal{E}$ if $\to_{\mathcal{R}\setminus\mathcal{S}}$ is terminating and the relations $\leftrightarrow^*_{\mathcal{E}\cup\mathcal{T}}$ and $\downarrow\downarrow_{\mathcal{R}\setminus\mathcal{S}}$ coincide.

From now on we write $t{\downarrow}$ for $t{\downarrow_{\mathcal{S}/\mathrm{AC}}}$ and $s{\downarrow_p}$ for $s[u{\downarrow}]_p$ where $u = s|_p$. We let $c(s,p,t)$ denote the multiset $\{s\}$ if $s{\downarrow_p} = s$ and $\{s,t\}$ otherwise.

Definition 6.3. Let $\mathcal{E}$ be a set of equations, $\mathcal{R}$ a rewrite system and $\succ$ an AC-compatible reduction order containing $\mathcal{R}$. The *cost* of an equational proof step is defined as follows:[2]

$$
\begin{aligned}
c(s \xleftrightarrow[u\approx v]{} t) &= (\bot, \{s\}, \bot, \bot, \bot) && \text{if } u \simeq v \in \mathrm{AC}\\
c(s \xleftrightarrow[u\approx v]{p} t) &= (\{s{\downarrow_p}, t{\downarrow_p}\}, \{s,t\}, \bot, \bot, \bot) && \text{if } u \simeq v \in \mathcal{E}\\
c(s \xleftrightarrow[\ell\to r]{p,\sigma} t) &= c(t \xleftrightarrow[r\leftarrow\ell]{p,\sigma} s) = (c(s,p,t), \{s\}, (s|_p){\downarrow}, \ell, r\sigma) && \text{if } \ell \to r \in \mathcal{R}\\
c(s \xleftrightarrow[\ell\to r]{} t) &= c(t \xleftrightarrow[r\leftarrow\ell]{} s) = (\bot, \{s\}, \bot, \bot, \bot) && \text{if } \ell \to r \in \mathcal{S}
\end{aligned}
$$

Cost tuples are compared by the lexicographic combination of $(\succ_{\mathsf{mul}}, (\leftrightarrow^*_{\mathrm{AC}})_{\mathsf{mul}})$ for the first two components, $(\rhd_{\mathrm{AC}}, \leftrightarrow^*_{\mathrm{AC}})$, $(\,\rhd_{\mathrm{AC}}\,, \leftrightarrow^*_{\mathrm{AC}})$, and $(\succ, \leftrightarrow^*_{\mathrm{AC}})$, all of which constitute order pairs. The constant $\bot$ is considered minimal in all of these proper orders. The cost of an equational proof is the multiset consisting of the costs of its steps. The proof order $\succcurlyeq_{\mathsf{NKB}}$ is the multiset extension of the order on proof step costs, and by $\Rightarrow^{\succ}_{\mathsf{NKB}}$ we denote the relation such that $P \Rightarrow^{\succ}_{\mathsf{NKB}} Q$ holds if and only if $P \succcurlyeq_{\mathsf{NKB}} Q$ and P and Q prove the same equation.

Note that the first component of the cost of an equality step and the third component of the cost of a rewrite step are only unique up to $(\leftrightarrow^*_{\mathrm{AC}})_{\mathsf{mul}}$ and $\leftrightarrow^*_{\mathrm{AC}}$, respectively. But since these cost components are compared by AC-compatible orderings it does not matter which representative of the respective equivalence class occurs in the cost tuple.

It is easy to show that this definition yields a proof reduction relation.

Lemma 6.4 ([75])**.** $\Rightarrow^{\succ}_{\mathsf{NKB}}$ *is a proof reduction relation.* □

We use an AC version of the Extended Critical Pair Lemma [47,48].

Lemma 6.5. *Let $\ell_1 \to r_1$ and $\ell_2 \to r_2$ be rewrite rules which admit a peak $P\colon s\ {}_{r_1\leftarrow\ell_1}\!\leftarrow \cdot \leftrightarrow^*_{\mathrm{AC}} \cdot \to_{\ell_2\to r_2} t$. If P does not contain an instance of an AC overlap then*

$$s \xrightarrow[\ell_2\to r_2/\mathrm{AC}]{*} \cdot \xleftarrow[r_1\leftarrow\ell_1/\mathrm{AC}]{*} t$$

Otherwise, there is some critical pair $u \leftarrow \times \rightarrow v$ in $\mathrm{CP}_{\mathrm{AC}}(\ell_1 \to r_1, \ell_2 \to r_2)$ or $\mathrm{CP}_{\mathrm{AC}}(\ell_1 \to r_1, (\ell_2 \to r_2)^e)$ such that

$$s \xleftrightarrow[\mathrm{AC}]{*} \cdot \xleftrightarrow[u\approx v]{} \cdot \xleftrightarrow[\mathrm{AC}]{*} t$$

2 Note that our definition differs from [75] by the additional third component, which was added to facilitate critical pair criteria.

Note that this implies that any non-joinable peak is an instance of an AC-critical pair between two rules where *at most one* rule is extended, so critical pairs between two extended rules of a rewrite system $\mathcal{R}$ can be ignored. The following lemma builds upon the previous statement and shows that both joining sequences and critical pairs admit smaller proofs.

Lemma 6.6. *Let $\mathcal{R}$ be a set of rewrite rules such that $\mathcal{R} \subseteq \succ$.*

(a) If $P\colon s\ {}_{\mathcal{S}}\!\leftarrow u \leftrightarrow^*_{\mathrm{AC}} u' \to_{\mathcal{S}} t$ then there is a proof $Q\colon s \to^*_{\mathcal{S}/\mathrm{AC}} \cdot\ {}^*_{\mathcal{S}/\mathrm{AC}}\!\leftarrow t$ such that $P \Rightarrow^{\succ}_{\mathsf{NKB}} Q$.

(b) If $P\colon s\ {}_{\mathcal{R}}\!\leftarrow u \leftrightarrow^*_{\mathrm{AC}} u' \to_{\mathcal{R}} t$ then we have $Q\colon s \to^*_{\mathcal{R}/\mathrm{AC}} \cdot\ {}^*_{\mathcal{R}/\mathrm{AC}}\!\leftarrow t$ such that $P \Rightarrow^{\succ}_{\mathsf{NKB}} Q$, or there is some critical pair $s' \leftarrow \times \to t'$ in $\mathrm{CP}_{\mathrm{AC}}(\mathcal{R}, \mathcal{R}^e)$ such that $P \Rightarrow^{\succ}_{\mathsf{NKB}} s \leftrightarrow^*_{\mathrm{AC}} \cdot \leftrightarrow_{s'\approx t'} \cdot \leftrightarrow^*_{\mathrm{AC}} t$.

(c) If $P\colon s\ {}_{\mathcal{R}}\!\leftarrow u \leftrightarrow^*_{\mathrm{AC}} u' \to_{\mathcal{S}} t$ then there is a proof $Q\colon s \to^*_{\mathcal{S}/\mathrm{AC}} \cdot\ {}^*_{\mathcal{R}/\mathrm{AC}}\!\leftarrow t$ such that $P \Rightarrow^{\succ}_{\mathsf{NKB}} Q$, or there is a critical pair $s' \leftarrow \times \to t'$ in $\mathrm{CP}_{\mathrm{AC}}(\mathcal{R}, \mathcal{S}^e) \cup \mathrm{CP}_{\mathrm{AC}}(\mathcal{S}, \mathcal{R})$ such that $P \Rightarrow^{\succ}_{\mathsf{NKB}} s \leftrightarrow^*_{\mathrm{AC}} \cdot \leftrightarrow_{s'\approx t'} \cdot \leftrightarrow^*_{\mathrm{AC}} t$.

Proof.

(a) By AC confluence of $\mathcal{S}$ there is a proof

$$Q\colon s = s_0 \xrightarrow[\mathcal{S}/\mathrm{AC}]{} \cdots \xrightarrow[\mathcal{S}/\mathrm{AC}]{} s_m \overset{*}{\underset{\mathrm{AC}}{\longleftrightarrow}} t_n \xleftarrow[\mathcal{S}/\mathrm{AC}]{} \cdots \xleftarrow[\mathcal{S}/\mathrm{AC}]{} t_0 = t$$

We have $c(P) = \{(\bot, \{u\}, \ldots),\ (\bot, \{u'\}, \ldots)\} \cup c_{\mathrm{AC}}(P)$, but as $\mathcal{S} \subseteq \succ$ the cost $c(Q)$ contains only tuples of the form $(\bot, \{v\}, \ldots)$ such that $s \succeq v$ or $t \succeq v$. From $u \succ s, t$ it follows that $P \Rightarrow^{\succ}_{\mathsf{NKB}} Q$.

(b) Let p, q be positions such that P is $s\ {}^{p}_{\mathcal{R}}\!\leftarrow u \leftrightarrow^*_{\mathrm{AC}} u' \to^{q}_{\mathcal{R}} t$. Then we have $c(P) = \{(c(u, p, s), \ldots),\ (c(u', q, t), \ldots)\} \cup c_{\mathrm{AC}}(P)$. If a proof

$$Q\colon s = s_0 \xrightarrow[\mathcal{R}/\mathrm{AC}]{p_1} \cdots \xrightarrow[\mathcal{R}/\mathrm{AC}]{p_m} s_m \overset{*}{\underset{\mathrm{AC}}{\longleftrightarrow}} t_n \xleftarrow[\mathcal{R}/\mathrm{AC}]{q_n} \cdots \xleftarrow[\mathcal{R}/\mathrm{AC}]{q_1} t_0 = t$$

exists then $c(Q)$ consists of tuples $(c(s'_i, p_{i+1}, s'_{i+1}), \ldots)$ for terms s'_i such that $s_i \leftrightarrow^*_{\mathrm{AC}} s'_i$ for all $0 \leqslant i < m$, together with tuples $(c(t'_j, q_{j+1}, t'_{j+1}), \ldots)$ for terms t'_j such that $t_j \leftrightarrow^*_{\mathrm{AC}} t'_j$ for all $0 \leqslant j < n$, plus some part $c_{\mathrm{AC}}(Q)$ which only contains cost tuples of the form $(\bot, \ldots)$. But as $\mathcal{R} \subseteq \succ$ and $\succ$ is AC compatible, $u \succ s'_i$ and $u \succ t'_j$ holds for all terms s'_i and t'_j occurring in $c(Q)$. Therefore $(c(u, p, s), \ldots) \in c(P)$ is greater than all cost tuples in $c(Q)$, which entails $P \Rightarrow^{\succ}_{\mathsf{NKB}} Q$.

If no such proof Q exists then by Lemma 6.5 there is some AC-critical pair $s' \leftarrow \times \to t'$ in $\mathrm{CP}_{\mathrm{AC}}(\mathcal{R}, \mathcal{R}^e)$ such that $s \leftrightarrow^*_{\mathrm{AC}} C[s'\sigma]$ and $t \leftrightarrow^*_{\mathrm{AC}} C[t'\sigma]$. Thus the proof $Q'\colon s \leftrightarrow^*_{\mathrm{AC}} C[s'\sigma] \leftrightarrow_{s'\approx t'} C[t'\sigma] \leftrightarrow^*_{\mathrm{AC}} t$ is valid and has cost $c(Q') = \{(\{C[s'\sigma]{\downarrow}, C[t'\sigma]{\downarrow}\}, \ldots)\} \cup c_{\mathrm{AC}}(Q')$. From $u \succ s, t$ and AC compatibility we infer $u \succ C[s'\sigma] \succ C[s'\sigma]{\downarrow}$ and $u \succ C[t'\sigma] \succ C[t'\sigma]{\downarrow}$, hence $(c(u, p, s), \ldots) \in c(P)$ is greater than all cost tuples in Q' such that $P \Rightarrow^{\succ}_{\mathsf{NKB}} Q'$.

(c) Let p, q be positions such that P is $s \overset{p}{\underset{\mathcal{R}}{\leftarrow}} u \leftrightarrow^*_{\mathrm{AC}} u' \overset{q}{\underset{\mathcal{S}}{\rightarrow}} t$. Then we have $c(P) = \{(c(u,p,s),\ldots),\ (\bot,\{u'\},\ldots)\} \cup c_{\mathrm{AC}}(P)$. If a proof

$$Q\colon s = s_0 \xrightarrow[\mathcal{S}/\mathrm{AC}]{p_1} \cdots \xrightarrow[\mathcal{S}/\mathrm{AC}]{p_m} s_m \xleftrightarrow[\mathrm{AC}]{*} t_n \xleftarrow[\mathcal{R}/\mathrm{AC}]{q_n} \cdots \xleftarrow[\mathcal{R}/\mathrm{AC}]{q_1} t_0 = t$$

exists then $c(Q)$ consists of tuples $(\bot,\{s'_i\},\ldots)$ such that $s_i \leftrightarrow^*_{\mathrm{AC}} s'_i$ for $0 \leqslant i < m$, together with tuples $(c(t'_j, q_{j+1}, t'_{j+1}),\ldots)$ such that $t_j \leftrightarrow^*_{\mathrm{AC}} t'_j$ for all $0 \leqslant j < n$, plus some part $c_{\mathrm{AC}}(Q)$. As $\mathcal{R} \subseteq \succ$ we have $u \succ t_j$ and by AC compatibility $u \succ t'_j$. Thus $c(u,p,s) \succ^{\mathsf{mul}} c(t'_j, q_{j+1}, t'_{j+1})$ for all $0 \leqslant j < n$. Since moreover $(c(u,p,s),\ldots)$ dominates the cost of all AC and $\mathcal{S}$-steps, it follows that $P \Rightarrow^{\succ}_{\mathsf{NKB}} Q$.

If no such proof Q exists then by Lemma 6.5 there is some AC-critical pair $s' \leftarrow \times \rightarrow t' \in \mathrm{CP}_{\mathrm{AC}}(\mathcal{R}, \mathcal{S}^e) \cup \mathrm{CP}_{\mathrm{AC}}(\mathcal{S}, \mathcal{R}^e)$ such that $s \leftrightarrow^*_{\mathrm{AC}} C[s'\sigma]$ and $t \leftrightarrow^*_{\mathrm{AC}} C[t'\sigma]$. Thus the proof $Q'\colon s \leftrightarrow^*_{\mathrm{AC}} C[s'\sigma] \leftrightarrow_{s' \approx t'} C[t'\sigma] \leftrightarrow^*_{\mathrm{AC}} t$ is valid and has cost $c(Q') = \{(\{C[s'\sigma], C[t'\sigma]\},\ldots)\} \cup c_{\mathrm{AC}}(Q')$. As $u \succ s, t$ we also have $u \succ C[s'\sigma], C[t'\sigma]$ by AC compatibility, so the tuple $(c(u,p,s),\ldots)$ dominates $c(Q')$ and we have $P \Rightarrow^{\succ}_{\mathsf{NKB}} Q'$. □

Note that in Lemma 6.6(c) it suffices to consider the AC-critical pairs in $\mathrm{CP}_{\mathrm{AC}}(\mathcal{R}, \mathcal{S}^e) \cup \mathrm{CP}_{\mathrm{AC}}(\mathcal{S}, \mathcal{R})$ (and thus ignore $\mathrm{CP}_{\mathrm{AC}}(\mathcal{S}, \mathcal{R}^e)$): by Lemma 6.5 extended rules are only necessary to cover the cases where two rules $\ell_1 \to r_1$ and $\ell_2 \to r_2$ are rooted by the same AC-symbol. Then all required critical pairs can be obtained by from overlaps $\langle \ell_1 \to r_1, p, (\ell_2 \to r_2)^e \rangle$, or, equivalently, from overlaps $\langle \ell_2 \to r_2, p, (\ell_1 \to r_1)^e \rangle$.

The notion of *$\mathcal{S}$-normalizing pairs* is crucial to normalized completion.[3]

Definition 6.7. Let $\mathcal{E}$ be a set of equations, $\mathcal{R}$ be a set of rewrite rules and u, v be terms such that $u \simeq v \in \mathcal{E}$. Let furthermore Θ and Ψ be functions such that $\Theta(u,v)$ is a set of equations and $\Psi(u,v)$ is a set of rewrite rules. Then (Θ, Ψ) constitutes an *$\mathcal{S}$-normalizing pair* for u and v if

(i) $\Theta(u,v)$ and $\Psi(u,v)$ are contained in $\leftrightarrow^*_{\mathcal{E} \cup \mathcal{R} \cup \mathcal{T}}$ and $\Psi(u,v) \subseteq \succ$,

(ii) for every equational proof P of the shape $s \xleftrightarrow[u \approx v]{\epsilon, \sigma} t$ there exists a proof Q in $(\mathcal{T}, \Theta(u,v), \Psi(u,v))$ such that $P \Rightarrow^{\succ}_{\mathsf{NKB}} Q$, and

(iii) for all rules $\ell \to r$ in $\Psi(u,v)$, all sets of rewrite rules $\mathcal{R}$ and all proofs P of the form $s\ \mathcal{S}\leftarrow\ w \leftrightarrow^*_{\mathrm{AC}} \cdot \to_{\ell \to r} \cdot \to^*_{\mathcal{R} \setminus \mathcal{S}}\ t$ there is a proof Q in $(\mathcal{T}, \Theta(u,v), \Psi(u,v) \cup \mathcal{R})$ such that $P \Rightarrow Q$ and all terms in Q are smaller than w.

Here condition (i) ensures that soundness and termination are preserved. Condition (ii) requires that all proofs using the equation $u \approx v$ can be replaced by smaller proofs, which is often achieved by adding the rule $u \to v$. Condition (iii) takes AC overlaps between rules in $\Psi(u,v)$ and $\mathcal{S}$ into account, but

3 The definition of normalizing pairs varies in the literature; the first reference in [73, Definition 4.4] is different from [75, Definition 3.5] and [76, Definition 3.1]. But none of these definitions allowed us to understand and reproduce the correctness proof (cf. Remark 6.18), thus we use a different notion.

since rules in $\Psi(u, v)$ may at a later stage get composed with other rules, the considered peaks take a more general shape.

Example 6.8. Take the theory ACU where $\mathcal{S} = \{x + \mathsf{0} \to x\}$ and consider the $\mathcal{S}$-normalized terms $u = -(x + y)$ and $v = (-x) + (-y)$. Let $\succ$ be an AC-RPO. If the precedence is $- \succ + \succ \mathsf{0}$, we have $u \succ v$. Then $\Theta(u, v) = \{-x \approx (-\mathsf{0}) + (-x)\}$ and $\Psi(u, v) = \{u \to v\}$ form a valid normalizing pair: Condition (i) is clearly satisfied. Also condition (ii) holds as any proof using $s \approx t$ can be transformed into a proof using $u \to v$ which is smaller by Definition 6.3 as $u = u{\downarrow}$. Finally, using Lemma 6.6 it is not hard to see that by adding the AC-critical pair in $\Theta(u, v)$ also condition (iii) holds. If the precedence is $+ \succ - \succ \mathsf{0}$ such that $v \succ u$, one may simply take $\Theta(v, u) = \varnothing$ and $\Psi(v, u) = \{v \to u\}$.[4]

Marché proposes a *general $\mathcal{S}$-normalizing pair* [75, Definition 3.9] which is applicable for any choice of the theory $\mathcal{S}$, where Ψ simply returns the oriented term pair $u \to v$ and Θ returns AC-critical pairs between $u \to v$ and a rule in $\mathcal{S}$. We show that this definition is also a normalizing pair according to Definition 6.7.

Definition 6.9. Let u and v be terms in $\mathcal{S}$-normal form such that $u \succ v$. The *general normalizing pair* $(\Theta_{\mathrm{gen}}, \Psi_{\mathrm{gen}})$ is defined by $\Psi_{\mathrm{gen}}(u, v) = \{u \to v\}$ and $\Theta_{\mathrm{gen}}(u, v) = \mathrm{CP}_{\mathrm{AC}}(u \to v, \mathcal{S}^e) \cup \mathrm{CP}_{\mathrm{AC}}(\mathcal{S}, u \to v)$.

Lemma 6.10. *If u and v are terms in $\mathcal{S}$-normal form such that $u \succ v$ then $(\Theta_{\mathrm{gen}}, \Psi_{\mathrm{gen}})$ forms a normalizing pair for u and v.*

Proof. We argue that $\Theta_{\mathrm{gen}}(u, v)$ and $\Psi_{\mathrm{gen}}(u, v)$ satisfy the three requirements demanded in Definition 6.7. Condition (i) is satisfied as due to $u \simeq v \in \mathcal{E}$ both $\Theta_{\mathrm{gen}}(u, v)$ and $\Psi_{\mathrm{gen}}(u, v)$ are contained in $\leftrightarrow^*_{\mathcal{E} \cup \mathcal{T}}$, and $\Psi_{\mathrm{gen}}(u, v) \subseteq \succ$ as $u \succ v$.

Concerning condition (ii), any proof $P\colon s \leftrightarrow^{\epsilon,\sigma}_{u \approx v} t$ can be transformed into $Q\colon s \leftrightarrow^{\epsilon,\sigma}_{u \to v} t$. We obtain the decrease $u \leftrightarrow^{\epsilon}_{u \approx v} v \Rightarrow^{\succ}_{\mathsf{NKB}} u \leftrightarrow^{\epsilon}_{u \to v} v$ because $\{(\{u, v\}, \ldots)\} \succ\!\!\succ_{\mathsf{NKB}} \{(\{u\}, \ldots)\}$. As $\Rightarrow^{\succ}_{\mathsf{NKB}}$ is a proof reduction relation also $P \Rightarrow^{\succ}_{\mathsf{NKB}} Q$ holds.

Finally, consider a proof P of the form $s \; {}_{\mathcal{S}}\!\leftarrow w \leftrightarrow^*_{\mathrm{AC}} w' \to^p_{u \to v} t \to_{\mathcal{R} \setminus \mathcal{S}} \hat{t}$. By Lemma 6.6, there exists a smaller proof of $s \approx t$ (and thus also of $s \approx \hat{t}$) if the peak in P does not constitute a proper overlap. Otherwise P must contain an instance of an AC-critical peak, so $s \leftrightarrow^*_{\mathrm{AC}} C[s'\sigma]$ and $t \leftrightarrow^*_{\mathrm{AC}} C[t'\sigma]$ for some context C, substitution σ, and AC-critical pair $s' \leftarrow \times \to t'$. According to Lemma 6.5 we may assume that one rule comes from $\mathcal{S}^e$ and one rule comes from $\mathcal{R}$. Hence $s' \simeq t' \in \Theta_{\mathrm{gen}}(u, v)$, which gives rise to the proof Q of the form $s \leftrightarrow^*_{\mathrm{AC}} \cdot \leftrightarrow_{s' \approx t'} \cdot \leftrightarrow^*_{\mathrm{AC}} t \to_{\mathcal{R} \setminus \mathcal{S}} \hat{t}$. We have $c(P) = \{(\bot, \{w\}, \ldots), (c(w', p, t), \ldots)\} \cup c_{\mathrm{AC}}(P) \cup c(P')$ for $P'\colon t \to_{\mathcal{R} \setminus \mathcal{S}} \hat{t}$, whereas $c(Q) = \{(\{C[s'\sigma]{\downarrow}, C[t'\sigma]{\downarrow}\}, \ldots)\} \cup c_{\mathrm{AC}}(Q) \cup c(P')$. We have $w' \succ t$, and by AC compatibility also $w' \succ s$, $w' \succ C[s'\sigma] \succ C[s'\sigma]{\downarrow}$, and $w' \succ C[t'\sigma] \succ C[t'\sigma]{\downarrow}$. Thus $(c(w', p, t), \ldots) \in c(P)$ is greater than all cost tuples in $c(Q)$, so $P \Rightarrow^{\succ}_{\mathsf{NKB}} Q$. This shows that also condition (iii) is satisfied. □

4 These normalizing pairs are actually instances of ACU-normalizing pairs as suggested in Definition 6.22.

orient	$\dfrac{\mathcal{E} \uplus \{s \simeq t\}, \mathcal{R}}{\mathcal{E} \cup \Theta(s,t), \mathcal{R} \cup \Psi(s,t)}$	if $s = s{\downarrow}$ and $t = t{\downarrow}$
deduce	$\dfrac{\mathcal{E}, \mathcal{R}}{\mathcal{E} \cup \{s \approx t\}, \mathcal{R}}$	if $s \leftarrow \rtimes \rightarrow t \in \mathrm{CP}_{\mathcal{L}}(\mathcal{R}, \mathcal{R}^e)$
delete	$\dfrac{\mathcal{E} \uplus \{s \approx t\}, \mathcal{R}}{\mathcal{E}, \mathcal{R}}$	if $s \leftrightarrow^*_{\mathrm{AC}} t$
normalize	$\dfrac{\mathcal{E} \uplus \{s \approx t\}, \mathcal{R}}{\mathcal{E} \cup \{s{\downarrow} \approx t{\downarrow}\}, \mathcal{R}}$	if $s \neq s{\downarrow}$ or $t \neq t{\downarrow}$
simplify	$\dfrac{\mathcal{E} \uplus \{s \simeq t\}, \mathcal{R}}{\mathcal{E} \cup \{s \simeq u\}, \mathcal{R}}$	if $t \to_{\mathcal{R} \setminus \mathcal{S}} u$
compose	$\dfrac{\mathcal{E}, \mathcal{R} \uplus \{s \to t\}}{\mathcal{E}, \mathcal{R} \cup \{s \to u\}}$	if $t \to_{\mathcal{R} \setminus \mathcal{S}} u$
collapse	$\dfrac{\mathcal{E}, \mathcal{R} \uplus \{t \to s\}}{\mathcal{E} \cup \{u \approx s\}, \mathcal{R}}$	if $t \xrightarrow[\ell \to r \setminus \mathcal{S}]{p,\sigma} u$ for $\ell \to r \in \mathcal{R}$ with $t \rhd_{\mathrm{AC}} \ell$ or $s \succ r\sigma$

Figure 6.1: The inference system NKB of normalized completion.

In Figure 6.1 we recall the inference system of normalized completion [75]. In the deduce rule, $\mathcal{L}$ denotes some theory such that $\mathrm{AC} \subseteq \mathcal{L} \subseteq \mathcal{T}$ holds.[5] In the orient rule, (Θ, Ψ) is assumed to form an $\mathcal{S}$-normalizing pair for the terms s and t.

An NKB inference sequence $(\mathcal{E}_0, \mathcal{R}_0) \vdash_{\mathsf{NKB}} (\mathcal{E}_1, \mathcal{R}_1) \vdash_{\mathsf{NKB}} (\mathcal{E}_2, \mathcal{R}_2) \vdash_{\mathsf{NKB}} \cdots$ is called a *run*. We will in the sequel assume that $\mathcal{R}_0 = \varnothing$ although all results in the remainder of this section generalize to the setting where $\mathcal{R}_0$ is non-empty, provided that for every rule $\ell \to r$ in $\mathcal{R}_0$ we have $\ell \succ r$ and $\ell = \ell{\downarrow}$. Again we let persistent equations $\mathcal{E}_\omega$ and rules $\mathcal{R}_\omega$ be defined by $\mathcal{E}_\omega = \bigcup_i \bigcap_{j>i} \mathcal{E}_j$ and $\mathcal{R}_\omega = \bigcup_i \bigcap_{j>i} \mathcal{R}_j$. A run *fails* if $\mathcal{E}_\omega$ is non-empty, it *succeeds* if $\mathcal{E}_\omega$ is empty and $\mathcal{R}_\omega$ is $\mathcal{S}$-convergent for $\mathcal{E}_0$.

Lemma 6.11. *Every* NKB *run constitutes an equational inference sequence with respect to* $\Rightarrow^{\succ}_{\mathsf{NKB}}$ *and* $\mathcal{T}$.

Proof. We show that in a run $(\mathcal{E}_0, \varnothing) \vdash_{\mathsf{NKB}} (\mathcal{E}_1, \mathcal{R}_1) \vdash_{\mathsf{NKB}} (\mathcal{E}_2, \mathcal{R}_2) \vdash_{\mathsf{NKB}} \cdots$ every inference step can be modeled by expand and contract inferences according to Definition 3.1.

– An orient step first expands by adding $\Theta(s,t)$ and $\Psi(s,t)$. These expansions are valid due to Definition 6.7(i). Afterwards a contract step removes

5 Thus if $\mathcal{T}$ itself is not decidable and finitary with respect to unification, one can simply use AC for $\mathcal{L}$. On the other hand it can also be beneficial to set $\mathcal{L}$ to some larger theory, for instance the set of unifiers obtained from ACU or ACUI unification are typically much smaller than those obtained from AC unification.

$s \simeq t$ from $\mathcal{E}$, which is justified because by Definition 6.7(ii) $\Theta(s,t)$ and $\Psi(s,t)$ admit a smaller proof of $s \leftrightarrow^{\epsilon}_{s\approx t} t$.

- A deduce step is clearly an expand inference for any choice of $\mathcal{L}$ because $\mathcal{L} \subseteq \mathcal{T}$.
- A delete step is a contraction as the proof $s \leftrightarrow_{s\approx t} t$ with cost $(\{s{\downarrow}, t{\downarrow}\}, \ldots)$ can be reduced to $s \leftrightarrow^*_{\mathrm{AC}} t$ where all steps have a cost of the shape $(\bot, \ldots)$.
- A normalize step can be viewed as an expansion adding $s{\downarrow} \approx t{\downarrow}$ followed by a contraction removing $s \approx t$, performing the proof transformation from $P\colon s \leftrightarrow_{s\approx t} t$ to the proof $Q\colon s \to^*_{\mathcal{S}/\mathrm{AC}} s{\downarrow} \leftrightarrow_{s{\downarrow}\approx t{\downarrow}} t{\downarrow} \;{}^*_{\mathcal{S}/\mathrm{AC}}\!\leftarrow t$. We have $c(P) = \{(\{s{\downarrow}, t{\downarrow}\}, \{s,t\}, \ldots)\}$, and $c(Q) = \{(\{s{\downarrow}, t{\downarrow}\}, \{s{\downarrow}, t{\downarrow}\}, \ldots)\} \cup c_{\mathcal{S}\cup\mathrm{AC}}(Q)$, where $c_{\mathcal{S}\cup\mathrm{AC}}(Q)$ collects the cost of all AC and $\mathcal{S}$-steps in Q, and hence only contains tuples of the form $(\bot, \ldots)$. Because $s \succeq s{\downarrow}$ and $t \succeq t{\downarrow}$ and at least one inequality is strict we have $c(P) \succnsim_{\mathsf{NKB}} c(Q)$.
- A simplify step is viewed as an expansion adding $s \simeq u$ followed by a contraction removing $s \simeq t$. This corresponds to the proof transformation where $P\colon s \leftrightarrow_{s\approx t} t$ is replaced by $Q\colon s \leftrightarrow_{s\approx u} u \;{}_{\mathcal{R}\setminus\mathcal{S}}\!\leftarrow t$. We have $c(P) = \{(\{s{\downarrow}, t{\downarrow}\}, \ldots)\}$ and $c(Q) = \{(\{s{\downarrow}, u{\downarrow}\}, \ldots), (\{t{\downarrow}\}, \ldots)\} \cup c_{\mathcal{S}\cup\mathrm{AC}}(Q)$,[6] where the cost $c_{\mathcal{S}\cup\mathrm{AC}}(Q)$ of all AC and $\mathcal{S}$-steps in Q consists only of tuples $(\bot, \ldots)$. Because of $t{\downarrow} \to_{\mathcal{R}} u \to^*_{\mathcal{S}/\mathrm{AC}} u{\downarrow}$ we have $t{\downarrow} \succ u{\downarrow}$, and so $c(P) \succnsim_{\mathsf{NKB}} c(Q)$.
- A compose step is viewed as an expansion adding $s \to u$ and a contraction removing $s \to t$. The latter corresponds to the proof transformation replacing $P\colon s \leftrightarrow^{\epsilon}_{s\to t} t$ by $Q\colon s \leftrightarrow^{\epsilon}_{s\to u} u \;{}^{p}_{\mathcal{R}\setminus\mathcal{S}}\!\leftarrow t$. We have $c(P) = \{(c(s,\epsilon,t), \{s\}, s{\downarrow}, s, t)\}$ and $c(Q) = \{(c(s,\epsilon,u), \{s\}, s{\downarrow}, s, u), (\{t{\downarrow}\}, \ldots)\} \cup c_{\mathcal{S}\cup\mathrm{AC}}(Q)$, so $c(P) \succnsim_{\mathsf{NKB}} c(Q)$ because $s \succ t \succeq t{\downarrow} \succ u$ implies $c(s,\epsilon,t) \succeq_{\mathsf{mul}} c(s,\epsilon,u)$ and $c(s,\epsilon,t) \succeq_{\mathsf{mul}} \{t{\downarrow}\}$.
- Finally, a collapse step constitutes an expansion adding $u \approx s$ followed by a contraction removing $t \to s$. This corresponds to the proof transformation replacing $P\colon t \leftrightarrow^{\epsilon}_{t\to s} s$ by $Q\colon t \to^{p,\sigma}_{\ell\to r\setminus\mathcal{S}} u \leftrightarrow_{u\approx s} s$. We have $c(P) = \{(c(t,\epsilon,s), \{t\}, t{\downarrow}, t, s)\}$ whereas the cost of Q amounts to $c(Q) = \{(\{t{\downarrow}\}, \{t{\downarrow}\}, t'|_p, \ell, r\sigma), (\{s{\downarrow}, u{\downarrow}\}, \ldots)\} \cup c_{\mathcal{S}\cup\mathrm{AC}}(Q)$ for some term t' such that $t' \leftrightarrow^*_{\mathrm{AC}} t{\downarrow}$. Note that $t'|_p = (t'|_p){\downarrow}$ because t' is an $\mathcal{S}/\mathrm{AC}$-normal form. From $t \succeq t{\downarrow}$ we obtain $c(t,\epsilon,s) \succeq_{\mathsf{mul}} \{t{\downarrow}\}$, and $t \succ s \succeq s{\downarrow}$, $t \succ u \succ u{\downarrow}$ imply $c(t,\epsilon,s) \succ_{\mathsf{mul}} \{s{\downarrow}, u{\downarrow}\}$. As $t{\downarrow} \trianglerighteq_{\mathrm{AC}} t'|_p$, and $t \rhd_{\mathrm{AC}} \ell$ or $s \succ r\sigma$ by assumption, we have $c(P) \succnsim_{\mathsf{NKB}} c(Q)$. □

We use the following notion of fairness to characterize runs that allow to simplify proofs which are not in normal form:

Definition 6.12. A nonfailing NKB run $(\mathcal{E}_0, \mathcal{R}_0) \vdash (\mathcal{E}_1, \mathcal{R}_1) \vdash (\mathcal{E}_2, \mathcal{R}_2) \vdash \cdots$ is *fair* if for any proof P in $\mathcal{T} \cup \mathcal{R}_\omega$ which is not a rewrite proof there is a proof Q in $(\mathcal{T}, \mathcal{E}_i, \mathcal{R}_i)$ for some i such that $P \Rightarrow^{\succ}_{\mathsf{NKB}} Q$.

6 Note that the cost $(\{t{\downarrow}\}, \ldots)$ is only unique up to equivalence $(\leftrightarrow^*_{\mathrm{AC}})_{\mathsf{mul}}$, but this does not cause a problem since $(\succ)_{\mathsf{mul}}$ is compatible with $(\leftrightarrow^*_{\mathrm{AC}})_{\mathsf{mul}}$.

We show that the original definition [75] constitutes a sufficient criterion for fairness in our sense. Beforehand, we state two auxiliary results about persistent rules and $\mathcal{L}$-critical pairs.

Lemma 6.13. *Assume an* NKB *run has a persistent rule $\ell \to r \in \mathcal{R}_\omega$ giving rise to a peak $P\colon s \mathrel{{}_{\mathcal{S}}\!\leftarrow} w \leftrightarrow^*_{\mathrm{AC}} w' \to^p_{\ell\to r\setminus\mathcal{S}} t$. Then there is a proof P' in $(\mathcal{T}, \mathcal{E}_\omega, \mathcal{R}_\omega)$ such that $P \Rightarrow^{\succ}_{\mathsf{NKB}} P'$, and for all $(T, \ldots) \in c(P')$ the set T contains only terms which are smaller than w.*

Proof. Let $(\mathcal{E}_0, \varnothing) \vdash (\mathcal{E}_1, \mathcal{R}_1) \vdash (\mathcal{E}_2, \mathcal{R}_2) \vdash \cdots$ be the run under consideration. A simple inductive argument shows that there must be some $i \geqslant 0$ such that a rule $\ell \to r' \in \Psi(s', t')$ is added in an orient step $(\mathcal{E}_i, \mathcal{R}_i) \vdash_{\mathsf{NKB}} (\mathcal{E}_{i+1}, \mathcal{R}_{i+1})$ applied to some equation $s' \simeq t' \in \mathcal{E}_i$, and $r' \to^*_{\mathcal{R}_\infty\setminus\mathcal{S}} r$ for $\mathcal{R}_\infty = \bigcup_j \mathcal{R}_j$. Note that $\ell \succ r'$ holds by Definition 6.7(i). Let $u \to v$ be the rule used in the step $w \to_{\mathcal{S}} s$. As $\ell \to r$ and $\ell \to r'$ have the same left-hand side there is also an AC-critical peak $_{v\leftarrow u}\!\leftarrow w \leftrightarrow^*_{\mathrm{AC}} w' \to_{\ell\to r'}$. According to Definition 6.7(iii), the peak $Q\colon s \mathrel{{}_{v\leftarrow u}\!\leftarrow} w \leftrightarrow^*_{\mathrm{AC}} w' \to_{\ell\to r'} \cdot \to^*_{\mathcal{R}_\infty\setminus\mathcal{S}} t$ can be transformed into a smaller proof Q' in $(\mathcal{T}, \Theta(s', t'), \Psi(s', t') \cup \mathcal{R}_\infty)$ which contains only terms that are smaller than w. As $(c(w', p, t), \ldots) \in c(P)$ and $w' \leftrightarrow^*_{\mathrm{AC}} w$ we have $P \Rightarrow^{\succ}_{\mathsf{NKB}} Q'$. By the Persistence Lemma 3.4 there is some proof P' in $(\mathcal{T}, \mathcal{E}_\omega, \mathcal{R}_\omega)$ such that $Q' \ (\Rightarrow^{\succ}_{\mathsf{NKB}})^= P'$, and by the definition of $\Rightarrow^{\succ}_{\mathsf{NKB}}$ for all tuples $(T, \ldots) \in c(P')$ the set T can still only contain terms smaller than w. Hence $P \Rightarrow^{\succ}_{\mathsf{NKB}} P'$. □

Lemma 6.14. *Let $\ell_1 \to r_1$ and $\ell_2 \to r_2$ be rewrite rules and $\mathrm{AC} \subseteq \mathcal{L} \subseteq \mathcal{T}$. If $s \simeq t \in \mathrm{CP}_{\mathrm{AC}}(\ell_1 \to r_1, \ell_2 \to r_2)$ then there is some critical pair $s' \leftarrow \times \to t' \in \mathrm{CP}_{\mathcal{L}}(\ell_1 \to r_1, \ell_2 \to r_2)$ and substitution ρ such that $s \leftrightarrow^*_{\mathcal{T}} s'\rho$ and $t \leftrightarrow^*_{\mathcal{T}} t'\rho$.*

Proof. If $s \simeq t \in \mathrm{CP}_{\mathrm{AC}}(\ell_1 \to r_1, \ell_2 \to r_2)$ then there must be an AC overlap $\langle \ell_1 \to r_1, p, \ell_2 \to r_2\rangle_\Sigma$ and a substitution $\sigma \in \Sigma$ such that $s \mathrel{{}_{r_1\leftarrow\ell_1}^{p,\sigma}\!\leftarrow} u \leftrightarrow^*_{\mathrm{AC}} u' \to^\sigma_{\ell_2\to r_2} t$. As $\mathrm{AC} \subseteq \mathcal{L}$ also $u \leftrightarrow^*_{\mathcal{L}} u'$ holds, so there must be an $\mathcal{L}$-overlap $\langle \ell_1 \to r_1, p, \ell_2 \to r_2\rangle_T$. As T is a complete set of unifiers there is some $\tau \in T$ and substitution ρ such that $x\sigma \leftrightarrow^*_{\mathcal{L}} x\tau\rho$ for all $x \in \mathcal{V}$. For the terms s' and t' corresponding to the peak $s' \mathrel{{}_{r_1\leftarrow\ell_1}^{p,\tau}\!\leftarrow} w \leftrightarrow^*_{\mathcal{L}} w' \to^\tau_{\ell_2\to r_2} t'$ we thus have $u \leftrightarrow^*_{\mathcal{L}} w\rho$. Therefore $s = u[r_1\sigma]_p \leftrightarrow^*_{\mathcal{L}} w\rho[r_1\tau\rho]_p = (w[r_1\tau]_p)\rho = s'\rho$ and $t = r_2\sigma \leftrightarrow^*_{\mathcal{L}} r_2\tau\rho = t'\rho$. It follows that $s \leftrightarrow^*_{\mathcal{T}} s'\rho$ and $t \leftrightarrow^*_{\mathcal{T}} t'\rho$ because $\mathcal{L} \subseteq \mathcal{T}$. □

Lemma 6.15. *A nonfailing* NKB *run satisfying $\mathrm{CP}_{\mathcal{L}}(\mathcal{R}_\omega, \mathcal{R}^e_\omega) \subseteq \bigcup_i \mathcal{E}_i$ is fair.*

Proof. Let γ be a run $(\mathcal{E}_0, \varnothing) \vdash (\mathcal{E}_1, \mathcal{R}_1) \vdash (\mathcal{E}_2, \mathcal{R}_2) \vdash \cdots$ such that $\mathcal{E}_\omega = \varnothing$ and $\mathrm{CP}_{\mathcal{L}}(\mathcal{R}_\omega, \mathcal{R}^e_\omega) \subseteq \bigcup_i \mathcal{E}_i$. We show that every minimal proof in $\mathcal{T} \cup \mathcal{R}_\omega$ is a normalized rewrite proof. Assume to the contrary that P is a minimal proof in $\mathcal{T} \cup \mathcal{R}_\omega$ which is not a rewrite proof. Thus P contains (i) a peak $s \mathrel{{}_{\mathcal{R}_\omega}\!\leftarrow} \cdot \leftrightarrow^*_{\mathrm{AC}} \cdot \to_{\mathcal{R}_\omega} t$, or (ii) a peak $s \mathrel{{}_{\mathcal{R}_\omega/\mathrm{AC}}\!\leftarrow} \cdot \to_{\mathcal{S}/\mathrm{AC}} t$ or $s \mathrel{{}_{\mathcal{S}/\mathrm{AC}}\!\leftarrow} \cdot \to_{\mathcal{R}_\omega/\mathrm{AC}} t$, or (iii) a subproof $u \to_{\mathcal{R}_\omega/\mathrm{AC}} t$ such that $u \neq u{\downarrow}$, or (iv) a peak $s \mathrel{{}_{\mathcal{S}/\mathrm{AC}}\!\leftarrow} \cdot \to_{\mathcal{S}/\mathrm{AC}} t$. For each of these cases we show that a smaller proof exists, contradicting minimality of P.

If a peak of the form (i) originates from a non-overlap then by Lemma 6.6(b) it could be replaced by a smaller proof. Otherwise, by Lemma 6.5 the peak

$s \mathrel{\mathcal{R}_\omega}\!\leftarrow \cdot \leftrightarrow^*_{\mathrm{AC}} \cdot \rightarrow_{\mathcal{R}_\omega} t$ must satisfy $s \leftrightarrow^*_{\mathrm{AC}} C[s'\sigma]$ and $t \leftrightarrow^*_{\mathrm{AC}} C[t'\sigma]$ for some critical pair $s' \leftarrow\!\!\times\!\!\rightarrow t'$ in $\mathrm{CP}_{\mathrm{AC}}(\mathcal{R}_\omega, \mathcal{R}^e_\omega)$. Assume $s' \leftarrow\!\!\times\!\!\rightarrow t'$ originates from a peak $P'\colon s' \mathrel{\mathcal{R}^p_\omega}\!\leftarrow w \leftrightarrow^*_{\mathrm{AC}} w' \rightarrow^q_{\mathcal{R}^e_\omega} t'$. We show that $\mathcal{T} \cup \mathcal{R}_\omega$ admits a smaller proof than P', which entails the existence of a smaller proof than P. By Lemma 6.14 there must also be an $\mathcal{L}$-critical pair $s'' \approx t''$ such that $s' \leftrightarrow^*_{\mathcal{T}} s''\rho$ and $t' \leftrightarrow^*_{\mathcal{T}} t''\rho$ for some substitution ρ. As $\mathcal{S}$ is AC convergent for $\mathcal{T}$, s' and $s''\rho$ as well as t' and $t''\rho$ have the same $\mathcal{S}$-normal forms, which we denote by $\hat{s}$ and $\hat{t}$, respectively. We have $c(P') = \{(c(w,p,s'),\ldots),(c(w',q,t'),\ldots)\} \cup c_{\mathrm{AC}}(P')$ while the proof $Q\colon s' \leftrightarrow^*_{\mathcal{S}\cup\mathrm{AC}} s''\rho \leftrightarrow_{s''\approx t''} t''\rho \leftrightarrow^*_{\mathcal{S}\cup\mathrm{AC}} t'$ has cost $c(Q) = \{(\{\hat{s},\hat{t}\},\ldots)\} \cup c_{\mathcal{S}\cup\mathrm{AC}}(Q)$, so $P' \Rightarrow^{\succ}_{\mathsf{NKB}} Q$ holds because $w \succ s' \succ \hat{s}$ and $w' \succ t' \succ \hat{t}$. As $\mathrm{CP}_{\mathcal{L}}(\mathcal{R}_\omega, \mathcal{R}^e_\omega) \subseteq \bigcup_i \mathcal{E}_i$ the proof Q actually exists in some $(\mathcal{T}, \mathcal{E}_i, \mathcal{R}_i)$. By the Persistence Lemma 3.4 there is also a proof Q' in $\mathcal{T} \cup \mathcal{R}_\omega$ such that $P' \Rightarrow^{\succ}_{\mathsf{NKB}} Q\ (\Rightarrow^{\succ}_{\mathsf{NKB}})^{=} Q'$.

Next, assume P contains a peak of the form (ii). If such a pattern originates from a non-overlap then by Lemma 6.6(c) it could be replaced by a smaller proof. Otherwise, by Lemma 6.5, the proof P must contain a proof corresponding to an AC-critical pair $s' \leftarrow\!\!\times\!\!\rightarrow t'$ in $\mathrm{CP}_{\mathrm{AC}}(\mathcal{R}_\omega, \mathcal{S}^e) \cup \mathrm{CP}_{\mathrm{AC}}(\mathcal{S}, \mathcal{R}_\omega)$. Then $s' \leftarrow\!\!\times\!\!\rightarrow t'$ must originate from an AC-critical peak Q of the form $s' \mathrel{r\leftarrow_\ell}\!\leftarrow \cdot \leftrightarrow^*_{\mathrm{AC}} \cdot \rightarrow_{u\to v} t'$ between rules $\ell \to r \in \mathcal{R}_\omega$ and $u \to v \in \mathcal{S}$, and a proof Q' in $\mathcal{T} \cup \mathcal{R}_\omega$ satisfying $Q \Rightarrow^{\succ}_{\mathsf{NKB}} Q'$ exists according to Lemma 6.13. This implies $P = P[Q] \Rightarrow^{\succ}_{\mathsf{NKB}} P[Q']$.

If P contains a subproof Q of the form (iii) we have $c(Q) = \{(\{u,t\},\ldots)\} \cup c_{\mathrm{AC}}(Q)$. Since $u \neq u{\downarrow}$ there is some step $u \rightarrow_{\mathcal{S}/\mathrm{AC}} s$, and thus a peak $P'\colon s \mathrel{\mathcal{S}/\mathrm{AC}}\!\leftarrow u \rightarrow_{\mathcal{R}_\omega/\mathrm{AC}} t$. If P' does not constitute a proper overlap then there exists a rewrite proof Q' of $s \approx t$ which contains only terms smaller than u. For $Q''\colon u \rightarrow_{\mathcal{S}/\mathrm{AC}} s$ the proof $Q''Q'$ is thus smaller than Q as $(\{u,t\},\ldots) \in c(Q)$ dominates all cost tuples in $c(Q''Q')$. If P' constitutes a critical peak then by Lemma 6.13 there exists a proof Q' of $s \approx t$ such that $P' \Rightarrow^{\succ}_{\mathsf{NKB}} Q'$ and for $(T,\ldots) \in c(Q')$ all terms in T are smaller than u. Again $Q \Rightarrow^{\succ}_{\mathsf{NKB}} Q''Q'$ holds.

Finally, if P contains a subproof of the form (iv) then AC convergence of $\mathcal{S}$ yields a smaller proof according to Lemma 6.6(a). □

Correctness Theorem 6.16. *A fair and nonfailing* NKB *run succeeds.*

Proof. Let the run $(\mathcal{E}_0, \varnothing) \vdash (\mathcal{E}_1, \mathcal{R}_1) \vdash (\mathcal{E}_2, \mathcal{R}_2) \vdash \cdots$ under consideration have length $\alpha \leqslant \omega$. As it is nonfailing we have $\mathcal{E}_\omega = \varnothing$. We show that $\leftrightarrow^*_{\mathcal{E}_0\cup\mathcal{T}} \subseteq \rightarrow^*_{\mathcal{R}\setminus\mathcal{S}} \cdot \leftrightarrow^*_{\mathcal{T}} \cdot \mathrel{\mathcal{R}\setminus\mathcal{S}}\!\!{}^*\!\leftarrow$. According to the Persistence Lemma 3.4, any pair of terms in $\leftrightarrow^*_{\mathcal{E}_0\cup\mathcal{T}}$ has a proof in $\mathcal{T} \cup \mathcal{R}_\omega$. Let P be such a proof which is minimal, and assume it is not a normalized rewrite proof. By fairness there exists a proof Q in $(\mathcal{T}, \mathcal{E}_i, \mathcal{R}_i)$ for some $i \geqslant 0$ such that $P \Rightarrow^{\succ}_{\mathsf{NKB}} Q$. According to persistence we also have a proof Q' in $\mathcal{T} \cup \mathcal{R}_\omega$ such that $Q\ (\Rightarrow^{\succ}_{\mathsf{NKB}})^{=} Q'$, and hence $P \Rightarrow^{\succ}_{\mathsf{NKB}} Q'$. This contradicts minimality of P. By the Soundness Lemma 3.2 the equational theories of $\mathcal{T} \cup \mathcal{R}_\omega$ and $\mathcal{T} \cup \mathcal{E}_0$ coincide, so $\mathcal{R}_\omega$ is $\mathcal{S}$-convergent for $\mathcal{E}_0$. □

Example 6.17. Consider an Abelian group with AC operator $\cdot$ and an endo-

morphism f as described by the following set of equations:

$$\mathsf{e} \cdot x \approx x \qquad \mathsf{i}(x) \cdot x \approx \mathsf{e} \qquad \mathsf{f}(x \cdot y) \approx \mathsf{f}(x) \cdot \mathsf{f}(y)$$

together with LPO with precedence $\mathsf{f} \succ \mathsf{i} \succ \cdot \succ \mathsf{e}$. We can obviously apply normalized completion with respect to AC such that $\mathcal{S} = \varnothing$. This results in the AC-convergent TRS $\mathcal{R}_{\mathrm{AC}}$:

$$\begin{array}{lll} \mathsf{e} \cdot x \to x & \mathsf{i}(x) \cdot x \to \mathsf{e} & \mathsf{i}(\mathsf{e}) \to \mathsf{e} \\ \mathsf{i}(\mathsf{i}(x)) \to x & \mathsf{i}(x \cdot y) \to \mathsf{i}(x) \cdot \mathsf{i}(y) & \mathsf{f}(x \cdot y) \to \mathsf{f}(x) \cdot \mathsf{f}(y) \\ \mathsf{f}(\mathsf{e}) \to \mathsf{e} & \mathsf{f}(\mathsf{i}(x)) \to \mathsf{i}(\mathsf{f}(x)) & \end{array}$$

Normalized completion can also be applied with respect to ACU such that $\mathcal{S}_{\mathrm{ACU}} = \{\mathsf{e} \cdot x \to x\}$, as $\mathsf{e} \cdot x \succ x$. We obtain the $\mathcal{S}_{\mathrm{ACU}}$-convergent TRS $\mathcal{R}_{\mathrm{ACU}}$:

$$\begin{array}{lll} \mathsf{i}(x) \cdot x \to \mathsf{e} & \mathsf{i}(\mathsf{e}) \to \mathsf{e} & \mathsf{i}(\mathsf{i}(x)) \to x \\ \mathsf{i}(x \cdot y) \to \mathsf{i}(x) \cdot \mathsf{i}(y) & \mathsf{f}(x \cdot y) \to \mathsf{f}(x) \cdot \mathsf{f}(y) & \mathsf{f}(\mathsf{e}) \to \mathsf{e} \\ \mathsf{f}(\mathsf{i}(x)) \to \mathsf{i}(\mathsf{f}(x)) & & \end{array}$$

Alternatively, we can consider $\mathcal{S}_{\mathrm{G}} = \{\mathsf{e} \cdot x \to x,\ \mathsf{i}(x) \cdot x \to \mathsf{e},\ \mathsf{i}(\mathsf{e}) \to \mathsf{e},\ \mathsf{i}(\mathsf{i}(x)) \to x,\ \mathsf{i}(x \cdot y) \to \mathsf{i}(x) \cdot \mathsf{i}(y)\}$ which is known to be an AC-convergent representation of Abelian groups [6]. Note that $\mathcal{S}_{\mathrm{G}} \subseteq \succ$. An NKB run with respect to $\mathcal{S}_{\mathrm{G}}$ results in the TRS $\mathcal{R}_{\mathrm{G}}$:

$$\mathsf{f}(x \cdot y) \to \mathsf{f}(x) \cdot \mathsf{f}(y) \qquad \mathsf{f}(\mathsf{e}) \to \mathsf{e} \qquad \mathsf{f}(\mathsf{i}(x)) \to \mathsf{i}(\mathsf{f}(x))$$

Remark 6.18. In [75, Definition 3.5] and [76, Definition 3.1], normalizing pairs are defined as follows. Given terms u and v such that $u = u{\downarrow}$, $v = v{\downarrow}$, and $u \succ v$, the functions (Θ, Ψ) form an $\mathcal{S}$-normalizing pair if and only if

(i) for any single-step proof $s \leftrightarrow_{u \approx v} t$ there is a proof P in $(\mathcal{T}, \Theta(u,v), \Psi(u,v))$ such that $s \leftrightarrow_{u \approx v} t \Rightarrow^{\succ}_{\mathsf{NKB}} P$, and

(ii) for all $\ell \to r \in \Psi(u,v)$, all sets of rules $\mathcal{R}$ and all r' such that $r \to^*_{\mathcal{R} \setminus \mathcal{S}} r'$ and any single-step irreducible[7] proof $s \to_{\ell \to r'} t$ there is a proof P in $(\mathcal{T}, \Theta(u,v), \Psi(u,v) \cup \mathcal{R})$ such that $s \to_{\ell \to r'} t \Rightarrow^{\succ}_{\mathsf{NKB}} P$.

Four issues arise with this definition.

(a) It does not require $\Theta(u,v)$ and $\Psi(u,v)$ to be part of the equational theory.

(b) It does not guarantee termination of $\Psi(u,v)$ together with previously oriented rules.

(c) Joinability of AC-critical pairs between $\mathcal{S}$ and $\Psi(u,v)$ is not ensured: Consider the simple example where the theory $\mathcal{E}_0 = \{x + \mathsf{a} \approx \mathsf{a}\}$ is to be completed with respect to $\mathcal{S} = \{y + \mathsf{b} \to \mathsf{b}\}$. We can choose $\Theta(x + \mathsf{a}, \mathsf{a}) = \varnothing$ and $\Psi(x + \mathsf{a}, \mathsf{a}) = \{x + \mathsf{a} \to \mathsf{a}\}$, satisfying (i) and (ii). We obtain the run

$$(\{x + \mathsf{a} \approx \mathsf{a}\}, \varnothing) \vdash (\varnothing, \{x + \mathsf{a} \to \mathsf{a}\})$$

which is obviously fair. But $\{x + \mathsf{a} \to \mathsf{a}\}$ is not $\mathcal{S}$-convergent as the AC-critical pair $\mathsf{a} \leftarrow \rtimes \to \mathsf{b}$ between $\mathcal{S}$ and $x + \mathsf{a} \to \mathsf{a}$ is not considered.

7 A proof is irreducible if it is minimal with respect to $\Rightarrow^{\succ}_{\mathsf{NKB}}$.

(d) The general normalizing pair [75, Definition 3.9] does not match this definition: Assume we orient $x + \mathsf{a} \approx \mathsf{a}$ as $x + \mathsf{a} \to \mathsf{a}$. The general normalizing pair sets $\Theta(x + \mathsf{a}, \mathsf{a}) = \mathrm{CP}_{\mathrm{AC}}(\mathcal{S}, x + \mathsf{a} \to \mathsf{a}) \cup \mathrm{CP}_{\mathrm{AC}}(x + \mathsf{a} \to \mathsf{a}, \mathcal{S}^e)$ and $\Psi(x + \mathsf{a}, \mathsf{a}) = \{x + \mathsf{a} \to \mathsf{a}\}$. Then property (ii) is not satisfied: for $\mathcal{R} = \varnothing$ and $r = r' = \mathsf{a}$ there exists no smaller proof than $x + \mathsf{a} \to^{\epsilon}_{x+\mathsf{a}\to\mathsf{a}} \mathsf{a}$ (and there is also no reason why such a proof should be necessary).

The different definition in [73, Definition 4.4] is as follows. Let u and v be terms such that $u = u{\downarrow}$, $v = v{\downarrow}$, and $u \succ v$. Then $(\Theta(u, v), \Psi(u, v))$ constitutes an $\mathcal{S}$-normalizing pair if and only if the following conditions are satisfied:

(i) Consider a single-step proof P of the form $s \leftrightarrow^{p,\sigma}_{u\approx v} t$ or $s \leftrightarrow^{p,\sigma}_{u\to v} t$ such that σ is in $\mathcal{S}$-normal form, and no $\mathcal{S}$-step applies at a position q in s or t such that $p \parallel q$, or q is above p but the two steps only form a variable overlap. Then there is a proof Q in $(\mathcal{T}, \Theta(u, v), \Psi(u, v))$ such that $P \Rightarrow^{\succ}_{\mathsf{NKB}} Q$.

(ii) For all $\ell \to r \in \Psi(u, v)$ we have $\Theta(\ell, r) \subseteq \Theta(u, v)$ and $\Psi(\ell, r) \subseteq \Psi(u, v)$.

With this definition similar problems arise: Again,

(a) $\Theta(u, v)$ and $\Psi(u, v)$ need not be part of the equational theory,

(b) termination of $\Theta(u, v)$ is not guaranteed, and

(c) AC-critical pairs between $\mathcal{S}$ and $\Psi(u, v)$ need not be considered.

(d) The general normalizing pair does not match this definition either: If $x+\mathsf{a} \approx \mathsf{a}$ is oriented as $x + \mathsf{a} \to \mathsf{a}$ then the resulting $\Theta(x + \mathsf{a}, \mathsf{a})$ and $\Psi(x + \mathsf{a}, \mathsf{a})$ do not admit a proof smaller than $x + \mathsf{a} \to^{\epsilon}_{x+\mathsf{a}\to\mathsf{a}} \mathsf{a}$, so (i) is not fulfilled.

Due to these problems the notion of normalizing pairs was modified according to Definition 6.7.

Completeness of normalized completion can be proved in a similar fashion as completeness of standard completion. First we state an easy consequence of the definition of $\mathcal{S}$-convergence.

Lemma 6.19. *Let $\mathcal{R}$ be $\mathcal{S}$-convergent for $\mathcal{E}$. If $s \leftrightarrow^*_{\mathcal{E}\cup\mathcal{T}} t$ then there is a proof*

$$s \xrightarrow[\mathcal{R}\backslash\mathcal{S}]{!} \cdot \xrightarrow[\mathcal{S}/\mathrm{AC}]{!} u \xleftrightarrow[\mathrm{AC}]{*} v \xleftarrow[\mathcal{S}/\mathrm{AC}]{!} \cdot \xleftarrow[\mathcal{R}\backslash\mathcal{S}]{!} t \tag{6.2}$$

and the terms u and v are unique up to AC equivalence.

Proof. Let s' and t' be normal forms of s and t with respect to $\to_{\mathcal{R}\backslash\mathcal{S}}$. By $\mathcal{S}$-convergence of $\mathcal{R}$, there exists a proof $s' \to^*_{\mathcal{R}\backslash\mathcal{S}} \cdot \leftrightarrow^*_{\mathcal{T}} \cdot \,{}^*_{\mathcal{R}\backslash\mathcal{S}}\!\leftarrow t'$. As s' and t' are $\to_{\mathcal{R}\backslash\mathcal{S}}$-irreducible, we must have $s' \leftrightarrow^*_{\mathcal{T}} t'$. Because $\mathcal{S}$ is AC convergent, $s' \to^!_{\mathcal{S}/\mathrm{AC}} u \leftrightarrow^*_{\mathrm{AC}} v \leftarrow^!_{\mathcal{S}/\mathrm{AC}} t'$ must hold. Note that u and v are irreducible in $\to_{\mathcal{S}/\mathrm{AC}}$, and by the definition of normalized rewriting, also in $\to_{\mathcal{R}\backslash\mathcal{S}}$. Now suppose that u' also satisfies (6.2). Then we have $u \leftrightarrow^*_{\mathcal{E}\cup\mathcal{T}} u'$, so there is a proof of $u \approx u'$ of the form (6.2). Since also u' must be irreducible with respect to $\to_{\mathcal{R}\backslash\mathcal{S}}$ and $\to_{\mathcal{S}/\mathrm{AC}}$, we have $u \leftrightarrow^*_{\mathrm{AC}} u'$. A symmetric argument shows that v is unique modulo AC. □

Completeness Theorem 6.20. *Assume $\mathcal{R}$ is a finite $\mathcal{S}$-convergent system for $\mathcal{E}$ and let $\succ$ be an AC-compatible reduction order that contains $\mathcal{R}$ and $\mathcal{S}$. Then any fair and nonfailing run from $\mathcal{E}$ using $\succ$ will produce an $\mathcal{S}$-convergent system in finitely many steps.*

Proof. Let $\mathcal{R}'$ denote the system obtained from $\mathcal{R}$ after replacing each right-hand side r by r' such that $r \to^!_{\mathcal{R}\backslash\mathcal{S}} \cdot \to^!_{\mathcal{S}/\mathrm{AC}} r'$. An argument similar to the one used in the proof of Lemma 6.19 shows that r' is unique modulo AC. The system $\mathcal{R}'$ is terminating because it is contained in $\succ$. Moreover, the relation $\downarrow\!\downarrow_{\mathcal{R}\backslash\mathcal{S}}$ is contained in $\downarrow\!\downarrow_{\mathcal{R}'\backslash\mathcal{S}}$ because $\to_{\mathcal{R}\backslash\mathcal{S}}$ and $\to_{\mathcal{R}'\backslash\mathcal{S}}$ have the same normal forms modulo AC. Let $s \leftrightarrow^*_{\mathcal{E}\cup\mathcal{T}} t$. As $\mathcal{R}$ is $\mathcal{S}$-convergent, there exists an equational proof of $s \approx t$ of the form (6.2) by Lemma 6.19. Thus there is also a proof

$$s \xrightarrow[\mathcal{R}'\backslash\mathcal{S}]{!} \cdot \xrightarrow[\mathcal{S}/\mathrm{AC}]{!} \cdot \xleftrightarrow[\mathrm{AC}]{*} \cdot \xleftarrow[\mathcal{S}/\mathrm{AC}]{!} \cdot \xleftarrow[\mathcal{R}'\backslash\mathcal{S}]{!} t \tag{6.3}$$

Hence $\mathcal{R}'$ is $\mathcal{S}$-convergent for $\mathcal{E}$. Now consider a nonfailing NKB run γ which starts from $(\mathcal{E}, \varnothing)$ and uses $\succ$. Let $\mathcal{R}_\omega$ be the set of persistent rules. Let $\ell \to r$ be a rule in $\mathcal{R}'$. Since $\ell \approx r$ belongs to the equational theory of $\mathcal{T} \cup \mathcal{E}$, it has a persistent equational proof P after a finite number of steps in γ. Note that r must be $\to_{\mathcal{R}_\omega\backslash\mathcal{S}}$-irreducible: If $r \to_{\mathcal{R}_\omega\backslash\mathcal{S}} r'$ for some term r' then $r \succ r'$ and $r \leftrightarrow^*_{\mathcal{E}\cup\mathcal{T}} r'$, so $\mathcal{R}'$ admits an $\mathcal{S}$-normalized rewrite proof for $r \approx r'$. As r is irreducible in $\to_{\mathcal{R}\backslash\mathcal{S}}$ and $\to_{\mathcal{S}/\mathrm{AC}}$, this proof must have the form $r \leftrightarrow^*_{\mathrm{AC}} \cdot \leftarrow^!_{\mathcal{S}/\mathrm{AC}} \cdot \leftarrow^!_{\mathcal{R}'\backslash\mathcal{S}} r'$. As $\mathcal{R}' \cup \mathcal{S} \subseteq \succ$ and $\succ$ is AC compatible, this contradicts $r \succ r'$. It follows that the proof P must have the form $\ell \to^+_{\mathcal{R}_\omega\backslash\mathcal{S}} \cdot \to^*_{\mathcal{S}/\mathrm{AC}} \cdot \leftrightarrow^*_{\mathrm{AC}} r$. Let $\mathcal{Q}$ denote the set of rules in $\mathcal{R}_\omega$ required for all these rewrite proofs for rules of $\mathcal{R}'$. The system $\mathcal{Q}$ is obviously terminating. As $\mathcal{R}$ is finite, so are $\mathcal{R}'$ and $\mathcal{Q}$, and hence γ derives all rules in $\mathcal{Q}$ after a finite number of steps. We claim that $\mathcal{Q}$ is $\mathcal{S}$-convergent for $\mathcal{E}$. First note that by the definition of normalized rewriting the relations $\to^*_{\mathcal{S}/\mathrm{AC}} \cdot \leftrightarrow^*_{\mathrm{AC}} \cdot \to_{\mathcal{Q}\backslash\mathcal{S}}$ and $\to_{\mathcal{Q}\backslash\mathcal{S}}$ coincide. Thus the inclusion $\to_{\mathcal{R}'\backslash\mathcal{S}} \subseteq \to^*_{\mathcal{Q}\backslash\mathcal{S}} \cdot \to^*_{\mathcal{S}/\mathrm{AC}} \cdot \leftrightarrow^*_{\mathrm{AC}}$ entails

$$\xrightarrow[\mathcal{R}'\backslash\mathcal{S}]{*} \subseteq \xrightarrow[\mathcal{Q}\backslash\mathcal{S}]{*} \cdot \xrightarrow[\mathcal{S}/\mathrm{AC}]{*} \cdot \xleftrightarrow[\mathrm{AC}]{*} \tag{6.4}$$

Hence for every proof $u \leftrightarrow^*_{\mathcal{E}\cup\mathcal{T}} v$ there exists an equational proof of the form (6.3), and due to (6.4) and $\to^*_{\mathcal{S}/\mathrm{AC}} \cdot \leftrightarrow^*_{\mathrm{AC}} \cdot \to^*_{\mathcal{S}/\mathrm{AC}} \cdot \leftrightarrow^*_{\mathrm{AC}} \subseteq \to^*_{\mathcal{S}/\mathrm{AC}} \cdot \leftrightarrow^*_{\mathrm{AC}}$, also a valley proof

$$u \xrightarrow[\mathcal{Q}\backslash\mathcal{S}]{!} \cdot \xrightarrow[\mathcal{S}/\mathrm{AC}]{!} \cdot \xleftrightarrow[\mathrm{AC}]{*} \cdot \xleftarrow[\mathcal{S}/\mathrm{AC}]{!} \cdot \xleftarrow[\mathcal{Q}\backslash\mathcal{S}]{!} v \tag{6.5}$$

using rules in $\mathcal{Q}$ is possible. □

A TRS $\mathcal{R}$ is called *$\mathcal{S}$-reduced* if for all rules $\ell \to r$ in $\mathcal{R}$ the term r is in normal form with respect to $\to_{\mathcal{S}/\mathrm{AC}}$ and $\to_{\mathcal{R}\backslash\mathcal{S}}$, and ℓ is in normal form with respect to $\to_{\mathcal{S}/\mathrm{AC}}$ and $\to_{\ell'\to r'\backslash\mathcal{S}}$ for every rule $\ell' \to r'$ in $\mathcal{R}$ different from $\ell \to r$. A TRS $\mathcal{R}$ is called *$\mathcal{S}$-canonical for $\mathcal{E}$* if it is both $\mathcal{S}$-reduced and $\mathcal{S}$-convergent for $\mathcal{E}$. Note that TRSs derived in a fair and nonfailing NKB run are $\mathcal{S}$-canonical

if simplify, compose and collapse are applied exhaustively, since an orient step may only be applied to terms in $\mathcal{S}$-normal form. We conclude this section with a uniqueness result about $\mathcal{S}$-canonical systems, and give a full proof, extending the proof sketch in [73].

Theorem 6.21. *Consider two TRSs $\mathcal{R}$ and $\mathcal{R}'$ which are $\mathcal{S}$-canonical for $\mathcal{E}$ and contained in the same AC-compatible reduction order $\succ$. Then $\mathcal{R}$ and $\mathcal{R}'$ are equal up to variable renaming and AC equivalence.*

Proof. Firstly, we show that the set of normal forms of $\to_{\mathcal{R}\setminus\mathcal{S}}$ and $\to_{\mathcal{R}'\setminus\mathcal{S}}$ coincide. Assume to the contrary that $\ell \to r$ is a rule in $\mathcal{R}$ such that its left-hand side ℓ is not reducible in $\to_{\mathcal{R}'\setminus\mathcal{S}}$. As both systems are $\mathcal{S}$-convergent for the same theory there exists a proof of the form

$$\ell \xrightarrow[\mathcal{R}'\setminus\mathcal{S}]{*} \cdot \xleftrightarrow[\mathcal{T}]{*} \cdot \xleftarrow[\mathcal{R}'\setminus\mathcal{S}]{*} r \tag{6.6}$$

Since ℓ is irreducible in $\to_{\mathcal{R}'\setminus\mathcal{S}}$ and $\to_{\mathcal{S}/\mathrm{AC}}$, this proof needs to have the form

$$r \xrightarrow[\mathcal{R}'\setminus\mathcal{S}]{*} \cdot \xrightarrow[\mathcal{S}/\mathrm{AC}]{*} \cdot \xleftrightarrow[\mathrm{AC}]{*} \ell$$

As $\mathcal{R}' \cup \mathcal{S} \subseteq \succ$ and $\succ$ is AC compatible, this contradicts $\ell \succ r$. Hence ℓ must be reducible in $\to_{\mathcal{R}'\setminus\mathcal{S}}$, and as the reasoning is symmetric, the normal forms of $\to_{\mathcal{R}\setminus\mathcal{S}}$ and $\to_{\mathcal{R}'\setminus\mathcal{S}}$ coincide.

Hence for every rule $\ell \to r$ in $\mathcal{R}$, the left-hand side ℓ is reducible by a rule $\ell' \to r'$ in $\mathcal{R}'$. We must have $\ell \leftrightarrow^*_{\mathrm{AC}} \ell'\theta$ for some renaming θ; if ℓ' reduces ℓ below the root or if θ is not a renaming, then ℓ' must also be reducible in $\to_{\mathcal{R}\setminus\mathcal{S}}$, contradicting the assumption that $\mathcal{R}$ is $\mathcal{S}$-reduced. For the same reason, $\ell' \to r'$ is the only rule in $\mathcal{R}'$ that allows to reduce ℓ in an $\to_{\mathcal{R}'\setminus\mathcal{S}}$-step. Moreover, since $\mathcal{R}'$ is $\mathcal{S}$-canonical, the term r' has to be in normal form with respect to $\to_{\mathcal{S}/\mathrm{AC}}$ and $\to_{\mathcal{R}'\setminus\mathcal{S}}$, so (6.6) must have the form

$$\ell \xrightarrow[\ell'\to r'\setminus\mathcal{S}]{\epsilon} r'\theta \xleftrightarrow[\mathrm{AC}]{*} \cdot \xleftarrow[\mathcal{S}/\mathrm{AC}]{*} \cdot \overset{*}{\underset{\mathcal{R}'\setminus\mathcal{S}}{\leftarrow}} r$$

As r is in normal form with respect to $\to_{\mathcal{R}\setminus\mathcal{S}}$ and $\to_{\mathcal{S}/\mathrm{AC}}$, and normal forms in $\to_{\mathcal{R}\setminus\mathcal{S}}$ and $\to_{\mathcal{R}'\setminus\mathcal{S}}$ coincide, we must have $r \leftrightarrow^*_{\mathrm{AC}} r'\theta$. Thus for every rewrite rule $\ell \to r$ in $\mathcal{R}$ there is a rule $\ell' \to r'$ in $\mathcal{R}'$ such that $\ell \leftrightarrow^*_{\mathrm{AC}} \ell'\theta$ and $r \leftrightarrow^*_{\mathrm{AC}} r'\theta$ for some renaming θ. By symmetry of the argument, $\mathcal{R}$ and $\mathcal{R}'$ are the same up to variable renaming and AC equivalence. □

6.1.1 Special Normalizing Pairs

Although general normalizing pairs as given in Definition 6.7 can always be applied, some theories allow for a more efficient choice. We briefly recall some specialized normalizing pairs presented in [75, 76].

$\mathcal{T}$	$\mathcal{S}$	$\Theta_{\mathcal{T}}(u,v)$
ACU$(+,\mathsf{0})$	$\{x+\mathsf{0}\to x\}$	$\{(u\approx v)\{x\mapsto\mathsf{0}\} \mid u\trianglerighteq x+w\}$
ACI$(+)$	$\{x+x\to x\}$	$\mathrm{CP}_{\mathrm{AC}}(u\to v,\{x+x\to x\}^e)$
ACUI$(+,\mathsf{0})$	$\{x+x\to x,\ x+\mathsf{0}\to x\}$	$\Theta_{\mathrm{ACU}}(u,v)\cup\Theta_{\mathrm{ACI}}(u,v)$
AC0$(\cdot,\mathsf{0})$	$\{x\cdot\mathsf{0}\to\mathsf{0}\}$	$\{(u\approx v)\{x\mapsto\mathsf{0}\} \mid u\trianglerighteq x\cdot w\}$
ACN$(+,\mathsf{0})$	$\{x+x\to\mathsf{0}\}$	$\mathrm{CP}_{\mathrm{AC}}(u\to v,\{x+x\to\mathsf{0}\}^e)$

Figure 6.2: Normalizing equations $\Theta_{\mathcal{T}}(u,v)$ for some theories $\mathcal{T}$.

Some Simple Theories

We first specialize general normalizing pairs (cf. Definition 6.7) to some common theories.

Definition 6.22. Let u and v be terms such that $u\succ v$. For $\mathcal{T}$ being one of the theories ACU, ACI, ACUI, AC0, and ACN, let $\Theta_{\mathcal{T}}$ be as given in Figure 6.2 and $\Psi_{\mathcal{T}}(u,v)=\{u\to v\}$.

Marché shows that these function pairs are actually specializations of the general normalizing pair for the respective theories, hence they also form normalizing pairs according to Definition 6.7.

Lemma 6.23 ([75, 76])**.** *For terms u and v such that $u\succ v$ the functions $(\Theta_{\mathcal{T}},\Psi_{\mathcal{T}})$ given in Definition 6.22 constitute normalizing pairs for u and v.*

Abelian Groups

If the theory $\mathcal{T}$ extends Abelian group theory, Marché proposes to use *symmetrization* to obtain more efficient normalizing pairs. This approach is briefly outlined in the remainder of this subsection, following the presentation in [75]. Let us thus assume that $\mathcal{T}$ contain the following equations:

$$x+(y+z)\approx(x+y)+z\qquad x+y\approx y+x\qquad \mathsf{0}+x\approx x\qquad (-x)+x\approx\mathsf{0}$$

This subtheory is represented by the AC convergent TRS $\mathcal{S}_{\mathrm{G}}$:

$$\mathsf{0}+x\to x\quad (-x)+x\to\mathsf{0}\quad -\mathsf{0}\to\mathsf{0}\quad -(-x)\to x\quad -(x+y)\to(-x)+(-y)$$

We abbreviate an expression $t+\cdots+t$ with n occurrences of t by nt. The following definition is inspired by [68]:

Definition 6.24. Let u, v be terms and w be the $\mathcal{S}_{\mathrm{G}}$-normal form of $u+(-v)$. If the term w can be written as $w=n_1w_1+n_2w_2+\cdots+n_kw_k$ such that $w_1\succ w_j$ for all $2\leqslant j\leqslant k$ then the *symmetrization* of u and v is defined as $symm(u,v)=(n_1,w_1,-n_2w_2-\cdots-n_kw_k)$. If no such presentation of w exists then the symmetrization of u and v is undefined.

Example 6.25. Let $u = 2\mathsf{a} + (-\mathsf{b}) + 2\mathsf{c}$ and $v = \mathsf{a} + 2\mathsf{b} + \mathsf{c}$. Then $w = u + (-v)$ normalizes to $\mathsf{a} - 3\mathsf{b} + \mathsf{c}$. Suppose we choose AC-RPO as reduction order. The symmetrization of u and v then depends on the precedence.

- Let $\succ_1$ satisfy $\mathsf{a} \succ_1 \mathsf{b} \succ_1 \mathsf{c} \succ_1 - \succ_1 + \succ_1 \mathsf{0}$. Note that $\succ_1$ is compatible with $\mathcal{S}_\mathrm{G}$. Then $\mathit{symm}(u, v) = (1, \mathsf{a}, (-3\mathsf{b}) + \mathsf{c})$.
- Let $\succ_2$ satisfy $\mathsf{b} \succ_2 \mathsf{a} \succ_2 \mathsf{c} \succ_2 - \succ_2 + \succ_2 \mathsf{0}$, which is also compatible with $\mathcal{S}_\mathrm{G}$. Then $\mathit{symm}(u, v) = (3, \mathsf{b}, (-\mathsf{a}) + (-\mathsf{c}))$.

Definition 6.26. Suppose u, v are terms such that $u \succ v$ and $\mathit{symm}(u, v) = (n, s, t)$. Then $(\Theta_\mathrm{G}, \Psi_\mathrm{G})$ is defined as follows. If $n = 1$ then $\Psi_\mathrm{G}(u, v) = \{s \to t\}$, otherwise $\Psi_\mathrm{G}(u, v)$ consists of the rules

$$ns \to t \qquad\qquad -s \to ((n-1)s + (-t)) \downarrow_{\mathcal{S}_\mathrm{G}}$$

whereas $\Theta_\mathrm{G}(u, v)$ is obtained by

$$\begin{aligned}
\Theta_\mathrm{G}(u, v) &= \Theta_\mathrm{ACU}(ns, t) \cup \Sigma_1(ns, t) \cup \Sigma_2(ns, t) \\
\Sigma_1(u, v) &= \mathrm{CP}_\mathrm{AC}(u \to v, \{x + (-x) \to \mathsf{0}\}^e) \\
\Sigma_2(u, v) &= \{u\sigma \approx v\sigma \mid \sigma = \{x \mapsto \mathsf{0}\} \text{ if } u \trianglerighteq -x \text{ such that } x \in \mathcal{V}\} \\
&\quad \cup \{u\sigma \approx v\sigma \mid \sigma = \{x \mapsto -y\} \text{ if } u \trianglerighteq -x \text{ such that } x \in \mathcal{V}\} \\
&\quad \cup \{u\sigma \approx v\sigma \mid \sigma = \{x \mapsto y + z\} \text{ if } u \trianglerighteq -x \text{ such that } x \in \mathcal{V}\}
\end{aligned}$$

Example 6.27. Consider the terms from Example 6.25. For $\succ_1$ we have $u \succ_1 v$, $\Psi_\mathrm{G}(u, v) = \{\mathsf{a} \to 3\mathsf{b} + (-\mathsf{c})\}$ and $\Theta_\mathrm{G}(u, v) = \varnothing$. For $\succ_2$ we have $u \succ_2 v$, $\Psi_\mathrm{G}(u, v) = \{3\mathsf{b} \to (-\mathsf{a}) + (-\mathsf{c}),\ -\mathsf{b} \to 2\mathsf{b} + \mathsf{a} + \mathsf{c}\}$ and $\Theta_\mathrm{G}(u, v) = \varnothing$.

Definition 6.28. An AC-compatible reduction order satisfies the *symmetrization property* if for all terms u, v, and w such that $u \succ v$, $u \succ w$, and $\mathrm{root}(u)$, $\mathrm{root}(v)$, and $\mathrm{root}(w)$ are not in $\{+, -, \mathsf{0}\}$, we have $u \succ (-v) + w$.

Lemma 6.29 ([75, 76])**.** *If the employed AC-compatible reduction order $\succ$ satisfies the symmetrization property then $(\Theta_\mathrm{G}, \Psi_\mathrm{G})$ is a normalizing pair.*

Commutative Rings

Symmetrization can also save many critical pairs in the case of commutative ring theory. Let $\mathcal{T}_\mathrm{CR}$ extend $\mathcal{T}_\mathrm{G}$ with the equations

$$x \cdot (y \cdot z) \approx (x \cdot y) \cdot z \quad x \cdot y \approx y \cdot x \quad \mathsf{1} \cdot x \approx x \quad x \cdot (y + z) \approx x \cdot y + x \cdot z$$

Then $\mathcal{S}_\mathrm{CR}$ can be chosen as the TRS adding the following rules to $\mathcal{S}_\mathrm{G}$:

$$\mathsf{1} \cdot x \to x \quad x \cdot (y + z) \to x \cdot y + x \cdot z \quad \mathsf{0} \cdot x \to x \quad (-x) \cdot y \to -(x \cdot y)$$

Definition 6.30. Suppose u, v are terms such that $u \succ v$ and $\mathit{symm}(u, v) = (n, s, t)$. Then $(\Theta_\mathrm{CR}, \Psi_\mathrm{CR})$ is defined as follows. If $n = 1$ then $\Psi_\mathrm{CR}(u, v) = \{s \to t\}$, otherwise $\Psi_\mathrm{CR}(u, v)$ consists of the rules

$$\begin{aligned}
ns &\to t & -s &\to ((n-1)s + (-t)) \downarrow_{\mathcal{S}_\mathrm{CR}} \\
n(x \cdot s) &\to (x \cdot t) \downarrow_{\mathcal{S}_\mathrm{CR}} & -(x \cdot s) &\to ((n-1)x \cdot s + -(x \cdot t)) \downarrow_{\mathcal{S}_\mathrm{CR}}
\end{aligned}$$

orient	$\dfrac{\mathcal{E} \uplus \{s \simeq t\}, \mathcal{R}}{\mathcal{E}, \mathcal{R} \cup \{s \to t\}}$	if $s \succ t$
deduce	$\dfrac{\mathcal{E}, \mathcal{R}}{\mathcal{E} \cup \{s \approx t\}, \mathcal{R}}$	if $s \leftarrow \rtimes \rightarrow t \in \mathrm{CP}_{\mathrm{AC}}(\mathcal{R}, \mathcal{R}^e)$
delete	$\dfrac{\mathcal{E} \uplus \{s \approx t\}, \mathcal{R}}{\mathcal{E}, \mathcal{R}}$	if $s \leftrightarrow^*_{\mathrm{AC}} t$
simplify	$\dfrac{\mathcal{E} \uplus \{s \simeq t\}, \mathcal{R}}{\mathcal{E} \cup \{s \simeq u\}, \mathcal{R}}$	if $t \to_{\mathcal{R}/\mathrm{AC}} u$
compose	$\dfrac{\mathcal{E}, \mathcal{R} \uplus \{s \to t\}}{\mathcal{E}, \mathcal{R} \cup \{s \to u\}}$	if $t \to_{\mathcal{R}/\mathrm{AC}} u$
collapse	$\dfrac{\mathcal{E}, \mathcal{R} \uplus \{t \to s\}}{\mathcal{E} \cup \{u \approx s\}, \mathcal{R}}$	if $t \xrightarrow[\ell \to r/\mathrm{AC}]{p,\sigma} u$ for $\ell \to r \in \mathcal{R}$ with $t \rhd_{\mathrm{AC}} \ell$ or $s \succ r\sigma$

Figure 6.3: The inference system AC of AC completion.

The set $\Theta_{\mathrm{CR}}(u, v)$ comprises the equations

$$\Theta_{\mathrm{CR}}(u,v) = \Theta_{\mathrm{ACU}(+,0)}(ns,t) \cup \Theta_{\mathrm{ACU}(\cdot,1)}(ns,t) \cup \Theta_{\mathrm{AC0}(\cdot,0)}(ns,t) \cup \Sigma_1(ns,t) \cup \Sigma_2(ns,t) \cup \Sigma_3(ns,t)$$

where $\Sigma_1(ns, t)$ and $\Sigma_2(ns, t)$ is defined as for Abelian groups while

$$\Sigma_3(u,v) = \{u\sigma \approx v\sigma \mid \sigma = \{x \mapsto x + y\} \text{ if } u \trianglerighteq x \cdot z \text{ such that } x, z \in \mathcal{V}\} \cup \{u\sigma \approx v\sigma \mid \sigma = \{x \mapsto -x\} \text{ if } u \trianglerighteq x \cdot z \text{ such that } x, z \in \mathcal{V}\}$$

Lemma 6.31 ([75, 76])**.** *If the employed AC-compatible reduction order $\succ$ satisfies the symmetrization property then $(\Theta_{\mathrm{CR}}, \Psi_{\mathrm{CR}})$ is a normalizing pair.*

6.1.2 AC Completion

For the case where $\mathcal{T}$ consists only of AC equations we have $\mathcal{S} = \varnothing$, and normalized completion can be simplified to the inference system AC in Figure 6.3. Note that in this case the general normalizing pair yields $\Psi(u, v) = \{u \to v\}$ and $\Theta(u, v) = \varnothing$ for all terms $u \succ v$.

An AC run is *fair* if it is fair with respect to $\Rightarrow^{\succ}_{\mathsf{NKB}}$ according to Definition 6.12. The following statements are obtained when specializing the corresponding normalized completion results.

Lemma 6.32. *A nonfailing* AC *run satisfying* $\mathrm{CP}_{\mathrm{AC}}(\mathcal{R}_\omega, \mathcal{R}^e_\omega) \subseteq \bigcup_i \mathcal{E}_i$ *is fair.* □

Correctness Theorem 6.33. *A fair and nonfailing* AC *run succeeds.* □

collapse	$\dfrac{\mathcal{E}, \mathcal{R} \uplus \{t \to s\}}{\mathcal{E} \cup \{u \approx s\}, \mathcal{R}}$	if $t \to_{\mathcal{R} \setminus \mathcal{S}} u$

Figure 6.4: The collapse rule in NKB′.

Note that as $\mathcal{S} = \varnothing$, normalized AC convergence coincides with AC convergence. Hence the adaptation of Completeness Theorem 6.20 can be stated as follows:

Completeness Theorem 6.34. *Assume $\mathcal{R}$ is a finite AC-convergent system for $\mathcal{E}$ and let $\succ$ be an AC-compatible reduction order that contains $\mathcal{R}$. Then any fair and nonfailing run from $\mathcal{E}$ using $\succ$ will produce an AC-convergent system in finitely many steps.* □

We remark that the inference system AC is somewhat simpler than the one obtained when specializing *extended completion* as presented in [6].

6.1.3 Finite Runs

As for standard completion, *finite* normalized completion can be run with a simpler collapse rule. Let the inference system NKB′ consist of the modified collapse rule in Figure 6.4 together with all other inference rules from Figure 6.1. Throughout this section we assume that all equations and rewrite rules in $\bigcup_i \mathcal{E}_i \cup \mathcal{R}_i$ are variable-disjoint. Thus, every equation and rewrite rule created during a run has fresh variables. Moreover, we will consider only *finite* NKB′ runs of a certain length $n \in \mathbb{N}$:

$$(\mathcal{E}_0, \varnothing) \vdash (\mathcal{E}_1, \mathcal{R}_1) \vdash (\mathcal{E}_2, \mathcal{R}_2) \vdash \cdots \vdash (\mathcal{E}_n, \mathcal{R}_n) \tag{6.7}$$

A run *fails* if $\mathcal{E}_n$ is non-empty, it *succeeds* if $\mathcal{E}_n$ is empty and $\mathcal{R}_n$ is $\mathcal{S}$-convergent for $\mathcal{E}_0$. We define a modified proof order $\Rightarrow^{\succ,n}_{\mathsf{NKB}}$ which depends on the actual length n of the run.

Definition 6.35. Consider a run of the form (6.7) which has length $n \in \mathbb{N}$ and uses the reduction order $\succ$. Let $(\mathcal{E}, \mathcal{R}) = (\mathcal{E}_i, \mathcal{R}_i)$ for some $0 \leqslant i \leqslant n$. The *cost* c_n of an equational proof step in $(\mathcal{T}, \mathcal{E}, \mathcal{R})$ is defined as follows:

$$c(s \underset{u \approx v}{\longleftrightarrow} t) = (\bot, \{s\}, \bot, 0) \qquad \text{if } u \simeq v \in \mathrm{AC}$$

$$c(s \overset{p}{\underset{u \approx v}{\longleftrightarrow}} t) = (\{s{\downarrow_p}, t{\downarrow_p}\}, \{s, t\}, \bot, 0) \qquad \text{if } u \simeq v \in \mathcal{E}$$

$$c(s \overset{p}{\underset{\ell \to r}{\longleftrightarrow}} t) = c(t \overset{p}{\underset{r \leftarrow \ell}{\longleftrightarrow}} s) = (c(s, p, t), \{s\}, (s|_p){\downarrow}, n - k) \qquad \text{if } k \text{ is maximal such that } \ell \to r \in \mathcal{R}_k$$

$$c(s \underset{\ell \to r}{\longleftrightarrow} t) = c(t \underset{r \leftarrow \ell}{\longleftrightarrow} s) = (\bot, \{s\}, \bot, 0) \qquad \text{if } \ell \to r \in \mathcal{S}$$

We compare costs with the lexicographic combination of $(\succ_{\mathsf{mul}}, (\leftrightarrow^*_{\mathrm{AC}})_{\mathsf{mul}})$ for the first two components, $(\rhd_{\mathrm{AC}}, \leftrightarrow^*_{\mathrm{AC}})$, and $(>, =)$ for the standard order $>$ on $\mathbb{N}$. The symbol $\bot$ is considered minimal in the former three orderings. The

cost of an equational proof is the multiset consisting of the costs of its steps. The proof order $\succnsim^n_{\mathsf{NKB}}$ is the multiset extension of the order on proof step costs, and $P \Rightarrow^{\succ,n}_{\mathsf{NKB}} Q$ holds if and only if $P \succnsim^n_{\mathsf{NKB}} Q$ and P and Q prove the same equation.

As a lexicographic combination of well-founded orders $\succnsim^n_{\mathsf{NKB}}$ is terminating. It is not difficult to show that the relation also constitutes a proof reduction relation.

Lemma 6.36. *The relation $\Rightarrow^{\succ,n}_{\mathsf{NKB}}$ is a proof reduction relation.* □

We define general normalizing pairs as in Definition 6.7, except that the proof reduction relation $\Rightarrow^{\succ,n}_{\mathsf{NKB}}$ is used. It is then easy to check that the results of Lemmas 6.6 and 6.10 carry over to NKB' as all comparisons of equational proofs involve only the first and the second cost component, and $\Rightarrow^{\succ}_{\mathsf{NKB}}$ and $\Rightarrow^{\succ,n}_{\mathsf{NKB}}$ coincide on these.

Lemma 6.37. *Every NKB' run constitutes an equational inference sequence with respect to $\Rightarrow^{\succ,n}_{\mathsf{NKB}}$ and $\mathcal{T}$.*

Proof. It has to be shown that every inference step $(\mathcal{E}_i, \mathcal{R}_i) \vdash_{\mathsf{NKB}'} (\mathcal{E}_{i+1}, \mathcal{R}_{i+1})$ can be modeled by expand and contract steps according to Definition 3.1. Except for compose and collapse steps this can be argued as in the proof of Lemma 6.11.

A compose step expands by adding $s \to u$ and contracts by removing $s \to t$. The latter corresponds to the proof transformation replacing $P\colon s \leftrightarrow^{\epsilon}_{s\to t} t$ by $Q\colon s \leftrightarrow^{\epsilon}_{s\to u} u \;{}^{p}_{\mathcal{R}\setminus\mathcal{S}}\!\leftarrow t$. We have $c(P) = \{(c(s,\epsilon,t), \{s\}, s{\downarrow}, n-i)\}$ since—by the assumption on variable-disjoint rules—the rule $s \to t$ cannot occur in any $\mathcal{R}_j$ with $j > i$. On the other hand, $c(Q) = \{(c(s,\epsilon,u), \{s\}, s{\downarrow}, n-j), (\{t{\downarrow}\}, \ldots)\} \cup c_{\mathcal{S}\cup\mathrm{AC}}(Q)$ for some $j > i$. From $s \succ t \succeq t{\downarrow} \succ u$ it follows that $c(s,\epsilon,t) \succeq_{\mathsf{mul}} c(s,\epsilon,u)$, and $c(s,\epsilon,t) \succeq_{\mathsf{mul}} \{t{\downarrow}\}$ and thus $c(P) \succnsim^n_{\mathsf{NKB}} c(Q)$.

A collapse step constitutes an expansion adding $u \approx s$ followed by a contraction removing $t \to s$, which is a proof transformation replacing $P\colon t \leftrightarrow^{\epsilon}_{t\to s} s$ by $Q\colon t \to^{p}_{\ell\to r\setminus\mathcal{S}} u \leftrightarrow^{\epsilon}_{u\approx s} s$. We have $c(P) = \{(c(t,\epsilon,s), \{t\}, t{\downarrow}, n-i)\}$ as by the assumption on variable-disjoint rules the rule $t \to s$ cannot occur in any $\mathcal{R}_j$ with $j > i$. On the other hand, $c(Q) = \{(\{t{\downarrow}\}, \{t{\downarrow}\}, t'|_p, n-j),\ (\{s{\downarrow}, u{\downarrow}\}, \ldots)\} \cup c_{\mathcal{S}\cup\mathrm{AC}}(Q)$ for some term t' such that $t' \leftrightarrow^*_{\mathrm{AC}} t{\downarrow}$ and some $j > i$. Because of $t \succ s \succeq s{\downarrow}$, $t \succ u \succeq u{\downarrow}$, $c(t,\epsilon,s) \succeq t{\downarrow}$ and $t{\downarrow} \trianglerighteq_{\mathrm{AC}} t'|_p$ we have $c(P) \succnsim^n_{\mathsf{NKB}} c(Q)$. □

An NKB' run is *fair* if it is fair with respect to $\Rightarrow^{\succ,n}_{\mathsf{NKB}}$ according to Definition 6.12. Note that Lemma 6.13 carries over to NKB' as it only exploits that the underlying calculus is an equational inference system. We can therefore prove the following results on fairness, correctness, and completeness in exactly the same way as for the inference system NKB.

Lemma 6.38. *A nonfailing NKB' run satisfying $\mathrm{CP}_{\mathcal{L}}(\mathcal{R}_\omega, \mathcal{R}^{e}_{\omega}) \subseteq \bigcup_i \mathcal{E}_i$ is fair.* □

Correctness Theorem 6.39. *A fair and nonfailing NKB' run succeeds.* □

Completeness Theorem 6.40. *Assume $\mathcal{R}$ is a finite $\mathcal{S}$-convergent system for $\mathcal{E}$ and let $\succ$ be an AC-compatible reduction order that contains $\mathcal{R}$ and $\mathcal{S}$. Then any fair and nonfailing* NKB' *run from $\mathcal{E}$ using $\succ$ will produce an $\mathcal{S}$-convergent system in finitely many steps.* □

Note that according to the latter result the (slightly) simpler NKB' is equally powerful as NKB when it comes to deriving finite $\mathcal{S}$-convergent TRSs.

6.1.4 Critical Pair Criteria

Also in the setting of normalized completion critical pair criteria pose a means to filter out critical pairs that can be ignored without compromising completeness. Again the *compositeness criterion* serves as a general condition. Let $\mathcal{L}$ be a fixed theory between AC and $\mathcal{T}$. A critical pair criterion CPC maps $(\mathcal{E}, \mathcal{R})$ to a set of equations such that $\mathrm{CPC}(\mathcal{E}, \mathcal{R})$ is a subset of $\mathrm{CP}_{\mathcal{L}}(\mathcal{R}, \mathcal{R}^e)$. We adapt the notion of compositeness such that the proof orders for normalized completion are used.

Definition 6.41. Let $\mathcal{E}$ be a set of equations, $\mathcal{R}$ a set of rewrite rules, and $\succcurlyeq$ a proof order using reduction order $\succ$. An equational proof P that has the form of a peak $s \leftarrow \cdot \leftrightarrow^*_{\mathcal{L}} \cdot \rightarrow t$ is *composite* in $(\mathcal{T}, \mathcal{E}, \mathcal{R})$ with respect to a proof order $\succcurlyeq$ if there exist terms $u_0, \ldots, u_{n+1}$ where $s = u_0$, $t = u_{n+1}$ and $u \succ u_i$ for all $0 \leqslant i \leqslant n+1$, and proofs $P_0, \ldots, P_n$ in $(\mathcal{T}, \mathcal{E}, \mathcal{R})$ such that P_i proves $u_i \approx u_{i+1}$ and $P \succcurlyeq P_i$ for all $1 \leqslant i \leqslant n$. The *compositeness criterion* $\mathrm{CCP}_{\mathcal{L}}(\mathcal{E}, \mathcal{R})$ returns all $\mathcal{L}$-critical pairs among rules in $\mathcal{R}$ for which the associated overlaps are composite.

We can now relax the result of Lemma 6.15 by showing that composite critical pairs can be ignored without affecting fairness.

Lemma 6.42. *Consider a nonfailing* NKB *run $\gamma\colon (\mathcal{E}_0, \varnothing) \vdash (\mathcal{E}_1, \mathcal{R}_1) \vdash (\mathcal{E}_2, \mathcal{R}_2) \vdash \cdots$ and let $\mathcal{C}$ be a subset of $\bigcup_i \mathrm{CCP}(\mathcal{E}_i, \mathcal{R}_i)$, where* CCP *is computed with respect to $\succcurlyeq_{\mathsf{NKB}}$. If $\mathrm{CP}_{\mathcal{L}}(\mathcal{R}_\omega, \mathcal{R}^e_\omega) \setminus \mathcal{C} \subseteq \bigcup_i \mathcal{E}_i$ then γ is fair.*

Proof. Induction on $\succcurlyeq_{\mathsf{NKB}}$ shows that any proof in $\mathcal{T} \cup \mathcal{R}_\omega$ can be transformed into a normalized rewrite proof.

Any non-rewrite proof must contain (i) a peak $s \;{}_{\mathcal{R}_\omega/\mathrm{AC}}\!\leftarrow \cdot \rightarrow_{\mathcal{R}_\omega/\mathrm{AC}} t$, or (ii) a peak $s \;{}_{\mathcal{R}_\omega/\mathrm{AC}}\!\leftarrow \cdot \rightarrow_{\mathcal{S}/\mathrm{AC}} t$, or (iii) a subproof $u \rightarrow_{\mathcal{R}_\omega/\mathrm{AC}} t$ such that $u \neq u{\downarrow}$, or (iv) a peak $s \;{}_{\mathcal{S}/\mathrm{AC}}\!\leftarrow \cdot \rightarrow_{\mathcal{S}/\mathrm{AC}} t$. In the latter three cases existence of a smaller proof can be argued as in Lemma 6.15. This also holds for (i) if the peak is a non-overlap, or if it is a proper overlap and the respective critical pair occurs in $\bigcup_i \mathcal{E}_i$. In all these cases this smaller proof can thus be transformed into a rewrite proof by the induction hypothesis. It remains to consider the subcase of (i) where there are a proof $P\colon s \;{}_{\mathcal{R}_\omega/\mathrm{AC}}\!\leftarrow u \rightarrow_{\mathcal{R}_\omega/\mathrm{AC}} t$ and a critical pair $\ell \simeq r \in \mathrm{CP}_{\mathcal{L}}(\mathcal{R}_\omega, \mathcal{R}^e_\omega)$ such that $s \leftrightarrow^*_{\mathcal{L}} C[\ell\sigma] \leftrightarrow_{\ell \approx r} C[r\sigma] \leftrightarrow^*_{\mathcal{L}} t$ but $\ell \simeq r$ does not occur in any set $\mathcal{E}_i$. Hence we must have $\ell \simeq r \in \mathrm{CCP}(\mathcal{E}_i, \mathcal{R}_i)$ for some i. Let the corresponding critical overlap be $P'\colon \ell \leftarrow v \leftrightarrow^*_{\mathcal{L}} v' \rightarrow r$, so $P = P[C[P'\sigma]]$. By definition, there are terms $v_0, \ldots, v_{n+1}$ such that $\ell = v_0$, $r = v_{n+1}$ and $v \succ v_i$, and $(\mathcal{E}_i, \mathcal{R}_i)$ admits proofs P_i of $v_i \approx v_{i+1}$ which are

smaller than P'. By the Persistence Lemma 3.4 there are respective proofs P_i' in $\mathcal{T} \cup \mathcal{R}_\omega$ such that $P_i \ (\Rightarrow_{\mathsf{NKB}}^{\succ})^= P_i'$. By the induction hypothesis all these proofs P_i' can be transformed into normalized rewrite proofs Q_i in $\mathcal{T} \cup \mathcal{R}_\omega$. Consequently all terms in the combined proof $Q\colon Q_1 \cdots Q_n$ of $\ell \approx r$ must be smaller than v, so $P' \Rightarrow_{\mathsf{NKB}}^{\succ} Q$ and hence $P = P[C[P'\sigma]] \Rightarrow_{\mathsf{NKB}}^{\succ} P[C[Q\sigma]]$. Hence, as P can be transformed into a smaller proof it can be transformed into a normalized rewrite proof by the induction hypothesis. □

An analogous proof can be used to derive a corresponding result for NKB', which relaxes Lemma 6.38.

Lemma 6.43. *Consider a nonfailing finite* NKB' *run* $\gamma\colon (\mathcal{E}_0, \varnothing) \vdash (\mathcal{E}_1, \mathcal{R}_1) \vdash \cdots \vdash (\mathcal{E}_n, \mathcal{R}_n)$ *and let* $\mathcal{C}$ *be a subset of* $\bigcup_i \mathrm{CCP}(\mathcal{E}_i, \mathcal{R}_i)$*, where* CCP *is computed with respect to* $\succ\!\!\succ_{\mathsf{NKB}}^n$*. If* $\mathrm{CP}_{\mathcal{L}}(\mathcal{R}_n, \mathcal{R}_n^e) \setminus \mathcal{C} \subseteq \bigcup_i \mathcal{E}_i$ *then* γ *is fair.* □

Several special cases of this general criterion can be checked efficiently. Consider an overlap o of the form $\langle \ell_1 \to r_1, p, \ell_2 \to r_2 \rangle_\Sigma$ giving rise to the set of critical peaks

$$P\colon s \xleftarrow[r_1 \leftarrow \ell_1]{p,\sigma} u \overset{*}{\underset{\mathcal{L}}{\leftrightarrow}} u' \xrightarrow[\ell_2 \to r_2]{\epsilon,\sigma} t \tag{6.8}$$

such that $\sigma \in \Sigma$. Let $c_{\mathcal{L}}(P)$ refer to the cost of the subproof $u \leftrightarrow_{\mathcal{L}}^* u'$.

$\mathcal{S}$-reducibility

If $u \neq u{\downarrow}$ or $u' \neq u'{\downarrow}$ in an overlap of the form (6.8) then the corresponding critical pair is called *$\mathcal{S}$-reducible*. The $\mathcal{S}$-reducibility criterion $\mathrm{SCP}_{\mathcal{L}}(\mathcal{E}, \mathcal{R})$ returns all $\mathcal{S}$-reducible $\mathcal{L}$-critical pairs among rules in $\mathcal{R}$.

Lemma 6.44. *Every $\mathcal{S}$-reducible $\mathcal{L}$-critical pair of the form* (6.8) *is composite with respect to* $\succ\!\!\succ_{\mathsf{NKB}}$ *and* $\succ\!\!\succ_{\mathsf{NKB}}^n$.

Proof. Let P be an overlap of the form (6.8), and assume $u \to_{\mathcal{S}/\mathrm{AC}} v$. We thus also have another equational proof $P_1 P_2$ of $s \approx t$, with

$$P_1\colon s \xleftarrow[r_1 \leftarrow \ell_1]{p,\sigma} u \xrightarrow[\mathcal{S}/\mathrm{AC}]{} v \qquad\qquad P_2\colon v \xleftarrow[\mathcal{S}/\mathrm{AC}]{} u \leftrightarrow_{\mathcal{L}}^* u' \xrightarrow[\ell_2 \to r_2]{\epsilon,\sigma} t$$

As u is $\mathcal{S}/\mathrm{AC}$-reducible we have $c(u', \epsilon, t) = \{u', t\}$, such that for both NKB and NKB' the proof costs amount to

$$\begin{aligned} c(P) &= \{(c(u,p,s), \ldots), (\{u', t\}, \ldots)\} \cup c_{\mathcal{L}}(P) \\ c(P_1) &= \{(c(u,p,s), \ldots), (\bot, \ldots)\} \cup c_{\mathrm{AC}}(P_1) \\ c(P_2) &= \{(\bot, \ldots), (\{u', t\}, \ldots)\} \cup c_{\mathrm{AC}}(P_2) \cup c_{\mathcal{L}}(P) \end{aligned}$$

where $c_{\mathrm{AC}}(P_i)$ corresponds to the complexities of possibly required AC-steps in $u \to_{\mathcal{S}/\mathrm{AC}} v$. Note that the complexities of AC steps are smaller than the first two cost tuples in $c(P)$. We have $P \succ\!\!\succ_{\mathsf{NKB}} P_1$ and $P \succ\!\!\succ_{\mathsf{NKB}} P_2$ ($P \succ\!\!\succ_{\mathsf{NKB}}^n P_1$ and $P \succ\!\!\succ_{\mathsf{NKB}}^n P_2$), so the AC-critical pair is composite for NKB (NKB'). A symmetric argument shows compositeness of any critical pair where u' is $\mathcal{S}/\mathrm{AC}$-reducible. □

In the sequel we thus assume that both u and u' are in normal form with respect to $\to_{\mathcal{S}/\mathrm{AC}}$. By AC convergence of $\mathcal{S}$ and $\mathcal{L} \subseteq \mathcal{S}$ we thus have $u \leftrightarrow^*_{\mathrm{AC}} u'$. Now assume there is a rewrite step $u \leftrightarrow^*_{\mathrm{AC}} \cdot \to_{\mathcal{R}} v$ using a rule $\ell_3 \to r_3$ at position q, such that $(\ell_3 \to r_3, q)$ is different from $(\ell_1 \to r_1, p)$ and $(\ell_2 \to r_2, \epsilon)$. Thus there are proofs

$$P_1 \colon s \xleftarrow[r_1 \leftarrow \ell_1]{p} u \xleftrightarrow[\mathrm{AC}]{*} v' \xrightarrow[\ell_3 \to r_3]{q} v \qquad P_2 \colon v \xleftarrow[r_3 \leftarrow \ell_3]{q} v' \xleftrightarrow[\mathrm{AC}]{*} u' \xrightarrow[\ell_1 \to r_1]{\epsilon} t \tag{6.9}$$

such that $P_1 P_2$ proves $s \approx t$.

Primality

An AC-critical pair (6.8) is *not prime* if there are proofs (6.9) such that $u|_p \rhd_{\mathrm{AC}} v'|_q$. The primality criterion $\mathrm{PCP}_{\mathrm{AC}}(\mathcal{E}, \mathcal{R})$ returns all AC-critical pairs among rules in $\mathcal{R}$ for which the associated overlaps are not prime.

Lemma 6.45. *Every non-prime $\mathcal{L}$-critical pair is composite with respect to $\succcurlyeq_{\mathsf{NKB}}$ and $\succcurlyeq^n_{\mathsf{NKB}}$.*

Proof. As u, u', and v' are in $\mathcal{S}$-normal form, the proof costs in (6.8) and (6.9) have the shape

$$\begin{aligned} c(P) &= \{(\{u\}, \{u\}, u|_p, \ldots), (\{u'\}, \{u'\}, u', \ldots)\} \cup c_{\mathrm{AC}}(P) \\ c(P_1) &= \{(\{u\}, \{u\}, u|_p, \ldots), (\{v'\}, \{v'\}, v'|_q, \ldots)\} \cup c_{\mathrm{AC}}(P_1) \\ c(P_2) &= \{(\{u'\}, \{u'\}, u', \ldots), (\{v'\}, \{v'\}, v'|_q, \ldots)\} \cup c_{\mathrm{AC}}(P_2) \end{aligned}$$

for the cost measures underlying both NKB and NKB'. From $u' \leftrightarrow^*_{\mathrm{AC}} v'$ we obtain $\{u'\} \succeq_{\mathsf{mul}} \{v'\}$. Therefore $u' \leftrightarrow^*_{\mathrm{AC}} u \trianglerighteq u|_p \rhd_{\mathrm{AC}} v'|_q$ and thus $u' \rhd_{\mathrm{AC}} v'|_q$, so we have $P \succcurlyeq_{\mathsf{NKB}} P_1$ and $P \succcurlyeq^n_{\mathsf{NKB}} P_1$. Furthermore, as $u|_p \rhd_{\mathrm{AC}} v'|_q$ we have $P \succcurlyeq_{\mathsf{NKB}} P_2$ and $P \succcurlyeq^n_{\mathsf{NKB}} P_2$. It follows that P is composite. □

As in standard completion, the unblockedness criterion $\mathrm{BCP}_{\mathrm{AC}}(\mathcal{E}, \mathcal{R})$ forms a special case of the primality criterion. This criterion considers all critical pairs superfluous where $x\sigma$ is $\mathcal{R}$/AC-reducible for some variable x occurring in ℓ_1 or ℓ_2.

Connectedness

Also a normalized completion variant of the connectedness criterion [61] can be defined. A critical pair originating from an overlap (6.8) is *connected below u* if there exists a sequence of single-step proofs $s = u_0 \leftrightarrow u_1 \leftrightarrow \cdots \leftrightarrow u_{n+1} = t$ in $(\mathcal{T}, \mathcal{E}, \mathcal{R})$ such that $u \succ u_i$ for all $1 \leqslant i \leqslant n$. Note that connectedness below u coincides with connectedness below u' due to AC compatibility of $\succ$.

Lemma 6.46. *If the $\mathcal{L}$-critical pair $s \leftarrow \rtimes \rightarrow t$ corresponding to an overlap* (6.8) *is connected below u then it is composite with respect to $\succcurlyeq_{\mathsf{NKB}}$ and $\succcurlyeq^n_{\mathsf{NKB}}$.*

Proof. Let P_i denote the single-step proof $u_i \leftrightarrow u_{i+1}$ in $(\mathcal{T}, \mathcal{E}, \mathcal{R})$ for all $0 \leqslant i \leqslant n$. Let $\succcurlyeq$ be one of $\succcurlyeq_{\mathsf{NKB}}$ or $\succcurlyeq^n_{\mathsf{NKB}}$. For both proof orders we

have $c(P) = \{(\{u\}, \ldots), (\{u'\}, \ldots)\} \cup c_{\mathrm{AC}}(P)$ while $c(P_i)$ contains only a single tuple of the form $(\{u_i\}, \ldots)$, $(\{u_{i+1}\}, \ldots)$, $(\{u_i, u_{i+1}\}, \ldots)$, or $(\bot, \ldots)$. Since $u \succ u_i, u_{i+1}$ we have $P \not\succcurlyeq P_i$ for all $0 \leqslant i \leqslant n$. As $P_0 \cdots P_n$ proves $s \approx t$, the $\mathcal{L}$-critical pair is composite. □

A criterion $\mathrm{WCP}_{\mathrm{AC}}(\mathcal{E}, \mathcal{R})$ performing the weak connectivity test is obtained in a similar way as for standard completion. Assume the critical pair corresponding to an overlap (6.8) admits a decomposition of the form (6.9). If both proofs P_1 and P_2 are either not proper overlaps or the respective critical pair was already considered, then the critical pair is *weakly connected*. It is easy to see that weak connectedness constitutes a special case of connectedness.

Since $\mathcal{S}$-reducibility, non-primality, and (weak) connectedness capture special cases of compositeness, these criteria can also be combined. For a critical pair $s \leftarrow \rtimes \rightarrow t$ originating from an overlap of the form (6.8), one can obtain a more powerful *mixed* criterion $\mathrm{MCP}_{\mathcal{L}}(\mathcal{E}, \mathcal{R})$ by performing the following checks:

(a) If u or u' is not in $\mathcal{S}$-normal form, then $s \approx t$ is composite as it is $\mathcal{S}$-reducible.

(b) Otherwise, if u' is reducible with some $\mathcal{R}$/AC-step strictly below the position p of the overlap then $s \approx t$ is composite as it is non-prime.

(c) Otherwise, one checks whether u or u' is (in addition to the rewrite steps involved in the overlap) $\mathcal{R}$/AC-reducible with some rule $\ell_3 \to r_3$ at position q. If there exists such a reduction such that for both $i \in \{1, 2\}$ the rules $\ell_i \to r_i$ and $\ell_3 \to r_3$ do either not form a proper overlap, or the corresponding critical pair was already considered then $s \approx t$ is composite according to the weak connectivity test.

6.2 Normalized Completion with Termination Tools

The inference rules in Figure 6.5 describe $\mathcal{T}$-normalized completion with termination tools (abbreviated NKBtt). In the orient rule, (Θ, Ψ) is again assumed to form an $\mathcal{S}$-normalizing pair for the terms s and t. A sequence $(\mathcal{E}_0, \varnothing, \varnothing) \vdash (\mathcal{E}_1, \mathcal{R}_1, \mathcal{C}_1) \vdash (\mathcal{E}_2, \mathcal{R}_2, \mathcal{C}_2) \vdash \cdots$ of NKBtt inference steps is called a *run*. Before giving a correctness proof we illustrate NKBtt on an example.

Example 6.47. Consider the initial set of equations $\mathcal{E}_0 = \{\mathsf{a} + x \approx \mathsf{b} + \mathsf{g}(\mathsf{a})\}$ where $+$ is an AC symbol with unit $\mathsf{0}$, such that the theory $\mathcal{T}$ can be represented by $\mathcal{S} = \{x + \mathsf{0} \to x\}$. Note that the given equation cannot be oriented with an AC-compatible simplification order. Thus any completion tool restricted to orders such as AC-RPO or AC-KBO [59] fails immediately. But termination tools can verify AC termination of the rule $\mathsf{a} + x \to \mathsf{b} + \mathsf{g}(\mathsf{a})$ using e.g. AC dependency pairs [2]. Hence the equation $\mathsf{a} + x \approx \mathsf{b} + \mathsf{g}(\mathsf{a})$ can be oriented in an NKBtt run. When using ACU-normalizing pairs (see Definition 6.22), this results in the state

$$\mathcal{E}_1\colon \quad \mathsf{a} + \mathsf{0} \approx \mathsf{b} + \mathsf{g}(\mathsf{a}) \quad \mathcal{R}_1\colon \quad \mathsf{a} + x \to \mathsf{b} + \mathsf{g}(\mathsf{a}) \quad \mathcal{C}_1\colon \quad \mathsf{a} + x \to \mathsf{b} + \mathsf{g}(\mathsf{a})$$

orient	$\dfrac{\mathcal{E} \uplus \{s \simeq t\}, \mathcal{R}, \mathcal{C}}{\mathcal{E} \cup \Theta(s,t), \mathcal{R} \cup \Psi(s,t), \mathcal{C}'}$	if $s = s{\downarrow}$, $t = t{\downarrow}$ and $\mathcal{C}' \cup \mathcal{S}$ is AC terminating for $\mathcal{C}' = \mathcal{C} \cup \Psi(s,t)$
deduce	$\dfrac{\mathcal{E}, \mathcal{R}, \mathcal{C}}{\mathcal{E} \cup \{s \approx t\}, \mathcal{R}, \mathcal{C}}$	if $s \approx t \in \mathrm{CP}_{\mathcal{L}}(\mathcal{R}, \mathcal{R}^e)$
delete	$\dfrac{\mathcal{E} \uplus \{s \approx t\}, \mathcal{R}, \mathcal{C}}{\mathcal{E}, \mathcal{R}, \mathcal{C}}$	if $s \leftrightarrow^*_{\mathrm{AC}} t$
normalize	$\dfrac{\mathcal{E} \uplus \{s \approx t\}, \mathcal{R}, \mathcal{C}}{\mathcal{E} \cup \{s{\downarrow} \approx t{\downarrow}\}, \mathcal{R}, \mathcal{C}}$	if $s \neq s{\downarrow}$ or $t \neq t{\downarrow}$
simplify	$\dfrac{\mathcal{E} \uplus \{s \simeq t\}, \mathcal{R}, \mathcal{C}}{\mathcal{E} \cup \{s \simeq u\}, \mathcal{R}, \mathcal{C}}$	if $t \to_{\mathcal{R} \setminus \mathcal{S}} u$
compose	$\dfrac{\mathcal{E}, \mathcal{R} \uplus \{s \to t\}, \mathcal{C}}{\mathcal{E}, \mathcal{R} \cup \{s \to u\}, \mathcal{C}}$	if $t \to_{\mathcal{R} \setminus \mathcal{S}} u$
collapse	$\dfrac{\mathcal{E}, \mathcal{R} \uplus \{t \to s\}, \mathcal{C}}{\mathcal{E} \cup \{u \approx s\}, \mathcal{R}, \mathcal{C}}$	if $t \to_{\mathcal{R} \setminus \mathcal{S}} u$

Figure 6.5: Normalized completion with termination tools (NKBtt).

After normalizing $\mathsf{a} + \mathsf{0}$ to a, we have

$$\mathcal{E}_2\colon \quad \mathsf{a} \approx \mathsf{b} + \mathsf{g}(\mathsf{a}) \qquad \mathcal{R}_2\colon \quad \mathsf{a} + x \to \mathsf{b} + \mathsf{g}(\mathsf{a}) \qquad \mathcal{C}_2\colon \quad \mathsf{a} + x \to \mathsf{b} + \mathsf{g}(\mathsf{a})$$

Since $\mathcal{C}_2 \cup \{\mathsf{b} + \mathsf{g}(\mathsf{a}) \to \mathsf{a}\}$ is AC terminating, we may perform an orient step:

$$\mathcal{E}_3\colon \qquad \mathcal{R}_3\colon \begin{array}{r} \mathsf{a} + x \to \mathsf{b} + \mathsf{g}(\mathsf{a}) \\ \mathsf{b} + \mathsf{g}(\mathsf{a}) \to \mathsf{a} \end{array} \qquad \mathcal{C}_3\colon \begin{array}{r} \mathsf{a} + x \to \mathsf{b} + \mathsf{g}(\mathsf{a}) \\ \mathsf{b} + \mathsf{g}(\mathsf{a}) \to \mathsf{a} \end{array}$$

In a subsequent compose step, the new rule can be used to reduce the first one:

$$\mathcal{E}_4\colon \qquad \mathcal{R}_4\colon \begin{array}{r} \mathsf{a} + x \to \mathsf{a} \\ \mathsf{b} + \mathsf{g}(\mathsf{a}) \to \mathsf{a} \end{array} \qquad \mathcal{C}_4\colon \begin{array}{r} \mathsf{a} + x \to \mathsf{b} + \mathsf{g}(\mathsf{a}) \\ \mathsf{b} + \mathsf{g}(\mathsf{a}) \end{array}$$

endenumeratetoa

Three applications of deduce yield the state

$$\mathcal{E}_7\colon \begin{array}{r} \mathsf{a} + \mathsf{g}(\mathsf{a}) \approx \mathsf{a} + \mathsf{a} \\ \mathsf{a} + \mathsf{a} \approx \mathsf{a} + \mathsf{b} \\ \mathsf{a} + \mathsf{a} \approx \mathsf{a} \end{array} \qquad \mathcal{R}_7\colon \begin{array}{r} \mathsf{a} + x \to \mathsf{a} \\ \mathsf{b} + \mathsf{g}(\mathsf{a}) \to \mathsf{a} \end{array} \qquad \mathcal{C}_7\colon \begin{array}{r} \mathsf{a} + x \to \mathsf{b} + \mathsf{g}(\mathsf{a}) \\ \mathsf{b} + \mathsf{g}(\mathsf{a}) \to \mathsf{a} \end{array}$$

Since all terms in $\mathcal{E}_7$ simplify to a, the resulting trivial equations can be deleted. As all critical pairs among rules in $\mathcal{R}_7$ were already deduced the run is fair, and $\mathcal{R}_7 = \{\mathsf{a} + x \to \mathsf{a},\ \mathsf{b} + \mathsf{g}(\mathsf{a}) \to \mathsf{a}\}$ is $\mathcal{S}$-convergent for $\mathcal{E}_0$.

We now show that NKBtt simulates NKB runs and vice versa.

Simulation Soundness Lemma 6.48. *Any run* $(\mathcal{E}_0, \varnothing, \varnothing) \vdash^n_{\mathsf{NKBtt}} (\mathcal{E}_n, \mathcal{R}_n, \mathcal{C}_n)$ *admits an* NKB′ *run* $(\mathcal{E}_0, \varnothing) \vdash^n_{\mathsf{NKB}'} (\mathcal{E}_n, \mathcal{R}_n)$ *using the AC-compatible reduction order* $\to^+_{(\mathcal{C}_n \cup \mathcal{S})/\mathrm{AC}}$.

Proof. Note that all TRSs $\mathcal{C}_i \cup \mathcal{S}$ are AC terminating. The relations $\to^+_{(\mathcal{C}_i \cup \mathcal{S})/\mathrm{AC}}$ are thus AC-compatible reduction orders, which we abbreviate by $\succ_i$. We prove the claim by induction on n, which is trivial for $n = 0$. For an NKBtt run $(\mathcal{E}_0, \varnothing, \varnothing) \vdash^* (\mathcal{E}_n, \mathcal{R}_n, \mathcal{C}_n) \vdash (\mathcal{E}_{n+1}, \mathcal{R}_{n+1}, \mathcal{C}_{n+1})$, the induction hypothesis yields a normalized completion run $(\mathcal{E}_0, \mathcal{R}_0) \vdash^* (\mathcal{E}_n, \mathcal{R}_n)$ using reduction order $\succ_n$. Since constraint rules are never removed we have $\mathcal{C}_n \subseteq \mathcal{C}_{n+1}$, so the same run can be obtained with $\succ_{n+1}$. Case distinction on the applied NKBtt rule shows that a step $(\mathcal{E}_n, \mathcal{R}_n) \vdash (\mathcal{E}_{n+1}, \mathcal{R}_{n+1})$ using $\succ_{n+1}$ is possible in NKB: If orient is applied to $s \simeq t$ then $\Psi(s,t) \subseteq \succ_{n+1}$ by definition, so NKB can apply orient as well. In all remaining cases the step can obviously be simulated by the corresponding NKB rule as no conditions on the order are involved. □

Simulation Completeness Lemma 6.49. *If* $(\mathcal{E}_0, \varnothing) \vdash^n_{\mathsf{NKB}'} (\mathcal{E}_n, \mathcal{R}_n)$ *is a valid* NKB′ *run using an AC-compatible reduction order* $\succ$ *then there is also a valid* NKBtt *run* $(\mathcal{E}_0, \varnothing, \varnothing) \vdash^n_{\mathsf{NKBtt}} (\mathcal{E}_n, \mathcal{R}_n, \mathcal{C}_n)$ *such that* $\mathcal{C}_n \subseteq \succ$.

Proof. By induction on n. For $n = 0$ the claim is trivially satisfied by setting $\mathcal{C}_0 = \varnothing$. So suppose $(\mathcal{E}_0, \varnothing) \vdash^n_{\mathsf{NKB}'} (\mathcal{E}_n, \mathcal{R}_n) \vdash (\mathcal{E}_{n+1}, \mathcal{R}_{n+1})$. The induction hypothesis yields an NKBtt run $(\mathcal{E}_0, \varnothing, \varnothing) \vdash^* (\mathcal{E}_n, \mathcal{R}_n, \mathcal{C}_n)$ such that $\mathcal{C}_n \subseteq \succ$. An easy case distinction on the last inference step $(\mathcal{E}_n, \mathcal{R}_n) \vdash (\mathcal{E}_{n+1}, \mathcal{R}_{n+1})$ shows that using $\succ$ for AC-termination checks allows for a corresponding NKBtt step: If the applied inference rule is orient we have $\mathcal{E}_n = \mathcal{E}_{n+1} \cup \{s \simeq t\}$ and $\mathcal{R}_{n+1} = \mathcal{R}_n \cup \Psi(s,t)$ such that $\Psi(s,t) \subseteq \succ$ as (Θ, Ψ) constitutes a normalizing pair. Thus for $\mathcal{C}' = \mathcal{C}_n \cup \Psi(s,t)$ also $\mathcal{C}' \subseteq \succ$ is satisfied, ensuring AC termination of the system $\mathcal{C}' \cup \mathcal{S}$ because $\mathcal{S} \subseteq \succ$ by assumption. Hence the NKBtt inference rule orient can be applied to obtain $(\mathcal{E}_n, \mathcal{R}_n, \mathcal{C}_n) \vdash (\mathcal{E}_n \setminus \{s \simeq t\}, \mathcal{R}_n \cup \Psi(s,t), \mathcal{C}')$. In the remaining cases one can set $\mathcal{C}_{n+1} = \mathcal{C}_n$ and replace the applied rule by its NKBtt counterpart since no conditions on the order are involved. □

Correctness Theorem 6.50. *Any finite nonfailing and fair* NKBtt *run succeeds.*

Proof. Let $(\mathcal{E}_0, \varnothing, \varnothing) \vdash^n (\varnothing, \mathcal{R}_n, \mathcal{C}_n)$ be a finite and fair run. According to Lemma 6.48 the same TRS $\mathcal{R}_n$ can be derived in a fair and nonfailing NKB′ run using the reduction order $\to^+_{(\mathcal{C}_n \cup \mathcal{S})/\mathrm{AC}}$. By Theorem 6.16 the TRS $\mathcal{R}_n$ is $\mathcal{S}$-convergent for $\mathcal{E}_0$. □

Example 6.51. Consider the following set of equations (adapted from [77]) describing addition on natural numbers represented in binary:

$$\#0 \approx \# \qquad (x+y)1 \approx x0 + y1 \qquad \mathsf{triple}(x) \approx x0 + x$$
$$(x+y)0 \approx x0 + y0 \qquad x0 + y0 + \#10 \approx x1 + y1$$

where + is an AC operator, 0 and 1 are unary operators in postfix notation, and # denotes the empty bit sequence. For example, #100 represents the number 4 in binary. When applying normalized completion with termination tools modulo AC, the following AC-convergent system is produced.

$$\begin{aligned}
\#0 &\to \# & \mathsf{triple}(x) &\to x0 + x \\
(x + \#)0 &\to x0 + \# & (x + \#)1 &\to x1 + \# \\
x0 + y0 &\to (x + y)0 & x0 + y0 + z &\to (x + y)0 + z \\
x0 + y1 &\to (x + y)1 & x0 + y1 + z &\to (x + y)1 + z \\
x1 + y1 &\to (x + y + \#1)0 & x1 + y1 + z &\to (x + y + \#1)0 + z
\end{aligned}$$

However, normalized completion using AC-RPO or AC-KBO does not succeed.

6.3 Normalized Multi-Completion with Termination Tools

Normalized multi-completion with termination tools closely resembles MKBtt and oMKBtt in that it simulates multiple NKBtt runs in parallel but exploits sharing to gain efficiency. We define normalized multi-completion with termination tools by the inference system MNKBtt working on sets of nodes. Figure 6.6 displays the inference rules of MNKBtt, where we use the notation $N_{\mathcal{E}}^{P}$ to abbreviate $\{\langle u : v, \varnothing, \varnothing, P, \varnothing, \varnothing\rangle \mid u \approx v \in \mathcal{E}\}$. In addition also the optional MKBtt rules gc and subsume (see Figure 4.4) can be used. For the sake of simplicity we assume that $\Psi(s,t) = \{s \to t\}$, as is the case for most theories under consideration. Otherwise, an orient step can add all nodes $\langle \ell : r, R_{lr}, \varnothing, \varnothing, \varnothing, \varnothing\rangle$ such that $\ell \to r \in \Psi(s,t)$ and $\langle \ell' : r', R_{rl}, \varnothing, \varnothing, \varnothing, \varnothing\rangle$ such that $\ell' \to r' \in \Psi(t,s)$.

An MNKBtt inference sequence $N_0 \vdash N_1 \vdash N_2 \vdash \cdots \vdash N_n$ where $N_0 = N_{\mathcal{E}}$ for some set of equations $\mathcal{E}$ is called a *run*. We establish some simple properties of MNKBtt runs.

Lemma 6.52. *Consider an* MNKBtt *run* $N_0 \vdash N_1 \vdash N_2 \vdash \cdots \vdash N_k$ *and a process* $p \in \mathcal{P}(N_k)$. *Then* $\mathcal{S} \cup C[N_k, p]$ *is AC terminating and* $R[N_k, p] \subseteq \to^{+}_{C[N_k,p]\cup\mathcal{S}}$.

Proof. By induction on k. In the base case $N_0 = N_{\mathcal{E}}$ for some set of equations $\mathcal{E}$. For the single process ϵ occurring in N_0 we have $R[N_0, \epsilon] = C[N_0, \epsilon] = \varnothing$, and $\mathcal{S}$ is AC terminating by assumption. In the induction step, assuming that the claim holds for N_k, a case distinction on the rule applied in $N_k \vdash N_{k+1}$ shows that it is also true for N_{k+1}. We use the notation from Figure 6.6 and abbreviate $\to^{+}_{C[N_k,p]\cup\mathcal{S}}$ by $\succ_k$.

- Assume orient is applied to a node $\langle s : t, R_0, R_1, E, C_0, C_1\rangle$. For a process $p \in \mathcal{P}(N_{k+1}) \setminus (R_{lr} \cup R_{rl})$ the claim holds as $R[N_k, p] = R[N_{k+1}, p]$ and $C[N_k, p] = C[N_{k+1}, p]$. If $p \in R_{lr}$ then $p \in E_{lr} \setminus E_{rl}$ or $p = q0$ such that $q \in U$. Let in the former case $p' = p$ and in the latter $p' = q$. We then have for both cases $C[N_{k+1}, p] = C[N_k, p'] \cup \{s \to t\}$, $C[N_{k+1}, p] \cup \mathcal{S}$ is AC terminating, and $R[N_{k+1}, p] = R[N_k, p'] \cup \{s \to t\}$, so with the induction

orient

$$\frac{N \uplus \{\langle s : t, R_0, R_1, E, C_0, C_1\rangle\}}{\mathit{split}_U(N) \cup \{\langle s : t, R_0 \cup R_{lr}, R_1 \cup R_{rl}, E', C_0 \cup R_{lr}, C_1 \cup R_{rl}\rangle\} \cup N^{R_{lr}}_{\Theta(s,t)} \cup N^{R_{rl}}_{\Theta(t,s)}}$$

if
- $s = s{\downarrow}$ and $t = t{\downarrow}$,
- $E_{lr}, E_{rl} \subseteq E$, and $E' = E \setminus (E_{lr} \cup E_{rl})$,
- $C[N,p] \cup \mathcal{S} \cup \{s \to t\}$ is AC terminating for all $p \in E_{lr}$ and $C[N,p] \cup \mathcal{S} \cup \{t \to s\}$ is AC terminating for all $p \in E_{rl}$,
- $R_{lr} = (E_{lr} \setminus E_{rl}) \cup \{p0 \mid p \in U\}$, and $R_{rl} = (E_{rl} \setminus E_{lr}) \cup \{p1 \mid p \in U\}$ for $U = E_{lr} \cap E_{rl}$, and
- $E_{lr} \cup E_{rl} \neq \varnothing$

deduce

$$\frac{N}{N \cup \{\langle s : t, \varnothing, \varnothing, R \cap R', \varnothing, \varnothing\rangle\}}$$

if
- $\langle \ell : r, R, \dots\rangle$, $\langle \ell' : r', R', \dots\rangle \in N$,
- $s \leftarrow \rtimes \rightarrow t \in \mathrm{CP}_{\mathcal{L}}(\ell \to r, (\ell' \to r')^e)$, and
- $R \cap R' \neq \varnothing$

delete

$$\frac{N \uplus \{\langle s : t, \varnothing, \varnothing, E, \varnothing, \varnothing\rangle\}}{N}$$

if $s \leftrightarrow^*_{\mathrm{AC}} t$ and $E \neq \varnothing$

normalize

$$\frac{N \uplus \{\langle s : t, R_0, R_1, E, C_0, C_1\rangle\}}{N \cup \{\langle s{\downarrow} : t{\downarrow}, \varnothing, \varnothing, E, \varnothing, \varnothing\rangle,\ \langle s : t, R_0, R_1, \varnothing, C_0, C_1\rangle\}}$$

if $s \neq s{\downarrow}$ or $t \neq t{\downarrow}$

rewrite

$$\frac{N \uplus \{\langle s : t, R_0, R_1, E, C_0, C_1\rangle\}}{N \cup \{\langle s : t, R_0 \setminus R, R_1 \setminus R, E \setminus R, C_0, C_1\rangle\} \cup \{\langle s : u, R_0 \cap R, \varnothing, (R_1 \cup E) \cap R, \varnothing, \varnothing\rangle\}}$$

if
- $\langle \ell : r, R, \dots\rangle \in N$ such that $t \to_{\ell \to r \setminus \mathcal{S}} u$, and
- $R \cap (R_0 \cup R_1 \cup E) \neq \varnothing$

Figure 6.6: Normalized multi-completion with termination tools (MNKBtt).

hypothesis we obtain $R[N_{k+1}, p] \subseteq \succ_{k+1}$. The argument for a process in R_{rl} is symmetric.

– Suppose rewrite is applied. We have $C[N_{k+1}, p] = C[N_k, p]$ and therefore $\succ_{k+1} = \succ_k$, and the constraint system $C[N_{k+1}, p] \cup \mathcal{S}$ is AC terminating by the induction hypothesis for all $p \in \mathcal{P}(N_{k+1})$.

For $p \in R \cap R_0$ we have $R[N_{k+1}, p] = (R[N_k, p] \setminus \{s \to t\}) \cup \{s \to u\}$, and $R[N_k, p]$ contains rules $s \to t$ and $\ell \to r$ such that $t \to_{\ell \to r \setminus \mathcal{S}} u$. As $R[N_k, p] \subseteq \succ_k$ we have $s \succ_k t$ and $\ell \succ_k r$. Since $\succ_k$ is a reduction order and $\mathcal{S} \subseteq \succ_k$ also $s \succ_k u$ (and thus $s \succ_{k+1} u$) holds. For $p \in R \cap (R_1 \cup E)$ we have $R[N_{k+1}, p] \subseteq R[N_k, p]$, so the proof obligation is implied by the induction hypothesis.

– If delete, deduce, normalize, or gc was applied then $C[N_{k+1}, p] = C[N_k, p]$ and $R[N_k, p] = R[N_{k+1}, p]$ for all $p \in \mathcal{P}(N_{k+1})$, so the claim follows from the induction hypothesis.

– The case of subsume can be argued as for MKBtt (see Lemma 4.24). □

We can hence immediately conclude that MNKBtt produces terminating rewrite systems:

Corollary 6.53. *In an* MNKBtt *run* $N_0 \vdash N_1 \vdash N_2 \vdash \cdots$ *the rewrite system* $R[N_k, p] \cup \mathcal{S}$ *is AC terminating for all node sets* N_k *and processes* $p \in \mathcal{P}(N_k)$. □

Since the inference rules of MNKBtt modify labels in the same way as the MKBtt rules the proof of the following result is identical to Lemma 4.26.

Lemma 6.54. *In an* MNKBtt *run* $N_0 \vdash N_1 \vdash N_2 \vdash \cdots$ *every node set* N_k *is well-encoded and satisfies the node condition.* □

The *split set* of an MNKBtt step and the *predecessor* of a process with respect to an inference step are defined as in Definition 4.27. Lemmas 6.55 and 6.56 show that an MNKBtt step corresponds to a (possibly empty) NKBtt step for all processes occurring in nodes, and conversely every NKBtt step can be modelled by MNKBtt. The relation $\vdash^=$ denotes the reflexive closure of the NKBtt inference relation $\vdash$.

Lemma 6.55. *Let* N, N' *be well-encoded node sets which satisfy the node condition. For an* MNKBtt *step* $N \vdash N'$ *with split set* U *the* NKBtt *step*

$$(E[N, p], R[N, p], C[N, p]) \vdash^= (E[N', p'], R[N', p'], C[N', p']) \tag{6.10}$$

is valid for all $p' \in \mathcal{P}(N')$ *such that* $p = pred_U(p')$. *There exists at least one process* $p' \in \mathcal{P}(N')$ *for which the step is not an equality step if the rule applied in* $N \vdash N'$ *is not* gc *or* subsume.

Proof. By case analysis on the MNKBtt rule applied in the step $N \vdash N'$, where the notation from Figure 6.6 is used.

– Suppose orient was applied to a node $n = \langle s : t, R_0, R_1, E, C_0, C_1 \rangle$, adding $n' = \langle s : t, R_0 \cup R_{lr}, R_1 \cup R_{rl}, E', C_0 \cup R_{lr}, C_1 \cup R_{rl} \rangle$. Hence $s = s{\downarrow}$ and $t = t{\downarrow}$ must hold. Let p' be a process in $\mathcal{P}(N')$ and $p = pred_U(p')$ be its predecessor. Let $\mathcal{E} = E[N \setminus \{n\}, p]$, $\mathcal{R} = R[N \setminus \{n\}, p]$, and $\mathcal{C} = C[N \setminus \{n\}, p]$. Obviously, $R[N' \setminus \{n'\}, p'] = R[N \setminus \{n\}, p]$ and $C[N' \setminus \{n'\}, p'] = C[N \setminus \{n\}, p]$, whereas $E[N' \setminus (\{n'\} \cup N_\Theta), p'] = E[N \setminus \{n\}, p]$ for $N_\Theta = N^{R_{lr}}_{\Theta(s,t)} \cup N^{R_{rl}}_{\Theta(t,s)}$. Three cases can be further distinguished.

i. If $p' \in R_{lr}$ then by definition of orient $R[n', p'] = C[n', p'] = \{s \to t\}$ and $C[N, p] \cup \{s \to t\} \cup \mathcal{S}$ is AC terminating. As R_{lr} and R_{rl} are disjoint, $E[N', p'] \setminus \mathcal{E} = \Theta(s, t)$. Step (6.10) is thus a valid orient step in NKBtt if $p \in E$. Since p' occurs in R_{lr}, either $p' \in E_{lr} \setminus E_{rl}$ or $p' = p0$ for some $p \in U$. If $p' \in E_{lr} \setminus E_{rl}$ then $p \in E$ follows from $p = pred_U(p') = p'$ and $E_{lr} \subseteq E$. Otherwise $p = pred_U(p')$ entails $p' = p0$ such that $p \in U \subseteq E$. As $p \in E$ we have $E[n, p] = \{s \simeq t\}$ and—because of the node condition—$R[n, p] = C[n, p] = \varnothing$. Hence the following orient step is valid in NKBtt:

$$(\mathcal{E} \uplus \{s \simeq t\}, \mathcal{R}, \mathcal{C}) \vdash (\mathcal{E} \cup \Theta(s, t), \mathcal{R} \cup \{s \to t\}, \mathcal{C} \cup \{s \to t\})$$

ii. The case where $p' \in R_{rl}$ is symmetric to the previous one.

iii. No process $p' \notin R_{lr} \cup R_{rl}$ is affected by $N \vdash N'$, so $p = p'$ and we have $E[n, p] = E[n', p']$, $R[n, p] = R[n', p']$ and $C[n, p] = C[n', p']$. The projection of the MNKBtt step to p' is thus an identity step $(\mathcal{E}, \mathcal{R}, \mathcal{C}) \vdash^{=} (\mathcal{E}, \mathcal{R}, \mathcal{C})$.

In all remaining cases $p = p'$ holds as no process splitting occurs.

– Suppose deduce adds a node $\langle s : t, \varnothing, \varnothing, R \cap R', \varnothing, \varnothing \rangle$. Then for all $p \in R \cap R'$ both $\ell_1 \to r_1$ and $\ell_2 \to r_2$ occur in $R[N, p]$, and $s \leftarrow \rtimes \rightarrow t$ constitutes an $\mathcal{L}$-critical pair between the two rules. Hence deduce can also be applied in the step (6.10), where indeed $E[N', p] = E[N, p] \cup \{s \approx t\}$. If $p \notin R \cap R'$ then (6.10) is an identity step.

– If delete removes a node $\langle s : t, \varnothing, \varnothing, E, \varnothing, \varnothing \rangle$ then $s \leftrightarrow^*_{\mathrm{AC}} t$ and $s \simeq t \in E[N, p]$ for all $p \in E$. Hence delete applies in the step (6.10), and we have $E[N', p] = E[N, p] \setminus \{s \approx t\}$. For all $p \notin E$ an identity step is obtained.

– If normalize replaces $\langle s : t, R_0, R_1, E, C_0, C_1 \rangle$ by $\langle s : t, R_0, R_1, \varnothing, C_0, C_1 \rangle$ and $\langle s{\downarrow} : t{\downarrow}, \varnothing, \varnothing, E, \varnothing, \varnothing \rangle$ then for a process $p \in E$ we have $E[N', p] = (E[N, p] \setminus \{s \approx t\}) \cup \{s{\downarrow} \approx t{\downarrow}\}$. Therefore (6.10) is a valid normalize step in NKBtt. For all $p \notin E$ an identity step is obtained.

– Finally, suppose rewrite was used. For every process $p \notin (R_0 \cup R_1 \cup E) \cap R$ an identity step is obtained. Otherwise, the following possibilities are distinct due to the node condition.

i. If $p \in R_0 \cap R$ then there are rules $s \to t$ and $\ell \to r$ in $R[N, p]$ such that $t \to_{\ell \to r \setminus \mathcal{S}} u$. We have $R[N', p] = R[N, p] \setminus \{s \to t\} \cup \{s \to u\}$, so (6.10) is a valid compose step.

ii. If $p \in E \cap R$ there is an equation $s \simeq t$ in $E[N, p]$ and a rule $\ell \to r$ in $R[N, p]$ that admit a normalized rewrite step $t \to_{\ell \to r \setminus \mathcal{S}} u$. We have $E[N', p] = E[N, p] \setminus \{s \simeq t\} \cup \{s \simeq u\}$, so (6.10) is a simplify step.

iii. If $p \in R_1 \cap R$ there are rules $\ell \to r$ and $t \to s$ in $R[N, p]$ such that $t \to_{\ell \to r \setminus \mathcal{S}} u$. As $R[N', p] = R[N, p] \setminus \{t \to s\}$ and $E[N', p] = E[N, p] \cup \{u \approx s\}$ we obtain (6.10) by an application of collapse.

For any inference rule besides gc and subsume, the non-emptiness requirement for the set of affected labels ensures that (6.10) is a strict step for some $p' \in \mathcal{P}(N')$. □

Lemma 6.56. *Let (Θ, Ψ) be a normalizing pair such that $\Psi(u, v) = \{u \to v\}$. Assume for an* NKBtt *inference step $(\mathcal{E}, \mathcal{R}, \mathcal{C}) \vdash (\mathcal{E}', \mathcal{R}', \mathcal{C}')$ using (Θ, Ψ) there exist a node set N and a process $p \in \mathcal{P}(N)$ such that $\mathcal{E} = E[N, p]$, $\mathcal{R} = R[N, p]$ and $\mathcal{C} = C[N, p]$. Then there is an* MNKBtt *inference step $N \vdash N'$ with split set U and a process $p' \in \mathcal{P}(N')$ such that $p = pred_U(p')$, $\mathcal{E}' = E[N', p']$, $\mathcal{R}' = R[N', p']$ and $\mathcal{C}' = C[N', p']$.*

Proof. In the following case distinction on the applied NKBtt rule, $(*)$ refers to the proof obligations $\mathcal{E}' = E[N', p']$, $\mathcal{R}' = R[N', p']$, and $\mathcal{C}' = C[N', p']$.

– Suppose orient is applied to an equation $s \simeq t \in \mathcal{E}$, so $s{\downarrow} = s$ and $t{\downarrow} = t$. There must be a node $n = \langle s : t, R_0, R_1, E, C_0, C_1 \rangle$ in N such that $p \in E$ and $\mathcal{C} \cup \{s \to t\} \cup \mathcal{S}$ is AC terminating. If $\mathcal{C} \cup \{t \to s\} \cup \mathcal{S}$ is AC terminating as well, we set $U = \{p\}$. For $E_{lr} = E_{rl} = \{p\}$, $R_{lr} = \{p0\}$ and $R_{rl} = \{p1\}$, orient yields

$$\begin{aligned} N' = &split_{\{p\}}(N \setminus \{n\}) \cup N^{\{p0\}}_{\Theta(s,t)} \cup N^{\{p1\}}_{\Theta(t,s)} \cup \\ &\{\langle s : t, R_0 \cup \{p0\}, R_1 \cup \{p1\}, E \setminus \{p\}, C_0 \cup \{p0\}, C_1 \cup \{p1\} \rangle\} \end{aligned}$$

For $p' = p0$ we then have $p = pred_U(p')$ such that $(*)$ is satisfied. If on the other hand $\mathcal{C} \cup \{t \to s\} \cup \mathcal{S}$ is not AC terminating then orient can be applied with $U = \varnothing$, $R_{lr} = \{p\}$, and $R_{rl} = \varnothing$ to obtain

$$N' = (N \setminus \{n\}) \cup \{\langle s : t, R_0 \cup \{p\}, R_1, E \setminus \{p\}, C_0 \cup \{p\}, C_1 \rangle\} \cup N^{\{p\}}_{\Theta(s,t)}$$

such that $(*)$ holds for $p' = p$.

In all remaining cases we can set $p' = p$ since no process splitting occurs.

– In the case where deduce generates $s \approx t$ from an $\mathcal{L}$-overlap involving rules $\ell_1 \to r_1$ and $\ell_2 \to r_2$, there are nodes $\langle \ell_1 : r_1, R, \dots \rangle$ and $\langle \ell_2 : r_2, R', \dots \rangle$ in N such that $p \in R \cap R'$. A deduce step in MNKBtt yields

$$N' = N \cup \{\langle s : t, \varnothing, \varnothing, R \cap R', \varnothing, \varnothing \rangle\}$$

which satisfies $(*)$ as well.

– If delete removes $s \approx t \in \mathcal{E}$ because $s \leftrightarrow^*_{\mathrm{AC}} t$ then N must contain a node $n = \langle s : t, R_0, R_1, E, C_0, C_1 \rangle$ such that $p \in E$. Since $s \approx t$ cannot be oriented into an AC-terminating rule, by Lemma 6.52 the labels R_0, R_1, C_0 and C_1 must be empty. Thus n can be removed by delete in MNKBtt.

– Suppose normalize replaces $s \approx t \in \mathcal{E}$ by $s{\downarrow} \approx t{\downarrow} \in \mathcal{E}'$. We must have $n = \langle s : t, R_0, R_1, E, C_0, C_1 \rangle \in N$ and the following normalize step in MNKBtt satisfies $(*)$:

$$N' = (N \setminus \{n\}) \cup \{\langle s : t, R_0, R_1, \varnothing, C_0, C_1 \rangle\} \cup \{\langle s{\downarrow} : t{\downarrow}, \varnothing, \varnothing, E, \varnothing, \varnothing \rangle\}$$

– If compose replaces $s \to t$ by $s \to u$ as $t \to_{\ell \to r \setminus \mathcal{S}} u$ then N contains nodes $n = \langle s : t, R_0, R_1, E, C_0, C_1 \rangle$ and $\langle \ell : r, R, \dots \rangle$ such that $p \in R_0 \cap R$. Thus rewrite applies and $(*)$ is satisfied as we obtain

$$N' = (N \setminus \{n\}) \cup \{\langle s : t, R_0 \setminus R, R_1 \setminus R, E \setminus R, C_0, C_1 \rangle\} \\ \cup \{\langle s : u, R_0 \cap R, \varnothing, (E \cup R_1) \cap R, \varnothing, \varnothing \rangle\}$$

– Suppose simplify replaces an equation $s \simeq t$ by $s \simeq u$ as $t \to_{\ell \to r \setminus \mathcal{S}} u$. Then N contains nodes $n = \langle s : t, R_0, R_1, E, C_0, C_1 \rangle$ and $\langle \ell : r, R, \dots \rangle$ such that $p \in E \cap R$. An application of rewrite admits a step

$$N' = (N \setminus \{n\}) \cup \{\langle s : t, R_0 \setminus R, R_1 \setminus R, E \setminus R, C_0, C_1 \rangle\} \\ \cup \{\langle s : u, R_0 \cap R, \varnothing, (E \cup R_1) \cap R, \varnothing, \varnothing \rangle\}$$

Since $p \in E \cap R$, $(*)$ holds.

– Finally, assume collapse is applied to turn a rule $t \to s$ into an equation $u \approx s$ because of a normalized rewrite step $t \to_{\ell \to r \setminus \mathcal{S}} u$. Then N contains nodes $n = \langle s : t, R_0, R_1, E, C_0, C_1 \rangle$ and $\langle \ell : r, R, \dots \rangle$ such that $p \in R_1 \cap R$. To satisfy $(*)$ we apply rewrite to obtain

$$N' = (N \setminus \{n\}) \cup \{\langle s : t, R_0 \setminus R, R_1 \setminus R, E \setminus R, C_0, C_1 \rangle\} \\ \cup \{\langle s : u, R_0 \cap R, \varnothing, (E \cup R_1) \cap R, \varnothing, \varnothing \rangle\}$$ □

We define the sequence $p_0, \dots, p_k$ of *ancestors* of a process p in an MNKBtt run γ as in Definition 4.30. As MNKBtt steps are reflected in NKBtt steps by Lemma 6.55 we obtain the following simulation result:

Corollary 6.57. *Suppose an* MNKBtt *run* $\gamma\colon N_0 \vdash N_1 \vdash \cdots \vdash N_k$ *gives rise to a process* $p \in \mathcal{P}(N_k)$ *having ancestors* $p_0, \dots, p_k$. *Let* $\mathcal{E}_i$, $\mathcal{R}_i$ *and* $\mathcal{C}_i$ *denote* $E[N_i, p_i]$, $R[N_i, p_i]$ *and* $C[N_i, p_i]$, *respectively. Then the sequence*

$$\gamma_p\colon (\mathcal{E}_0, \mathcal{R}_0, \mathcal{C}_0) \vdash^{=} (\mathcal{E}_1, \mathcal{R}_1, \mathcal{C}_1) \vdash^{=} \cdots \vdash^{=} (\mathcal{E}_k, \mathcal{R}_k, \mathcal{C}_k)$$

is a valid NKBtt *run, called the* projection *of* γ *to* p. □

Using projections, the notions of success, failure and fairness given for NKBtt can be naturally extended to MNKBtt in exactly the same way as MKBtt inherits them from KBtt (see Definition 4.32). It is then straightforward to establish a correctness result for MNKBtt.

Correctness Theorem 6.58. *Let* $N_{\mathcal{E}}$ *be the initial node set for a set of equations* $\mathcal{E}$ *and let* $\gamma\colon N_{\mathcal{E}} \vdash^* N$ *be a finite* MNKBtt *run which is nonfailing and fair for some* $p \in \mathcal{P}(N)$. *Then* $R[N, p]$ *is* $\mathcal{S}$*-convergent for* $\mathcal{E}$.

Proof. By Corollary 6.57 there is a corresponding fair and nonfailing NKBtt run $\gamma_p : (\mathcal{E}, \varnothing, \varnothing) \vdash^* (\varnothing, R[N, p], C[N, p])$. Since γ_p is finite, $R[N, p]$ is $\mathcal{S}$-convergent for $\mathcal{E}$ according to Theorem 6.50. □

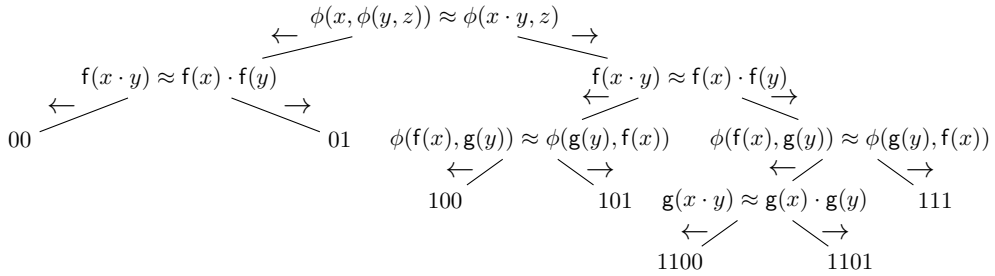


Figure 6.7: Part of the process tree developed in a run on CGA where process 1101 succeeds.

Example 6.59. We consider the system CGA describing an Abelian group with a group action ϕ on itself such that two endomorphisms f and g commute with respect to ϕ, as described by the set of equations $\mathcal{E}$:

$$\begin{aligned} \phi(\mathsf{e}, x) &\approx x & \phi(x, \phi(y, z)) &\approx \phi(x \cdot y, z) & \phi(\mathsf{f}(x), \mathsf{g}(y)) &\approx \phi(\mathsf{g}(y), \mathsf{f}(x)) \\ \mathsf{f}(\mathsf{e}) &\approx \mathsf{e} & \mathsf{f}(x \cdot y) &\approx \mathsf{f}(x) \cdot \mathsf{f}(y) \\ \mathsf{g}(\mathsf{e}) &\approx \mathsf{e} & \mathsf{g}(x \cdot y) &\approx \mathsf{g}(x) \cdot \mathsf{g}(y) \end{aligned}$$

together with $\mathcal{T}$ axiomatizing Abelian group theory:

$$x \cdot y \approx y \cdot x \qquad (x \cdot y) \cdot z \approx x \cdot (y \cdot z) \qquad x \cdot \mathsf{e} \approx x \qquad x \cdot x^{-1} \approx \mathsf{e}$$

which is representable by the AC-convergent system $\mathcal{S}_{\mathrm{G}}$ given in Example 6.17. Several equations in $\mathcal{E}$ are orientable in both directions. An NKBtt run thus often has to decide for one orientation, excluding many other orientation possibilities. In contrast, an MNKBtt run can keep track of all possibilities. In the course of a run it creates a tree of processes where each branch corresponds to a sequence of orientation decisions. Part of the process tree developed in a run on CGA is shown in Figure 6.7, where process 1101 succeeds with the following $\mathcal{S}$-convergent system:

$$\begin{aligned} \mathsf{f}(\mathsf{e}) &\to \mathsf{e} & \mathsf{f}(x \cdot y) &\to \mathsf{f}(x) \cdot \mathsf{f}(y) & \mathsf{i}(\mathsf{f}(x)) &\to \mathsf{f}(\mathsf{i}(x)) \\ \mathsf{g}(\mathsf{e}) &\to \mathsf{e} & \mathsf{g}(x \cdot y) &\to \mathsf{g}(x) \cdot \mathsf{g}(y) & \mathsf{i}(\mathsf{g}(x)) &\to \mathsf{g}(\mathsf{i}(x)) \\ \phi(\mathsf{e}, x) &\to x & \phi(\mathsf{f}(x), \mathsf{e}) &\to \mathsf{f}(x) & \phi(x, \mathsf{f}(y)) &\to \phi(\mathsf{f}(y) \cdot x, \mathsf{e}) \\ \phi(x, \phi(y, z)) &\to \phi(x \cdot y, z) & \phi(\mathsf{g}(x), \mathsf{e}) &\to \mathsf{g}(x) & \phi(x, \mathsf{g}(y)) &\to \phi(\mathsf{g}(y) \cdot x, \mathsf{e}) \end{aligned}$$

Normalized completion could naturally also be performed with respect to different theories, for instance ACU or group theory extended with equations axiomatizing one or both of the endomorphisms f and g, or the group action ϕ. Note that the use of termination tools is beneficial here as the equation $\phi(\mathsf{f}(x), \mathsf{g}(y)) \approx \phi(\mathsf{g}(y), \mathsf{f}(x))$ cannot be oriented with AC-RPO or AC-KBO.

Example 6.60. Consider ring theory with two commuting multiplicative mappings f and g as defined by AC axioms for $+$ together with the set of equations

$$\begin{array}{rcl} x + \mathsf{0} \approx x & \mathsf{f}(\mathsf{1}) \approx \mathsf{1} & x \cdot (y + z) \approx (x \cdot y) + (x \cdot z) \\ x + (-x) \approx \mathsf{0} & \mathsf{g}(\mathsf{1}) \approx \mathsf{1} & (x + y) \cdot z \approx (x \cdot z) + (x \cdot z) \\ \mathsf{1} \cdot x \approx x & \mathsf{f}(x \cdot y) \approx \mathsf{f}(x) \cdot \mathsf{f}(y) & (x \cdot y) \cdot z \approx x \cdot (y \cdot z) \\ x \cdot \mathsf{1} \approx x & \mathsf{g}(x \cdot y) \approx \mathsf{g}(x) \cdot \mathsf{g}(y) & \mathsf{f}(x) \cdot \mathsf{g}(y) \approx \mathsf{g}(y) \cdot \mathsf{f}(x) \end{array}$$

Since the equation $\mathsf{f}(x) \cdot \mathsf{g}(y) \approx \mathsf{g}(y) \cdot \mathsf{f}(x)$ cannot be oriented with AC-RPO or AC-KBO, completion tools relying on such standard orders fail. But when normalized completion with termination tools is performed modulo group theory $\mathcal{S}_{\mathrm{G}}$, we obtain the following $\mathcal{S}_{\mathrm{G}}$-convergent TRS:

$$\begin{array}{rcl} \mathsf{1} \cdot x \to x & x \cdot (y + z) \to (x \cdot y) + (x \cdot z) & \mathsf{f}(\mathsf{1}) \to \mathsf{1} \\ x \cdot \mathsf{1} \to x & (x + y) \cdot z \to (x \cdot z) + (y \cdot z) & \mathsf{g}(\mathsf{1}) \to \mathsf{1} \\ \mathsf{0} \cdot x \to \mathsf{0} & (-x) \cdot y \to -(x \cdot y) & \mathsf{f}(x \cdot y) \to \mathsf{f}(x) \cdot \mathsf{f}(y) \\ x \cdot \mathsf{0} \to \mathsf{0} & x \cdot (-y) \to -(x \cdot y) & \mathsf{g}(x \cdot y) \to \mathsf{g}(x) \cdot \mathsf{g}(y) \\ (x \cdot y) \cdot z \to x \cdot (y \cdot z) & \mathsf{f}(x) \cdot \mathsf{g}(y) \to \mathsf{g}(y) \cdot \mathsf{f}(x) & \end{array}$$

Alternatively, normalized completion can for instance be performed modulo ring theory or ring theory with endomorphisms.

Chapter 7

Constrained Equalities in Completion-like Procedures

With *maximal completion*, Klein and Hirokawa [56] proposed to formulate Knuth-Bendix completion as a maximal termination problem. Their implementation via encodings in (maximal) satisfiability problems turned out to be surprisingly efficient, though limited to the classical reduction orders LPO and KBO. This raises the question whether such an approach could also be beneficial for variants of completion such as ordered completion and completion modulo theories, and inductive theorem proving techniques like rewriting induction. These equational deduction methods share three crucial characteristics with standard completion: (i) they search for a rewrite system that satisfies a joinability requirement imposed by input equalities, (ii) they work saturation-based, and (iii) their search critically depends on a user-supplied reduction order.

But some characteristics of maximal completion become problematic in the settings of the above-mentioned completion-like procedures:

(1) When exploring multiple branches of the search tree, all critical pairs originating from some orientation are considered by all branches (in contrast to multi-completion). Maintaining this large pool of equations turned out to be a bottleneck in the implementation, and is even more critical for ordered completion due to a larger set of (extended) critical pairs.

(2) This problem is compounded as maximal completion does not feature inter-reduction, which is decisive for efficiency in completion-like procedures.

(3) Maximal completion always aims at orienting a maximal number of equations in the current equation pool. However, this *maximality principle* is not suitable for settings such as rewriting induction (cf. Example 7.35).

We will hence investigate a generalized version of maximal completion based on *constrained equalities*. Our framework allows to describe the above-mentioned completion-like procedures in an abstract but uniform way by saturating a set of constrained equalities $\mathcal{C}$ with respect to a suitable *transformer*. The procedure succeeds as soon as a TRS is found which satisfies certain constraints induced by $\mathcal{C}$. This abstract setting allows for simple correctness proofs, and translating the constraint satisfiability problem into some suitable logic admits a straightforward but competitive implementation. In such an implementation, the use of constrained equalities allows for more control over joinability requirements

and facilitates inter-reduction. Compared to maximal completion, this avoids the above mentioned drawbacks (1)—(3). On the other hand our method can still perform pure maximal completion by adjusting parameters appropriately (see Section 8.2). For standard completion and rewriting induction this approach was already described in [55]. We here additionally account on ordered completion and normalized completion.

Section 7.1 introduces constrained equalities. In Section 7.2 we instantiate the framework to standard completion, in Section 7.3 to ordered completion, and in Section 7.4 to normalized completion. Finally, in Section 7.5 we describe how the approach can be used for rewriting induction.

7.1 Constrained Equalities

The following definition fixes our notion of constraints and the data structure of constrained equalities, which are central to our framework.

Definition 7.1. Let a *termination constraint* C be defined as follows:

$$C ::= \ell \to r \mid \top \mid \bot \mid \neg C \mid C \vee C \mid C \wedge C$$

Given a TRS $\mathcal{R}$, the relation $\mathcal{R} \models C$ is inductively defined such that $\mathcal{R} \models \ell \to r$ if and only if $\ell \to r \in \mathcal{R}$, and all other operators have their standard semantics as boolean connectives. If $\mathcal{R} \models C$ the constraint C is said to be *satisfied* by $\mathcal{R}$. A *constrained equality* $\langle s \approx t, C\rangle$ consists of an equation $s \approx t$ and a termination constraint C. A *constrained equational system* (CES) $\mathcal{C}$ is a set of constrained equalities.

Example 7.2. Consider the following set of equations $\mathcal{E}$:

$$\mathsf{a}(x,y) \approx \mathsf{b}(x) \tag{1}$$
$$\mathsf{c}(y) \approx \mathsf{a}(x,y) \tag{2}$$
$$\mathsf{f}(\mathsf{b}(x)) \approx \mathsf{b}(x) \tag{3}$$
$$\mathsf{f}(\mathsf{a}(x,y)) \approx \mathsf{d} \tag{4}$$

and let the example CES $\mathcal{C}$ consist of the following constrained equalities:

$$\langle \mathsf{c}(y) \approx \mathsf{b}(x), \mathsf{a}(x,y) \to \mathsf{b}(x) \wedge \mathsf{f}(\mathsf{b}(x)) \to \mathsf{b}(x)\rangle$$
$$\langle \mathsf{f}(\mathsf{a}(x,y)) \approx \mathsf{d}, \neg(\mathsf{a}(x,y) \to \mathsf{b}(x) \vee \mathsf{a}(x,y) \to \mathsf{c}(y))\rangle$$

To denote constraints in a more succinct way, we will for an equation numbered (n) denote the rule obtained by orienting it from left to right by (n) and the reversed rule by (n'). Hence $\mathcal{C}$ is written as follows:

$$\langle \mathsf{c}(y) \approx \mathsf{b}(x), (1) \wedge (3)\rangle \qquad \langle \mathsf{f}(\mathsf{a}(x,y)) \approx \mathsf{d}, \neg((1) \vee (2'))\rangle$$

The following notations are frequently used throughout this chapter.

$$\begin{aligned}
\mathcal{E}^{\top} &= \{\langle s \approx t, \top\rangle \mid s \approx t \in \mathcal{E}\} \\
\mathcal{C}{\downarrow_{\mathcal{R}}} &= \left\{\left\langle s{\downarrow_{\mathcal{R}}} \approx t{\downarrow_{\mathcal{R}}}, C \wedge \bigwedge \mathcal{R}\right\rangle \mid \langle s \approx t, C\rangle \in \mathcal{C} \text{ and } s{\downarrow_{\mathcal{R}}} \neq t{\downarrow_{\mathcal{R}}}\right\} \\
\mathcal{C} \ominus \mathcal{R} &= \left\{\left\langle s \approx t, C \wedge \neg \bigwedge \mathcal{R}\right\rangle \mid \left\langle s \approx t, C\right\rangle \in \mathcal{C}\right\} \\
\mathcal{C}[\![\mathcal{R}]\!] &= \{s \approx t \mid \left\langle s \approx t, C\right\rangle \in \mathcal{C} \text{ and } \mathcal{R} \models C\}
\end{aligned}$$

The set $\mathcal{E}^{\top}$ denotes the CES corresponding to a set of equalities with no constraints attached. By $\mathcal{C}{\downarrow_{\mathcal{R}}}$ we denote the result of normalizing all equations in $\mathcal{C}$ with respect to a TRS $\mathcal{R}$, keeping only those which are not joinable, and adding $\mathcal{R}$ as constraint. The CES $\mathcal{C} \ominus \mathcal{R}$ contains all constrained equalities in $\mathcal{C}$ where constraints are extended with the negation of $\mathcal{R}$. The *projection* $\mathcal{C}[\![\mathcal{R}]\!]$ of $\mathcal{C}$ to $\mathcal{R}$ is the set of all equations whose constraints in $\mathcal{C}$ are satisfied by $\mathcal{R}$.

Example 7.3. Let $\mathcal{E}$ be the set of equations from Example 7.2. Then $\mathcal{E}^{\top}$ is

$$\begin{array}{ll}
\langle \mathsf{a}(x,y) \approx \mathsf{b}(x), \top\rangle & \langle \mathsf{c}(y) \approx \mathsf{a}(x,y), \top\rangle \\
\langle \mathsf{f}(\mathsf{b}(x)) \approx \mathsf{b}(x), \top\rangle & \langle \mathsf{f}(\mathsf{a}(x,y)) \approx \mathsf{d}, \top\rangle
\end{array}$$

Consider the following TRS $\mathcal{R}$:

$$\mathsf{a}(x,y) \to \mathsf{b}(x) \qquad\qquad \mathsf{f}(\mathsf{b}(x)) \to \mathsf{b}(x)$$

Then $\mathcal{C} = (\mathcal{E}^{\top}){\downarrow_{\mathcal{R}}} = \{\langle \mathsf{c}(y) \approx \mathsf{b}(x), R\rangle, \langle \mathsf{f}(\mathsf{b}(x)) \approx \mathsf{d}, R\rangle\}$, where $R = (1) \wedge (3)$. Note that $(\mathcal{E}^{\top}){\downarrow_{\mathcal{R}'}}$ would not be unique for, e.g., $\mathcal{R}' = \mathcal{R} \cup \{\mathsf{a}(x,y) \to \mathsf{c}(y)\}$. The set CES $\mathcal{E}^{\top} \ominus \mathcal{R}$ consists of the constrained equalities

$$\begin{array}{ll}
\langle \mathsf{a}(x,y) \approx \mathsf{b}(x), \neg R\rangle & \langle \mathsf{c}(y) \approx \mathsf{a}(x,y), \neg R\rangle \\
\langle \mathsf{f}(\mathsf{b}(x)) \approx \mathsf{b}(x), \neg R\rangle & \langle \mathsf{f}(\mathsf{a}(x,y)) \approx \mathsf{d}, \neg R\rangle
\end{array}$$

whereas $\mathcal{C} \ominus \mathcal{R} = \{\langle \mathsf{c}(y) \approx \mathsf{b}(x), \bot\rangle, \langle \mathsf{f}(\mathsf{b}(x)) \approx \mathsf{d}, \bot\rangle\}$ (after constraint simplification). We have $\mathcal{E}^{\top}[\![\mathcal{R}]\!] = \mathcal{E}$, $\mathcal{C}[\![\mathcal{R}]\!] = \{\mathsf{c}(y) \approx \mathsf{b}(x), \mathsf{f}(\mathsf{b}(x)) \approx \mathsf{d}\}$, and $(\mathcal{E}^{\top} \ominus \mathcal{R})[\![\mathcal{R}]\!] = \varnothing$.

The proofs of the following properties are straightforward by unfolding the above definitions.

Lemma 7.4. *If* $\mathcal{E}^{\top}{\downarrow_{\mathcal{R}}} = \varnothing$ *then* $\mathcal{E} \subseteq {\downarrow_{\mathcal{R}}}$. □

Lemma 7.5. $(\mathcal{C} \ominus \mathcal{R})[\![\mathcal{R}]\!] = \varnothing$. □

An ES $\mathcal{E}$ is said to be *ground joinable by* $\mathcal{R}$ if $s\sigma \downarrow_{\mathcal{R}} t\sigma$ for every $s \approx t \in \mathcal{E}$ and substitution σ that is grounding for all terms in $\mathcal{E}$. The concept of a transformer is central to our framework:[1]

Definition 7.6. A function S mapping CESs to CESs is a *(ground) transformer* if for any CES $\mathcal{C}$ and TRS $\mathcal{R}$ the set of equations $\mathcal{C}[\![\mathcal{R}]\!]$ is (ground) joinable by $\mathcal{R}$ whenever $S(\mathcal{C})[\![\mathcal{R}]\!]$ is (ground) joinable by $\mathcal{R}$.

1 A (ground) transformer was called a *(ground) reduction* in [55].

The following sufficient criterion will in the sequel be used to verify that a function constitutes a (ground) transformer.

Lemma 7.7. *A function S is a transformer if for any CES $\mathcal{C}$ and TRS $\mathcal{R}$*

$$\mathcal{C}[\![\mathcal{R}]\!] \subseteq \xrightarrow[\mathcal{R}]{*} \cdot \xleftrightarrow[S(\mathcal{C})[\![\mathcal{R}]\!]]{=} \cdot \xleftarrow[\mathcal{R}]{*} \tag{7.1}$$

and a ground transformer if the inclusion (7.1) *holds restricted to ground terms.*

Proof. Suppose $S(\mathcal{C})[\![\mathcal{R}]\!]$ is joinable by $\mathcal{R}$. By (7.1), for any $s \approx t \in \mathcal{C}[\![\mathcal{R}]\!]$ there are terms u and v with $s \to^*_\mathcal{R} u \leftrightarrow^=_{S(\mathcal{C})[\![\mathcal{R}]\!]} v \;{}^*_\mathcal{R}\!\leftarrow t$. As $u \downarrow_\mathcal{R} v$ by assumption, also $s \downarrow_\mathcal{R} t$ holds.

If $S(\mathcal{C})[\![\mathcal{R}]\!]$ is ground joinable by $\mathcal{R}$ then by (7.1), for any $s \approx t \in \mathcal{C}[\![\mathcal{R}]\!]$ and grounding substitution σ, some u and v satisfy $s\sigma \to^*_\mathcal{R} u \leftrightarrow^=_{S(\mathcal{C})[\![\mathcal{R}]\!]} v \;{}^*_\mathcal{R}\!\leftarrow t\sigma$. As u and v are ground-joinable by assumption this also holds for $s\sigma$ and $t\sigma$. □

Next we define a particular pattern for functions from CESs to CESs which is parameterized by two functions F and $\mathfrak{R}$. It will turn out that all functions matching this pattern constitute transformers. Actually all transformers considered in the sequel will fit this shape, only F and $\mathfrak{R}$ are instantiated appropriately.

Definition 7.8. Let $\mathcal{C}$ be a CES, $\mathfrak{R}$ a function mapping a CES to a set of terminating TRSs, and F a function mapping a TRS to an ES. Then $S_\mathcal{R}$ and S are defined as follows:

$$S_\mathcal{R}(\mathcal{C}) = (\mathcal{C} \ominus \mathcal{R}) \;\cup\; \mathcal{C}{\downarrow_\mathcal{R}} \;\cup\; F(\mathcal{R})^\top{\downarrow_\mathcal{R}}$$
$$S(\mathcal{C}) = \bigcup_{\mathcal{R} \in \mathfrak{R}(\mathcal{C})} S_\mathcal{R}(\mathcal{C})$$

Lemma 7.9. *S is a (ground) transformer.*

Proof. Let $\mathcal{H}$ be an arbitrary TRS and $s \approx t \in \mathcal{C}[\![\mathcal{H}]\!]$. According to Lemma 7.7 and the definition of S it is sufficient to show that $s\sigma \to^*_\mathcal{H} \cdot \leftrightarrow^=_{S_\mathcal{R}(\mathcal{C})[\![\mathcal{H}]\!]} \cdot \;{}^*_\mathcal{H}\!\leftarrow t\sigma$ for all $\mathcal{R} \in \mathfrak{R}(\mathcal{C})$. As $s \approx t \in \mathcal{C}[\![\mathcal{H}]\!]$ there must be some $\langle s \approx t, C\rangle \in \mathcal{C}$ such that $\mathcal{H} \models C$. We distinguish three cases.

- If $\mathcal{R} \subseteq \mathcal{H}$ and $s \downarrow_\mathcal{R} t$ then $s\sigma \downarrow_\mathcal{R} t\sigma$ holds for all (ground) substitutions σ, and thus $s\sigma \downarrow_\mathcal{H} t\sigma$.
- If $\mathcal{R} \subseteq \mathcal{H}$ but $s \not\downarrow_\mathcal{R} t$ then $\mathcal{H} \models C \wedge \bigwedge \mathcal{R}$, and $\langle s{\downarrow_\mathcal{R}} \approx t{\downarrow_\mathcal{R}}, C \wedge \bigwedge \mathcal{R}\rangle \in \mathcal{C}{\downarrow_\mathcal{R}}$. Thus $s{\downarrow_\mathcal{R}} \approx t{\downarrow_\mathcal{R}} \in S_\mathcal{R}(\mathcal{C})[\![\mathcal{H}]\!]$. Then for all (ground) substitutions σ of s and t we have $s\sigma \to^*_\mathcal{H} s{\downarrow_\mathcal{R}}\sigma \leftrightarrow_{S_\mathcal{R}(\mathcal{C})[\![\mathcal{H}]\!]} t{\downarrow_\mathcal{R}}\sigma \;{}^*_\mathcal{H}\!\leftarrow t\sigma$.
- If $\mathcal{R} \not\subseteq \mathcal{H}$ then $\mathcal{H} \models C \wedge \neg \bigwedge \mathcal{R}$, and $\langle s \approx t, C \wedge \neg \bigwedge \mathcal{R}\rangle \in \mathcal{C} \ominus \mathcal{R}$. Thus we have $s \approx t \in S_\mathcal{R}(\mathcal{C})[\![\mathcal{H}]\!]$. Clearly $s\sigma \leftrightarrow_{S_\mathcal{R}(\mathcal{C})[\![\mathcal{H}]\!]} t\sigma$ holds for all (ground) substitutions σ of s and t. □

In the sequel we will refer to transformers S according to Definition 7.8 as *standard transformers*. From the definition of a (ground) transformer and Lemma 7.9 it is clear that the composition of a transformer with itself is again a transformer.

Corollary 7.10. *If S is a (ground) transformer then S^n is a (ground) transformer for all $n \geqslant 0$.* □

We present a straightforward observation before turning to a concrete application of our framework.

Lemma 7.11. *If $S(\mathcal{C})[\![\mathcal{R}]\!] = \varnothing$ then $F(\mathcal{R}) \subseteq \downarrow_{\mathcal{R}}$ for all $\mathcal{R} \in \mathfrak{R}(\mathcal{C})$.*

Proof. If $S(\mathcal{C})[\![\mathcal{R}]\!] = \varnothing$ then $S_{\mathcal{R}}(\mathcal{C})[\![\mathcal{R}]\!] = \varnothing$ for all $\mathcal{R} \in \mathfrak{R}(\mathcal{C})$. Let $\mathcal{R} \in \mathfrak{R}(\mathcal{C})$ and $s \approx t \in F(\mathcal{R})$, so $\langle s{\downarrow} \approx t{\downarrow}, \bigwedge \mathcal{R}\rangle \in F(\mathcal{R})^{\top}{\downarrow_{\mathcal{R}}}$. As $F(\mathcal{R})^{\top}{\downarrow_{\mathcal{R}}} \subseteq S_{\mathcal{R}}(\mathcal{C})$ and $S_{\mathcal{R}}(\mathcal{C})[\![\mathcal{R}]\!] = \varnothing$ we must have $\{\langle s{\downarrow_{\mathcal{R}}} \approx t{\downarrow_{\mathcal{R}}}, \bigwedge \mathcal{R}\rangle\}[\![\mathcal{R}]\!] = \varnothing$. Since $\mathcal{R} \models \bigwedge \mathcal{R}$ this implies $s{\downarrow_{\mathcal{R}}} = t{\downarrow_{\mathcal{R}}}$, so $s \downarrow_{\mathcal{R}} t$. □

7.2 Standard Completion

We will now instantiate the standard transformer S from Definition 7.8 to a transformer S_{KB} that allows us to formulate a completion procedure in the constrained equality framework.

Definition 7.12. Given an ES $\mathcal{E}$, let S_{KB} be the standard transformer such that for all $\mathcal{R}$ in $\mathfrak{R}(\mathcal{C})$ we have $\mathcal{R} \subseteq \leftrightarrow^*_{\mathcal{E}}$ and $F(\mathcal{R}) = \mathrm{CP}(\mathcal{R})$, for any CES $\mathcal{C}$.

Saturating a CES with respect to the transformer S_{KB} constitutes a completion procedure, as was already shown in [55]:

Theorem 7.13. *Let $\mathcal{C} = S^n_{\mathrm{KB}}(\mathcal{E}^{\top})$. If $\mathcal{C}[\![\mathcal{R}]\!] = \varnothing$ for some $\mathcal{R} \in \mathfrak{R}(\mathcal{C})$ then $\mathcal{R}$ is convergent for $\mathcal{E}$.*

Proof. Let $\mathcal{R} \in \mathfrak{R}(\mathcal{C})$ and $\mathcal{C}[\![\mathcal{R}]\!] = \varnothing$. According to Corollary 7.10 S^n_{KB} is a transformer, hence $\mathcal{E} \subseteq \downarrow_{\mathcal{R}}$. Since $\mathcal{R} \subseteq \leftrightarrow^*_{\mathcal{E}}$ by assumption we obtain $\leftrightarrow^*_{\mathcal{R}} = \leftrightarrow^*_{\mathcal{E}}$. We now distinguish two cases. If $n = 0$ then $\mathcal{E}$ must be empty. Since termination of $\mathcal{R}$ and the inclusion $\mathcal{R} \subseteq \leftrightarrow^*_{\mathcal{E}}$ entail emptiness of $\mathcal{R}$, the claim trivially holds. If $n > 0$, it is not difficult to see that $\mathcal{C}[\![\mathcal{R}]\!] = \varnothing$ implies $S_{\mathrm{KB}}(\mathcal{C})[\![\mathcal{R}]\!] = \varnothing$. Thus by Lemma 7.11 we have $\mathrm{CP}(\mathcal{R}) \subseteq \downarrow_{\mathcal{R}}$ which implies confluence of $\mathcal{R}$. Since $\mathcal{R}$ is terminating by construction and $\leftrightarrow^*_{\mathcal{R}} = \leftrightarrow^*_{\mathcal{E}}$, $\mathcal{R}$ is convergent for $\mathcal{E}$. □

The choice of suitable TRSs $\mathcal{R}$ is obviously a challenge in an implementation. This issue is addressed in Chapter 8. We now illustrate our procedure by means of an example.

Example 7.14. Consider the ES $\mathcal{E}$ from Example 3.18:

$$\mathsf{a}(x, y) \approx \mathsf{b}(x) \tag{1}$$
$$\mathsf{a}(x, y) \approx \mathsf{c}(y) \tag{2}$$
$$\mathsf{f}(\mathsf{b}(x)) \approx \mathsf{b}(x) \tag{3}$$
$$\mathsf{f}(\mathsf{a}(x, y)) \approx \mathsf{d} \tag{4}$$

In the initial CES $\mathcal{C}_0 = \mathcal{E}^\top = \{\langle(1), \top\rangle, \langle(2), \top\rangle, \langle(3), \top\rangle, \langle(4), \top\rangle\}$ we can orient all equations by choosing $\mathcal{R}_0 = \{(1), (2), (3), (4)\}$. Let us abbreviate $R_0 = (1) \wedge (2) \wedge (3) \wedge (4)$. We then have

$$\begin{aligned}
\mathcal{C}_0 \ominus \mathcal{R}_0 &= \{\langle(1), \neg R_0\rangle, \langle(2), \neg R_0\rangle, \langle(3), \neg R_0\rangle, \langle(4), \neg R_0\rangle\} \\
\mathcal{C}_0{\downarrow_{\mathcal{R}_0}} &= \varnothing \\
F(\mathcal{R}_0)^\top{\downarrow_{\mathcal{R}_0}} &= \{\langle(5), \top\rangle, \langle(6), \top\rangle, \langle(7), \top\rangle\}{\downarrow_{\mathcal{R}_0}} \\
&= \{\langle(5), R_0\rangle, \langle(7), R_0\rangle, \langle(8), R_0\rangle\}
\end{aligned}$$

where (5), (6), and (7) stem from the overlaps $\langle(1), \epsilon, (2)\rangle$, $\langle(1), 1, (4)\rangle$, and $\langle(2), 1, (4)\rangle$, and (8) is the result of $\mathcal{R}_0$-normalizing (6):

$$\mathsf{b}(x) \approx \mathsf{c}(y) \tag{5}$$
$$\mathsf{f}(\mathsf{b}(x)) \approx \mathsf{d} \tag{6}$$
$$\mathsf{f}(\mathsf{c}(x)) \approx \mathsf{d} \tag{7}$$
$$\mathsf{b}(x) \approx \mathsf{d} \tag{8}$$

For $\mathfrak{R}(\mathcal{C}_0) = \{\mathcal{R}_0\}$ we therefore obtain the following CES $\mathcal{C}_1 = S_{\mathrm{KB}}(\mathcal{C}_0)$:

$$\mathcal{C}_1 = \{\langle(1), \neg R_0\rangle, \langle(2), \neg R_0\rangle, \langle(3), \neg R_0\rangle, \langle(4), \neg R_0\rangle, \langle(5), R_0\rangle, \langle(7), R_0\rangle, \langle(8), R_0\rangle\}$$

Let $\mathcal{R}_1 = \mathcal{R}_0 \cup \{(7), (8)\} \in \mathfrak{R}(\mathcal{C}_1)$ and $R_1 = R_0 \wedge (7) \wedge (8)$, so $\neg R_0 \wedge \neg R_1$ can be simplified to $\neg R_0$. This yields

$$\begin{aligned}
\mathcal{C}_1 \ominus \mathcal{R}_1 &= \{\langle(1), \neg R_0\rangle, \langle(2), \neg R_0\rangle, \langle(3), \neg R_0\rangle, \langle(4), \neg R_0\rangle\} \\
&\quad \cup \{\langle(5), R_0 \wedge \neg R_1\rangle, \langle(6), R_0 \wedge \neg R_1\rangle, \langle(7), R_0 \wedge \neg R_1\rangle\} \\
\mathcal{C}_1{\downarrow_{\mathcal{R}_1}} &= \{\langle(9), R_0 \wedge R_1\rangle\} \\
F(\mathcal{R}_1)^\top{\downarrow_{\mathcal{R}_1}} &= \{\langle(9), R_1\rangle\}
\end{aligned}$$

where the additional equation (9) is the result of rewriting (5):

$$\mathsf{d} \approx \mathsf{c}(x) \tag{9}$$

Note that $\mathcal{C}_1{\downarrow_{\mathcal{R}_1}}$ and $F(\mathcal{R}_1)^\top{\downarrow_{\mathcal{R}_1}}$ can be merged. For $\mathfrak{R}(\mathcal{C}_1) = \{\mathcal{R}_1\}$ we get the following CES $\mathcal{C}_2 = S_{\mathrm{KB}}(\mathcal{C}_1)$:

$$\begin{aligned}
\mathcal{C}_2 = \{&\langle(1), \neg R_0\rangle, \langle(2), \neg R_0\rangle, \langle(3), \neg R_0\rangle, \langle(4), \neg R_0\rangle, \langle(5), R_0 \wedge \neg R_1\rangle\} \\
\cup \{&\langle(6), R_0 \wedge \neg R_1\rangle, \langle(7), R_0 \wedge \neg R_1\rangle, \langle(9), R_1\rangle\}
\end{aligned}$$

For $\mathfrak{R}(\mathcal{C}_2) = \{\mathcal{R}_2\}$ where $\mathcal{R}_2 = \mathcal{R}_1 \cup \{(9')\}$ we now have $S_{\mathrm{KB}}(\mathcal{C}_2)[\![\mathcal{R}_2]\!] = \varnothing$, so $\mathcal{R}_2$ is convergent for $\mathcal{E}$. This would also hold for the smaller TRS $\mathcal{R}_2' = \{(1), (6), (8), (9')\}$. Note that the run might proceed slightly differently if different normal forms are chosen, but it will still succeed.

In Example 3.18 we argued that any KB run which initially orients (2) must fail—but our completion run did not fail despite the fact that (2) was oriented in the beginning. This example thus illustrates that in contrast to the completion methods considered in previous chapters, the completion procedure induced by the constraint equality framework is not vulnerable to unfortunate selection sequences.

7.3 Ordered Completion

We will now instantiate the above framework to ordered completion. The function F is slightly changed in that it gets the CES $\mathcal{C}$ as additional parameter.

Definition 7.15. Given an ES $\mathcal{E}$, let S_{O} be the standard transformer such that for any CES $\mathcal{C}$ every $\mathcal{R}$ in $\mathfrak{R}(\mathcal{C})$ is totally terminating and satisfies $\mathcal{R} \subseteq \leftrightarrow_{\mathcal{E}}^*$, and $F(\mathcal{R}, \mathcal{C}) = \mathrm{CP}_{\triangleright}(\mathcal{R} \cup \mathcal{C}[\![\mathcal{R}]\!])$.

Before being able to prove correctness of an ordered completion procedure relying on S_{O} we need some auxiliary results.

Lemma 7.16. *If $\mathcal{R} \subseteq \succ$ for some total reduction order $\succ$ and $\mathrm{CP}_{\succ}(\mathcal{R} \cup \mathcal{E}) \subseteq \downarrow_{\mathcal{R}} \cup \leftrightarrow_{\mathcal{E}}$ then $(\mathcal{R}, \mathcal{E})$ is ground convergent with respect to $\succ$.*

Proof. Let $\mathcal{R}' = \mathcal{R} \cup \mathcal{E}_{\succ}$. This (possibly infinite) TRS is obviously terminating. The assumption $\mathrm{CP}_{\succ}(\mathcal{R} \cup \mathcal{E}) \subseteq \downarrow_{\mathcal{R}} \cup \leftrightarrow_{\mathcal{E}}$ implies that $\mathrm{CP}(\mathcal{R}') \subseteq \downarrow_{\mathcal{R}} \cup \leftrightarrow_{\mathcal{E}}$ and thus $\mathrm{CP}(\mathcal{R}') \subseteq \downarrow_{\mathcal{R}'}$. By the Critical Pair Lemma 3.8 the TRS $\mathcal{R}'$ is thus locally ground confluent, and by Newman's Lemma ground confluent. Hence $(\mathcal{R}, \mathcal{E})$ is ground convergent with respect to $\succ$. □

Lemma 7.17. *If $S(\mathcal{C})[\![\mathcal{R}]\!] = \mathcal{C}[\![\mathcal{R}]\!]$ and $\mathcal{R} \in \mathfrak{R}(\mathcal{C})$, then $F(\mathcal{R}, \mathcal{C}) \subseteq \downarrow_{\mathcal{R}} \cup \leftrightarrow_{\mathcal{C}[\![\mathcal{R}]\!]}$.*

Proof. If $S(\mathcal{C})[\![\mathcal{R}]\!] = \mathcal{C}[\![\mathcal{R}]\!]$ then $S_{\mathcal{R}}(\mathcal{C})[\![\mathcal{R}]\!] = \mathcal{C}[\![\mathcal{R}]\!]$ for all $\mathcal{R} \in \mathfrak{R}(\mathcal{C})$. Let $\mathcal{R} \in \mathfrak{R}(\mathcal{C})$ and $s \approx t \in F(\mathcal{R})$, so $\langle s{\downarrow} \approx t{\downarrow}, \bigwedge \mathcal{R} \rangle \in F(\mathcal{R}, \mathcal{C})^{\top}{\downarrow_{\mathcal{R}}}$. If $s \leftrightarrow_{\mathcal{C}[\![\mathcal{R}]\!]} t$ does not hold then we must have $\{\langle s{\downarrow_{\mathcal{R}}} \approx t{\downarrow_{\mathcal{R}}}, \bigwedge \mathcal{R} \rangle\}[\![\mathcal{R}]\!] = \varnothing$. From $\mathcal{R} \models \bigwedge \mathcal{R}$ it follows that $s{\downarrow_{\mathcal{R}}} = t{\downarrow_{\mathcal{R}}}$, so $s \downarrow_{\mathcal{R}} t$. □

The following result states that if the $\mathcal{R}$-projection of a CES $\mathcal{C}$ is saturated with respect to the ground transformer S_{O} then the system $(\mathcal{C}[\![\mathcal{R}]\!], \mathcal{R})$ is ground-convergent. It thus proposes an ordered completion procedure based on the constrained equality framework.

Theorem 7.18. *Let $\mathcal{C} = S_{\mathrm{O}}^n(\mathcal{E}^{\top})$ such that $S_{\mathrm{O}}(\mathcal{C})[\![\mathcal{R}]\!] = \mathcal{C}[\![\mathcal{R}]\!]$ for some $\mathcal{R} \in \mathfrak{R}(\mathcal{C})$. Then $(\mathcal{C}[\![\mathcal{R}]\!], \mathcal{R})$ is ground convergent for $\mathcal{E}$ with respect to a ground-total reduction order $\succ$.*

Proof. Let $\succ$ be a ground-total reduction order such that $\mathcal{R} \subseteq \succ$ holds. Such an ordering exists as $\mathcal{R}$ is totally terminating. Thus every ground instance of an equation in $\mathcal{C}[\![\mathcal{R}]\!]$ is orientable by $\succ$. Let $\mathcal{R}' = \mathcal{C}[\![\mathcal{R}]\!]_{\succ}$ be the TRS consisting of all these ground instances. All equations in $\mathcal{C}[\![\mathcal{R}]\!]$ are trivially ground-joinable by $\mathcal{R}'$, and thus also by $\mathcal{R} \cup \mathcal{R}'$. As S_{O}^n is a transformer by Corollary 7.10 and $S_{\mathrm{O}}(\mathcal{C})[\![\mathcal{R}]\!] = \mathcal{C}[\![\mathcal{R}]\!]$, all equations in $\mathcal{E}$ are ground-joinable by $\mathcal{R} \cup \mathcal{R}'$.

By Lemma 7.17, $\mathrm{CP}_{\triangleright}(\mathcal{R} \cup \mathcal{C}[\![\mathcal{R}]\!]) \subseteq \downarrow_{\mathcal{R}} \cup \mathcal{C}[\![\mathcal{R}]\!]$. As $\triangleright \subseteq \succ$ implies $\mathrm{CP}_{\succ}(\mathcal{R} \cup \mathcal{C}[\![\mathcal{R}]\!]) \subseteq \mathrm{CP}_{\triangleright}(\mathcal{R} \cup \mathcal{C}[\![\mathcal{R}]\!])$ we have $\mathrm{CP}_{\succ}(\mathcal{R} \cup \mathcal{C}[\![\mathcal{R}]\!]) \subseteq \downarrow_{\mathcal{R}} \cup \mathcal{C}[\![\mathcal{R}]\!]$. By Lemma 7.16 it follows that $(\mathcal{C}[\![\mathcal{R}]\!], \mathcal{R})$ is ground convergent for $\mathcal{E}$ with respect to $\succ$. □

The following example illustrates the induced ordered completion procedure.

Example 7.19. Consider the ES $\mathcal{E}$ from Example 5.13. From the three initial constrained equalities $\mathcal{C}_0 = \mathcal{E}^\top = \{\langle(1), \top\rangle, \langle(2), \top\rangle, \langle(3), \top\rangle\}$ we may choose $\mathcal{R}_0 = \{(1), (2)\} \in \mathfrak{R}(\mathcal{C}_0)$. Abbreviating $R_0 = (1) \wedge (2)$ results in the CES

$$\mathcal{C}_1 = \{\langle(1), \neg R_0\rangle, \langle(2), \neg R_0\rangle, \langle(3), \top\rangle, \langle(4), R_0\rangle, \langle(5), R_0\rangle\}$$

where (4) is derived from the critical overlap $\langle(3), 1, (1)\rangle$, while (5) stems from rewriting the extended critical pair obtained from $\langle(3'), 1, (2)\rangle$:

$$(x + -x) \cdot 1 \approx 0 \tag{4}$$
$$0 \approx x + -x \tag{5}$$

Now we can for instance take $\mathcal{R}_1 = \{(1), (2), (4), (5')\} \in \mathfrak{R}(\mathcal{C}_1)$. We write R_1 as a shorthand for $1 \wedge 2 \wedge 4 \wedge 5'$, and obtain

$$\begin{aligned}\mathcal{C}_2 =& \{\langle(1), \neg R_0 \wedge \neg R_1\rangle, \langle(2), \neg R_0 \wedge \neg R_1\rangle, \langle(3), \neg R_1\rangle, \langle(4), R_0 \wedge \neg R_1\rangle\} \cup \\ & \{\langle(5), R_0 \wedge \neg R_1\rangle, \langle(6), R_1\rangle, \langle(7), R_1\rangle\}\end{aligned}$$

where (6) and (7) are obtained from rewriting (2) with $(5')$ and the extended critical overlap $\langle(3'), \epsilon, (5')\rangle$, respectively:

$$0 \cdot 1 \approx 0 \tag{6}$$
$$-x + x \approx 0 \tag{7}$$

For $\mathcal{R}_2 = \{(5'), (6), (7)\} \in \mathfrak{R}(\mathcal{C}_2)$ and $\mathcal{C}_3 = S_{\mathrm{O}}(\mathcal{C}_2)$ we now have $\mathcal{C}_3[\![\mathcal{R}_2]\!] = S_{\mathrm{O}}(\mathcal{C}_3)[\![\mathcal{R}_2]\!]$, so $(\mathcal{C}_3[\![\mathcal{R}_2]\!], \mathcal{R}_2)$ is ground convergent for $\mathcal{E}$; and as $\mathcal{C}_3[\![\mathcal{R}_2]\!]$ is empty, $\mathcal{R}_2$ is actually convergent for $\mathcal{E}$.

We remark that in contrast to the confluence criterion $\mathrm{CP}(\mathcal{R}) \subseteq \downarrow_{\mathcal{R}}$ used for standard completion, the criterion for ground convergence stated in Lemma 7.16 is not a necessary one.

Example 7.20. Consider the system $(\mathcal{E}, \mathcal{R})$:

$$(x \cdot y) \cdot z \to x \cdot (y \cdot z) \qquad x \cdot y \approx y \cdot x \qquad x \cdot (y \cdot z) \approx y \cdot (x \cdot z)$$

Let $\succ$ be an LPO. Then $(\mathcal{E}, \mathcal{R})$ does not satisfy the criterion in Lemma 7.16 as witnessed by $x \cdot (z \cdot y) \leftarrow \rtimes \rightarrow y \cdot (x \cdot z) \in \mathrm{CP}_{\succ}(\mathcal{E} \cup \mathcal{R})$. But $(\mathcal{E}, \mathcal{R})$ can be shown ground-convergent by checking all possible relationships between ground instantiations of variables occurring in critical pairs [78].

In fact ground confluence of terminating ordered rewrite systems was shown to be undecidable [52], although it is decidable for systems compatible with a certain class of reduction orders (including recursive path orders) [27].

7.4 Normalized Completion

We next discuss a normalized completion procedure based on constrained equalities. Since the underlying rewrite relation is different from the setting in the

previous sections we will first adapt some definitions, borrowing notation from Chapter 6. Let $\mathcal{S}$ be a fixed AC-convergent TRS for the theory $\mathcal{T}$. In the sequel we will only consider TRSs $\mathcal{R}$ such that $\mathcal{R} \cup \mathcal{S}$ is AC terminating, and hence also $\to_{\mathcal{R}\backslash\mathcal{S}}$ is AC terminating. For a TRS $\mathcal{R}$ and a term s, let $s{\downarrow\downarrow}_{\mathcal{R}\backslash\mathcal{S}}$ denote a term t such that $s \to^!_{\mathcal{R}\backslash\mathcal{S}} \cdot \to^!_{\mathcal{S}/\mathrm{AC}} t$. We define $\mathcal{C}{\downarrow\downarrow}_{\mathcal{R}\backslash\mathcal{S}}$ as follows:

$$\mathcal{C}{\downarrow\downarrow}_{\mathcal{R}\backslash\mathcal{S}} = \left\{ \left\langle s{\downarrow\downarrow}_{\mathcal{R}\backslash\mathcal{S}} \approx t{\downarrow\downarrow}_{\mathcal{R}\backslash\mathcal{S}}, C \wedge \bigwedge \mathcal{R} \right\rangle \mid \langle s \approx t, C\rangle \in \mathcal{C} \text{ and } s{\downarrow\downarrow}_{\mathcal{R}\backslash\mathcal{S}} \not\approx t{\downarrow\downarrow}_{\mathcal{R}\backslash\mathcal{S}} \right\}$$

Next, the notion of a transformer is adapted to the notion of an $\mathcal{S}$-transformer which takes $\mathcal{S}$-normalized rewriting into account.

Definition 7.21. A function S constitutes an *$\mathcal{S}$-transformer* if for any CES $\mathcal{C}$ and TRS $\mathcal{R}$ we have $\mathcal{C}[\![\mathcal{R}]\!] \subseteq {\downarrow\downarrow}_{\mathcal{R}\backslash\mathcal{S}}$ whenever $S(\mathcal{C})[\![\mathcal{R}]\!] \subseteq {\downarrow\downarrow}_{\mathcal{R}\backslash\mathcal{S}}$.

Similar as for transformers in Lemma 7.7 we give a simple criterion for a function to be an $\mathcal{S}$-transformer.

Lemma 7.22. *S is an $\mathcal{S}$-transformer if for any CES $\mathcal{C}$ and TRS $\mathcal{R}$*

$$\mathcal{C}[\![\mathcal{R}]\!] \subseteq \xrightarrow[\mathcal{R}\backslash\mathcal{S}]{*} \cdot \xrightarrow[\mathcal{S}/\mathrm{AC}]{*} \cdot \xleftrightarrow[S(\mathcal{C})[\![\mathcal{R}]\!]]{=} \cdot \xleftarrow[\mathcal{S}/\mathrm{AC}]{*} \cdot \xleftarrow[\mathcal{R}\backslash\mathcal{S}]{*} \tag{7.2}$$

Proof. Suppose $S(\mathcal{C})[\![\mathcal{R}]\!] \subseteq {\downarrow\downarrow}_{\mathcal{R}\backslash\mathcal{S}}$. By assumption, for any $s \approx t \in \mathcal{C}[\![\mathcal{R}]\!]$ there are terms u and v such that $s \to^*_{\mathcal{R}\backslash\mathcal{S}} \cdot \to^*_{\mathcal{S}/\mathrm{AC}} u \leftrightarrow^=_{S(\mathcal{C})[\![\mathcal{R}]\!]} v \ {}^*_{\mathcal{S}/\mathrm{AC}}\!\leftarrow \cdot \ {}^*_{\mathcal{R}\backslash\mathcal{S}}\!\leftarrow t$. As $u {\downarrow\downarrow}_{\mathcal{R}\backslash\mathcal{S}} v$ also $s {\downarrow\downarrow}_{\mathcal{R}\backslash\mathcal{S}} t$ holds. □

Next we define a function which will turn out to form an $\mathcal{S}$-transformer. Its shape closely resembles that of a standard transformer, except that we now consider $\mathcal{S}$-normalized rewriting and respective normal forms.

Definition 7.23. Given an ES $\mathcal{E}$, let S_{N} be the mapping defined as follows:

$$S_{\mathcal{R}}(\mathcal{C}) = (\mathcal{C} \ominus \mathcal{R}) \ \cup \ \mathcal{C}{\downarrow\downarrow}_{\mathcal{R}\backslash\mathcal{S}} \ \cup F(\mathcal{R})^\top {\downarrow\downarrow}_{\mathcal{R}\backslash\mathcal{S}}$$
$$S_{\mathrm{N}}(\mathcal{C}) = \bigcup_{\mathcal{R}\in\mathfrak{R}(\mathcal{C})} S_{\mathcal{R}}(\mathcal{C})$$

where all $\mathcal{R}$ in $\mathfrak{R}(\mathcal{C})$ satisfy $\mathcal{R} \subseteq \leftrightarrow^*_{\mathcal{E}\cup\mathcal{T}}$, $\mathcal{R} \cup \mathcal{S}$ is AC terminating, and $F(\mathcal{R}) = \mathrm{CP}_{\mathcal{L}}(\mathcal{R}^e) \cup \mathrm{CP}_{\mathrm{AC}}(\mathcal{S}, \mathcal{R}) \cup \mathrm{CP}_{\mathrm{AC}}(\mathcal{R}, \mathcal{S}^e)$ for some theory $\mathcal{L}$ between AC and $\mathcal{T}$.

Lemma 7.24. *S_{N} is an $\mathcal{S}$-transformer.*

Proof. Let $s \approx t \in \mathcal{C}[\![\mathcal{H}]\!]$. According to Lemma 7.22 it is sufficient to show $s\sigma \to^*_{\mathcal{H}\backslash\mathcal{S}} \cdot \to^*_{\mathcal{S}/\mathrm{AC}} \cdot \leftrightarrow^=_{S_{\mathcal{R}}(\mathcal{C})[\![\mathcal{H}]\!]} \cdot \ {}^*_{\mathcal{S}/\mathrm{AC}}\!\leftarrow \cdot \ {}^*_{\mathcal{H}\backslash\mathcal{S}}\!\leftarrow t\sigma$ for any $\mathcal{R} \in \mathfrak{R}(\mathcal{C})$. By assumption $\langle s \approx t, C\rangle \in \mathcal{C}$ for some C such that $\mathcal{H} \models C$. We distinguish three cases.

- If $\mathcal{R} \subseteq \mathcal{H}$ and $s \downarrow\downarrow_{\mathcal{R}\backslash\mathcal{S}} t$ then also $s \downarrow\downarrow_{\mathcal{H}\backslash\mathcal{S}} t$.
- If $\mathcal{R} \subseteq \mathcal{H}$ but $s \not\downarrow\downarrow_{\mathcal{R}\backslash\mathcal{S}} t$ then $\mathcal{H} \models C \wedge \bigwedge \mathcal{R}$, and $\langle s' \approx t', C \wedge \bigwedge \mathcal{R}\rangle \in \mathcal{C}{\downarrow\downarrow}_{\mathcal{R}\backslash\mathcal{S}}$, where $s' = s{\downarrow\downarrow}_{\mathcal{R}\backslash\mathcal{S}}$ and $t' = t{\downarrow\downarrow}_{\mathcal{R}\backslash\mathcal{S}}$. Thus $s' \approx t' \in S_{\mathcal{R}}(\mathcal{C})[\![\mathcal{H}]\!]$, so

$$s \xrightarrow[\mathcal{H}\backslash\mathcal{S}]{*} \cdot \xrightarrow[\mathcal{S}/\mathrm{AC}]{*} s' \xleftrightarrow[S_{\mathcal{R}}(\mathcal{C})[\![\mathcal{H}]\!]]{} t' \xleftarrow[\mathcal{S}/\mathrm{AC}]{*} \cdot \xleftarrow[\mathcal{H}\backslash\mathcal{S}]{*} t$$

– If $\mathcal{R} \not\subseteq \mathcal{H}$ then $\mathcal{H} \models C \wedge \neg \bigwedge \mathcal{R}$, and $\langle s \approx t, C \wedge \neg \bigwedge \mathcal{R} \rangle \in \mathcal{C} \ominus \mathcal{R}$. Thus we have $s \approx t \in S_{\mathcal{R}}(\mathcal{C})[\![\mathcal{H}]\!]$. □

Before stating a result that induced a normalized completion procedure we need to verify that consider the critical pairs in $F(\mathcal{R})$ is sufficient to guarantee normalized $\mathcal{S}$-convergence. The following result serves this purpose.

Lemma 7.25. *Suppose $\mathcal{R} \cup \mathcal{S}$ is AC terminating and*

$$\mathrm{CP}_{\mathcal{L}}(\mathcal{R}^e) \cup \mathrm{CP}_{\mathrm{AC}}(\mathcal{S}, \mathcal{R}) \cup \mathrm{CP}_{\mathrm{AC}}(\mathcal{R}, \mathcal{S}^e) \subseteq \mathrel{\downdownarrows}_{\mathcal{R}\setminus\mathcal{S}} \tag{7.3}$$

for some theory $\mathcal{L}$ such that $\mathrm{AC} \subseteq \mathcal{L} \subseteq \mathcal{T}$. Then $\mathcal{R}$ is $\mathcal{S}$-convergent.

Proof. Lemma 6.14 shows that $\mathrm{CP}_{\mathrm{AC}}(\mathcal{R}^e) \subseteq \leftrightarrow^*_{\mathcal{T}} \cdot \leftrightarrow_{\mathrm{CP}_{\mathcal{L}}(\mathcal{R}^e)} \cdot \leftrightarrow^*_{\mathcal{T}}$. By AC convergence of $\mathcal{S}$ for $\mathcal{T}$, assumption (7.3), and Lemma 6.5, the relation $\to_{(\mathcal{R}\cup\mathcal{S})/\mathrm{AC}}$ is thus locally confluent modulo AC. As $\to_{(\mathcal{R}\cup\mathcal{S})/\mathrm{AC}}$ is terminating it is confluent according to the counterpart of Newman's Lemma for confluence modulo an equivalence [45, Lemma 7].

Hence there exists a joining sequence $s \to^*_{(\mathcal{R}\cup\mathcal{S})/\mathrm{AC}} v \;{}^*_{(\mathcal{R}\cup\mathcal{S})/\mathrm{AC}}\!\!\leftarrow t$ for all terms s, t such that $s \leftrightarrow^*_{(\mathcal{R}\cup\mathcal{S})/\mathrm{AC}} t$. It remains to show that $s \mathrel{\downdownarrows}_{\mathcal{R}\setminus\mathcal{S}} t$. To this end, we assume that the term v is in normal form with respect to $\to_{(\mathcal{R}\cup\mathcal{S})/\mathrm{AC}}$, and show that $u \to^*_{\mathcal{R}\setminus\mathcal{S}} \cdot \to^*_{\mathcal{S}/\mathrm{AC}} v$ whenever $u \to^!_{(\mathcal{R}\cup\mathcal{S})/\mathrm{AC}} v$ for any term u. The proof is by induction on u and the well-founded relation $\to^+_{(\mathcal{R}\cup\mathcal{S})/\mathrm{AC}}$. In the base case u is in normal form and there is nothing to prove. Suppose $u \to_{(\mathcal{R}\cup\mathcal{S})/\mathrm{AC}} u' \to^!_{(\mathcal{R}\cup\mathcal{S})/\mathrm{AC}} v$. By the induction hypothesis, $u' \to^*_{\mathcal{R}\setminus\mathcal{S}} \cdot \to^*_{\mathcal{S}/\mathrm{AC}} v$. If $u \to_{\mathcal{R}/\mathrm{AC}} u'$ such that $u = u{\downarrow_{\mathcal{S}/\mathrm{AC}}}$ or $u \to_{\mathcal{S}/\mathrm{AC}} u'$ then it is easy to see that $u \to^*_{\mathcal{R}\setminus\mathcal{S}} \cdot \to^*_{\mathcal{S}/\mathrm{AC}} v$ holds as well. So assume $u \to_{\mathcal{R}/\mathrm{AC}} u'$ such that there is some term w with $u \to_{\mathcal{S}/\mathrm{AC}} w$. As $\to_{(\mathcal{R}\cup\mathcal{S})/\mathrm{AC}}$ is confluent we have $w \downarrow_{(\mathcal{R}\cup\mathcal{S})/\mathrm{AC}} v$, and as $v = v{\downarrow_{(\mathcal{R}\cup\mathcal{S})/\mathrm{AC}}}$ actually $w \to^*_{(\mathcal{R}\cup\mathcal{S})/\mathrm{AC}} v$ must hold. We may apply the induction hypothesis to conclude $w \to^*_{\mathcal{R}\setminus\mathcal{S}} \cdot \to^*_{\mathcal{S}/\mathrm{AC}} v$ and thus also $u \to^*_{\mathcal{R}\setminus\mathcal{S}} \cdot \to^*_{\mathcal{S}/\mathrm{AC}} v$. □

Similar to the cases of standard and ordered completion, we show that saturating a CES with respect to the $\mathcal{S}$-transformer S_{N} constitutes a normalized completion procedure.

Theorem 7.26. *Let $\mathcal{C} = S^n_{\mathrm{N}}(\mathcal{E}^\top)$. If $\mathcal{C}[\![\mathcal{R}]\!] = \varnothing$ for some $\mathcal{R} \in \mathfrak{R}(\mathcal{C})$ then $\mathcal{R}$ is $\mathcal{S}$-convergent for $\mathcal{E}$.*

Proof. Suppose $\mathcal{R} \in \mathfrak{R}(\mathcal{C})$ and $\mathcal{C}[\![\mathcal{R}]\!] = \varnothing$. By Lemma 7.24 S_{N} is an $\mathcal{S}$-transformer, and an inductive argument shows that also S^n_{N} is an $\mathcal{S}$-transformer. Hence $\mathcal{E} \subseteq \mathrel{\downdownarrows}_{\mathcal{R}\setminus\mathcal{S}}$, and since $\mathcal{R} \subseteq \leftrightarrow^*_{\mathcal{E}\cup\mathcal{T}}$ by assumption, the relations $\leftrightarrow^*_{\mathcal{E}\cup\mathcal{T}}$ and $\leftrightarrow^*_{\mathcal{R}\cup\mathcal{T}}$ coincide. If $n = 0$ then $\mathcal{E} \subseteq \leftrightarrow^*_{\mathcal{T}}$. As $\mathcal{R} \subseteq \leftrightarrow^*_{\mathcal{E}\cup\mathcal{T}}$ and $\mathcal{R} \cup \mathcal{S}$ is AC terminating we must have $\mathcal{R} \subseteq \to^+_{\mathcal{S}/\mathrm{AC}}$, so the claim holds by AC convergence of $\mathcal{S}$. If $n > 0$ it is not difficult to see that $\mathcal{C}[\![\mathcal{R}]\!] = \varnothing$ implies $S_{\mathrm{N}}(\mathcal{C})[\![\mathcal{R}]\!] = \varnothing$, so $F(\mathcal{R}) \subseteq \mathrel{\downdownarrows}_{\mathcal{R}\setminus\mathcal{S}}$. As $\mathcal{R} \cup \mathcal{S}$ is AC terminating, $\mathcal{R}$ is $\mathcal{S}$-convergent by Lemma 7.25. □

The following example illustrates how Theorem 7.26 induces a normalized completion procedure.

Example 7.27. Consider the TRS $\mathcal{S}$ representing Abelian groups:

$$\mathsf{e} \cdot x \to x \tag{1}$$
$$\mathsf{i}(x) \cdot x \to \mathsf{0} \tag{2}$$
$$\mathsf{i}(\mathsf{0}) \to \mathsf{0} \tag{3}$$
$$\mathsf{i}(\mathsf{i}(x)) \to x \tag{4}$$
$$\mathsf{i}(x \cdot y) \to \mathsf{i}(x) \cdot \mathsf{i}(y) \tag{5}$$

Let $\mathcal{L} = \mathrm{AC}$ and consider one equation axiomatizing a group endomorphism:

$$\mathsf{f}(x \cdot y) \approx \mathsf{f}(x) \cdot \mathsf{f}(y) \tag{6}$$

From the initial CES $\mathcal{C}_0 = \mathcal{E}^\top = \{\langle (6), \top \rangle\}$ we can for instance choose $\mathcal{R}_0 = \{(6')\}$. Writing $R_0 = (6')$, this results in the CES

$$\mathcal{C}_1 = \{\langle (6), \neg R_0 \rangle, \langle (7), R_0 \rangle, \langle (8), R_0 \rangle, \langle (9), R_0 \rangle, \langle (10), R_0 \rangle, \langle (11), R_0 \rangle\}$$

where the following new equations are in $\mathrm{CP}_{\mathrm{AC}}(\mathcal{S}_{\mathrm{G}}, (6')) \cup \mathrm{CP}_{\mathrm{AC}}((6'), \mathcal{S}^e_{\mathrm{G}})$:

$$\mathsf{i}(\mathsf{f}(x \cdot y)) \approx \mathsf{i}(\mathsf{f}(x)) \cdot \mathsf{i}(\mathsf{f}(y)) \tag{7}$$
$$\mathsf{f}(y) \approx \mathsf{i}(\mathsf{f}(x)) \cdot \mathsf{f}(x \cdot y) \tag{8}$$
$$\mathsf{i}(\mathsf{f}(x)) \cdot \mathsf{i}(\mathsf{f}(y)) \cdot \mathsf{f}(x \cdot y) \approx \mathsf{e} \tag{9}$$
$$\mathsf{i}(\mathsf{f}(x)) \cdot \mathsf{i}(\mathsf{f}(y)) \cdot \mathsf{i}(z) \approx \mathsf{i}(\mathsf{f}(x \cdot y)) \cdot \mathsf{i}(z) \tag{10}$$
$$\mathsf{i}(\mathsf{f}(x)) \cdot \mathsf{i}(\mathsf{f}(y)) \cdot \mathsf{i}(z) \cdot \mathsf{i}(w) \approx \mathsf{i}(\mathsf{f}(x \cdot y)) \cdot \mathsf{i}(z) \cdot \mathsf{i}(w) \tag{11}$$

Taking $\mathcal{R}_1 = \{(6), (7), (8'), (9), (10'), (11')\}$ now yields the CES

$$\begin{aligned}\mathcal{C}_2 = {} & \{\langle (6), \neg R_0 \wedge \neg R_2 \rangle, \langle (7), R_0 \wedge \neg R_1 \rangle, \langle (8), R_0 \wedge \neg R_1 \rangle, \langle (9), R_0 \wedge \neg R_1 \rangle\} \\ & \cup \{\langle (10), R_0 \wedge \neg R_1 \rangle, \langle (11), R_0 \wedge \neg R_1 \rangle, \langle (12), R_1 \rangle, \langle (13), R_1 \rangle\}\end{aligned}$$

for the following additional equations in $F(\mathcal{R}_1)\!\downarrow\!\!\downarrow_{\mathcal{R}_1 \setminus \mathcal{S}}$:

$$\mathsf{f}(\mathsf{e}) \cdot \mathsf{f}(x) \approx \mathsf{f}(x) \tag{12}$$
$$\mathsf{f}(\mathsf{i}(x)) \cdot \mathsf{f}(x) \approx \mathsf{f}(\mathsf{e}) \tag{13}$$

If we then choose $\mathcal{R}_2 = \mathcal{R}_1 \cup \{(12), (13)\}$, the resulting CES

$$\begin{aligned}\mathcal{C}_3 = {} & \{\langle (6), \neg R_0 \wedge \neg R_2 \rangle, \langle (7), R_0 \wedge \neg R_1 \rangle, \langle (8), R_0 \wedge \neg R_1 \rangle, \langle (9), R_0 \wedge \neg R_1 \rangle\} \\ & \cup \{\langle (10), R_0 \wedge \neg R_1 \rangle, \langle (11), R_0 \wedge \neg R_1 \rangle, \langle (12), R_1 \wedge \neg R_2 \rangle, \langle (13), R_1 \wedge \neg R_2 \rangle\} \\ & \cup \{\langle (14), R_2 \rangle, \langle (15), R_2 \rangle, \ldots\}\end{aligned}$$

considers the following equations:

$$\mathsf{f}(\mathsf{e}) \approx \mathsf{e} \tag{14}$$
$$\mathsf{f}(\mathsf{i}(x)) \approx \mathsf{i}(\mathsf{f}(x)) \tag{15}$$

besides 7 additional ones. For $\mathcal{R}_3 = \mathcal{R}_2 \cup \{(14), (15)\}$ we now have $S_{\mathrm{N}}(\mathcal{C}_3)[\![\mathcal{R}_3]\!] = \varnothing$, so $\mathcal{R}_3$ is $\mathcal{S}$-convergent for $\mathcal{E}$. This would also hold for the $\mathcal{S}$-canonical system $\mathcal{R}'_3 = \{(6), (14), (15)\}$.

7.5 Rewriting Induction

Finally the constrained completion framework can also be instantiated to an inductive theorem proving tool based on rewriting induction [55]. We briefly recall some relevant definitions related to inductive theorem proving.

Definition 7.28. An ES $\mathcal{H}$ is an *inductive consequence* of $\mathcal{R}_0$, denoted $\mathcal{R}_0 \vdash_i \mathcal{H}$, if $s\sigma \leftrightarrow^*_{\mathcal{R}_0} t\sigma$ for all $s \approx t$ in $\mathcal{H}$ and ground substitutions σ.

Let $\mathcal{R}_0$ be a TRS with defined symbols $\mathcal{F}_\mathcal{D}$ and constructor symbols $\mathcal{F}_\mathcal{C}$. A term $t = f(t_1, \ldots, t_n)$ is *basic* with respect to $\mathcal{R}_0$ if $f \in \mathcal{F}_\mathcal{D}$ and $t_i \in \mathcal{T}(\mathcal{F}_\mathcal{C}, \mathcal{V})$ for all $1 \leqslant i \leqslant n$. We call a position $p \in \mathcal{P}\mathrm{os}(t)$ basic in t if $t|_p$ is basic. A TRS is *quasi-reducible* if no ground basic term is in normal form.

Example 7.29. Let $\mathcal{F} = \{\mathsf{0}, \mathsf{s}, +\}$ be a sorted signature such that $\mathsf{0} : \mathit{nat}, \mathsf{s} : \mathit{nat} \to \mathit{nat}$ and $+ : \mathit{nat} \times \mathit{nat} \to \mathit{nat}$. Consider the following TRS $\mathcal{R}_0$:

$$x + \mathsf{0} \to x \tag{1}$$
$$x + \mathsf{s}(y) \to \mathsf{s}(x + y) \tag{2}$$

The only defined symbol in $\mathcal{R}_0$ is $+$. Hence any basic term must be of the form $\mathsf{s}^n(\mathsf{0}) + \mathsf{s}^m(\mathsf{0})$ for some $n, m \geqslant 0$. Obviously no such term is in normal form, therefore $\mathcal{R}_0$ is quasi-reducible.

Definition 7.30. Let $\mathcal{R}_0$ and $\mathcal{R}$ be TRSs such that $\mathcal{R}_0$ is quasi-reducible. Then $\mathcal{R}$ is $\mathcal{R}_0$-*expandable* if for every $\ell \to r \in \mathcal{R}$ there exists a position in ℓ which is basic with respect to $\mathcal{R}_0$. We write $\mathrm{Expd}(\mathcal{R}_0, \mathcal{R})$ for the set of all critical pairs originating from overlaps $\langle \ell_1 \to r_1, p, \ell_2 \to r_2 \rangle$ such that $\ell_1 \to r_1 \in \mathcal{R}_0$, $\ell_2 \to r_2 \in \mathcal{R}$, and p is a fixed (depending on $\ell_2 \to r_2$) basic position in ℓ_2.

Example 7.31. The TRS $\mathcal{R}$ consisting of the single rule

$$(x + y) + z \to x + (y + z) \tag{3}$$

is $\mathcal{R}_0$-expandable with respect to the TRS $\mathcal{R}_0$ from Example 7.29 as position 1 in the left-hand side is basic. Because of the overlaps $\langle (1), 1, (3) \rangle$ and $\langle (2), 1, (3) \rangle$ we have $\mathrm{Expd}(\mathcal{R}_0, \mathcal{R}) = \{x + z \approx x + (\mathsf{0} + z),\ \mathsf{s}(x + y) + z \approx x + (\mathsf{s}(y) + z)\}$.

Let $\mathfrak{R}(\mathcal{C})$ be a set of $\mathcal{R}_0$-expandable, terminating TRSs such that for every $\mathcal{R} \in \mathfrak{R}(\mathcal{C})$ we have $\mathcal{R}_0 \subseteq \mathcal{R}$ and $\ell\sigma \leftrightarrow^*_{\mathcal{R}_0 \cup \mathcal{E}} r\sigma$ for all $\ell \to r \in \mathcal{R}$ and ground substitutions σ. Let $S_{\mathrm{RI}}(\mathcal{C})$ be the standard transformer such that $F(\mathcal{R}) = \mathrm{Expd}(\mathcal{R}_0, \mathcal{R})$. The following result by Reddy [86] forms the basis of rewriting induction.

Lemma 7.32. *Let $\mathcal{R}$ and $\mathcal{H}$ be TRSs such that $\mathcal{R}$ is quasi-reducible and $\mathcal{R} \cup \mathcal{H}$ is terminating. If* $\mathrm{Expd}(\mathcal{R}, \mathcal{H}) \subseteq \downarrow_{\mathcal{R} \cup \mathcal{H}}$ *then $\mathcal{R} \vdash_i \mathcal{H}$.* □

Lemma 7.32 is now used to show that saturating a CES with respect to the ground transformer S_{RI} constitutes a rewriting induction procedure.

Theorem 7.33. *Let $\mathcal{C} = S^n_{\mathrm{RI}}(\mathcal{E}^\top)$. If $\mathcal{C}[\![\mathcal{R}]\!] = \varnothing$ for some $\mathcal{R} \in \mathfrak{R}(\mathcal{C})$ then $\mathcal{R}_0 \vdash_i \mathcal{E}$.*

Proof. Let $\mathcal{R} \in \mathfrak{R}(\mathcal{C})$ and $\mathcal{C}[\![\mathcal{R}]\!] = \varnothing$. By Lemma 7.9 the function S_{RI} is a ground transformer, so $\mathcal{E}^{\top}$ is ground joinable by $\mathcal{R}$. If $n = 0$ then $\mathcal{E}$ must be empty such that the claim trivially holds. Otherwise, since $\mathcal{C}[\![\mathcal{R}]\!] = \varnothing$ also $S_{\mathrm{RI}}(\mathcal{C})[\![\mathcal{R}]\!] = \varnothing$. Hence $\mathrm{Expd}(\mathcal{R}_0, \mathcal{R})$ is ground joinable by $\mathcal{R}$. Since $\mathcal{R}_0 \subseteq \mathcal{R}$ and $\mathcal{R}$ is terminating, $\mathcal{R}_0 \vdash_i \mathcal{R}$ holds by Lemma 7.32. As $\mathcal{E}^{\top}$ is ground joinable by $\mathcal{R}$ also $\mathcal{R}_0 \vdash_i \mathcal{E}$ holds. □

To illustrate our rewriting induction procedure we verify that associativity is an inductive consequence of the TRS $\mathcal{R}_0$ used as running example.

Example 7.34. Consider the quasi-reducible TRS $\mathcal{R}_0$ from Example 7.29. Let $\mathcal{E}$ consist of the single equality

$$(x + y) + z \approx x + (y + z) \tag{3}$$

We obtain the initial CES $\mathcal{C}_0 = \{\langle (3), \top \rangle\}$. Suppose $\mathfrak{R}(\mathcal{C}_0)$ consists of the TRS $\mathcal{R}_1 = \{(1), (2), (3)\}$. Then expanding (3) at the basic position 1 results in

$$\mathcal{C}_1 = \{\langle (3), \neg R_1 \rangle, \langle (4), R_1 \rangle, \langle (5), R_1 \rangle\rangle\}$$

with

$$x + z \approx x + (\mathsf{0} + z) \tag{4}$$

$$\mathsf{s}(x + y) + z \approx x + (\mathsf{s}(y) + z) \tag{5}$$

In fact it is not difficult to see that orienting (5) and (3) from left to right will lead to an infinite run. However, if $\mathfrak{R}(\mathcal{C}_1)$ contains $\mathcal{R}_2 = \{(1), (2), (3')\}$ then expansion of $(3')$ at position 2 considers the overlaps $\langle (1), 2, (3') \rangle$ and $\langle (2), 2, (3') \rangle$. These give rise to the equations

$$x + y \approx (x + y) + \mathsf{0} \qquad\qquad x + \mathsf{s}(y + z) \approx (x + y) + \mathsf{s}(z)$$

which are both joinable by $\mathcal{R}_2$. Hence we obtain

$$\mathcal{C}_2 = \{\langle (3), \neg R_1 \wedge \neg R_2 \rangle, \langle (4), R_1 \wedge \neg R_2 \rangle, \langle (5), R_1 \wedge \neg R_2 \rangle\}$$

and as $\mathcal{C}[\![\mathcal{R}_2]\!] = \varnothing$ we conclude that $\mathcal{R}_0 \vdash_i \mathcal{E}$.

This example illustrates that similar as in Knuth-Bendix completion, the orientation of an equation to a rule is crucial, and a bad choice can easily lead to divergence. But in the constrained completion setting a bad choice can always be recovered at a later stage.

It is worth noting that the maximality principle which forms the basis of maximal completion is not appropriate for rewriting induction. Maximal completion may always orient as many rules as possible because of the following fact: Suppose $\mathcal{R} \subseteq \leftrightarrow_{\mathcal{E}}^*$, $\mathcal{R}$ is terminating, and $\mathrm{CP}(\mathcal{R}) \subseteq \downarrow_{\mathcal{R}}$. If $\mathcal{R}' \subseteq \leftrightarrow_{\mathcal{E}}^*$, $\mathcal{R} \subseteq \mathcal{R}'$, and $\mathcal{R}'$ is terminating then $\mathrm{CP}(\mathcal{R}') \subseteq \downarrow_{\mathcal{R}'}$.

In case of rewriting induction, this corresponds to the following property: Suppose $\mathcal{R}_0$ is quasi-reducible and $\mathcal{R}$ is a TRS such that $\mathcal{R}_0 \vdash_i \mathcal{R}$, $\mathcal{R}_0 \cup \mathcal{R}$ is terminating, and $\mathrm{Expd}(\mathcal{R}_0, \mathcal{R}) \subseteq \downarrow_{\mathcal{R}_0 \cup \mathcal{R}}$. If $\mathcal{R} \subseteq \mathcal{R}'$ and $\mathcal{R}_0 \cup \mathcal{R}'$ is terminating as well then also $\mathrm{Expd}(\mathcal{R}_0, \mathcal{R}') \subseteq \downarrow_{\mathcal{R}_0 \cup \mathcal{R}'}$. But this need not hold, as the following example shows [55]:

Example 7.35. Consider the following TRS $\mathcal{R}_0$:

$$\mathsf{0} + x \to x \tag{1}$$
$$\mathsf{s}(x) + y \to \mathsf{s}(x + y) \tag{2}$$

and the equations

$$(x + y) + z \approx x + (y + z) \tag{3}$$
$$x + \mathsf{s}(x) \approx \mathsf{s}(x + x) \tag{4}$$

For $\mathcal{R} = \{(3)\}$ and $\mathcal{R}' = \{(3), (4)\}$ we have $\mathcal{R}_0 \vdash_i \mathcal{R}'$. The TRS $\mathcal{R}'$ is terminating and $\mathrm{Expd}(\mathcal{R}_0, \mathcal{R}) \subseteq \downarrow_{\mathcal{R}_0 \cup \mathcal{R}}$ if (3) is expanded at position 1, but if (4) is expanded at position ϵ then $\mathrm{Expd}(\mathcal{R}_0, \mathcal{R}') \not\subseteq \downarrow_{\mathcal{R}_0 \cup \mathcal{R}'}$ as $\mathsf{s}(x + \mathsf{s}(\mathsf{s}(x))) \not\downarrow_{\mathcal{R}_0 \cup \mathcal{R}'} \mathsf{s}(\mathsf{s}(x) + \mathsf{s}(x))$.

Chapter 8

Implementation and Experiments

In this chapter we describe our implementations of a tool for multi-completion with termination tools as well as a completion tool based on the constrained equality framework, and report on experimental results.

8.1 A Multi-Completion Tool

Our tool mkbTT implements classical, ordered, and normalized multi-completion with termination tools. The tool is written in the programming language OCaml. Binaries and sources are available from the tool's website

`http://cl-informatik.uibk.ac.at/mkbtt/`

where also a web interface can be found.

8.1.1 Implementation

In the following sections we provide implementation details which were found to be of special importance.

Control

The basic control of mkbTT is a multi-completion variant of a DISCOUNT loop, very similar to the one originally proposed for completion with multiple reduction orders [62]. Pseudo-code describing the control loop is given in Figure 8.1. The procedure advances two node sets containing *open* nodes No and *closed* nodes Nc, corresponding to passive and active facts. Intuitively, closed nodes have been fully exploited with respect to the orient and rewrite inference rules, and to every pair of closed nodes deduce has been applied exhaustively. Therefore, at the beginning of a run Nc is empty whereas No contains the set of initial nodes.

We briefly describe the components occurring in the main control loop. At the beginning of each recursive call it is checked whether some process succeeded. In standard and normalized completion, a process p is considered *successful* if it does not occur in an open node or in a closed equation label, i.e., all of $E[Nc, p]$, $R[No, p]$ and $E[No, p]$ are empty. In the setting of ordered completion, a process p is successful if it does not occur in No. If no successful process exists but there are open nodes left, *choose* selects a node n from the set of open nodes. Different selection strategies are considered for this purpose (see Section 8.1.1).

```
procedure mkbTT(No, Nc)
if ∃ successful process p then return p
else if No = ∅ then fail
else n := choose(No) ;
     No := (No \ {n}) ⊎ rewrite({n}, Nc) ;
     if n ≠ ⟨..., ∅, ∅, ∅, ∅, ∅⟩ then
        (n, No, Nc) := orient(n, No, Nc) ;
        if n ≠ ⟨..., ∅, ∅, ..., ..., ...⟩ then
           No := No ⊎ delete(rewrite(Nc, {n})) ;
           Nc := gc(Nc) ;
           Nd := deduce(n, Nc) ;
           No := No ⊎ gc(delete(Nd ⊎ rewrite(Nd, Nc))) ;
           Nc := Nc ⊎ {n} ;
mkbTT(No, Nc) ;
```

Figure 8.1: Control in mkbTT.

The function $rewrite(N, N')$ applies rewrite to nodes in N by employing nodes in N' as rules. Nodes are implemented as mutable structures, so the original objects are modified and only newly created nodes are returned. In $\mathcal{S}$-normalized completion runs, all nodes are immediately $\mathcal{S}$-normalized upen creation. The function call $orient(n, No, Nc)$ is used to apply the orient inference to node n. It returns the modified node along with the node sets No and Nc adapted by the *split* operation. Immediately after *rewrite* and *deduce* calls, *delete* is invoked to filter out nodes with equal terms. After having been subjected to *rewrite*, all labels in a node might become empty. The purpose of *gc* is to filter out such nodes. The call $deduce(n, Nc)$ returns all equational consequences originating from deduce inference steps involving node n together with some node from Nc. To avoid redundant nodes, the union operation $\uplus$ exploits the subsume inference.

Termination Checking

Two possibilities for termination checks in orient inference steps are supported. Our tool can interface any external termination checker which complies to a minimal interface: It must take as argument the name of a file specifying the termination problem in TPDB format[1] and print `YES` on the first line of the output if (AC) termination could be established. Alternatively, termination checks can be performed internally by interfacing T$_{\text{T}}$T$_2$ [60] with the user's favourite termination strategy supplied in T$_{\text{T}}$T$_2$'s strategy language. For AC-termination, we added implementations of AC-RPO [87] and AC-KBO [94], both based on SMT encodings.

1 `http://www.lri.fr/~marche/tpdb/`

Term Indexing

Equational reasoning tools typically spend a major part of computation time on rewriting and deducing equational consequences. A variety of sophisticated term indexing techniques [93] have been developed in order to speed up filtering out matching and unifiable terms. Also mkbTT relies on indexing techniques to quickly sift through nodes that are applicable for inferences. For instance, for deduce inferences the retrieval of unifiable (sub)terms is needed. For rewrite steps, encompassments have to be retrieved, which can be achieved by repeatedly retrieving subsumptions. To select unifiable terms, mkbTT implements *path indexing* and *discrimination trees* [41,79], for variant and subsumption retrieval also *code trees* [103] are supported. Moreover, AC discrimination trees [8] are used for subsumption retrieval if normalized completion is applied and AC operators are present.

Selection Strategies

An iteration of mkbTT's main control loop starts by selecting a node to process next. For this purpose a *choice* function picks the node where a given cost heuristic evaluates to a minimal value. The measure applied in this selection has significant impact on performance. To allow for a variety of possibilities, a strategy language is defined that is general enough to cover selection strategies that proved to be useful, but also captures some concepts used in choice strategies of other tools. A strategy is specified by the grammar rule

$$\mathit{strategy} \;::=\; \texttt{?} \mid \texttt{(}\mathit{node}\texttt{,}\mathit{strategy}\texttt{)} \mid \mathit{float}\texttt{(}\mathit{strategy} : \mathit{strategy}\texttt{)}$$

Here *node* refers to a node property, which is in turn defined via properties of process sets, processes, sets of equations, rewrite systems and term pairs:

$$\begin{aligned}
\mathit{node} &::= \texttt{data(}\mathit{term_pair}\texttt{)} \mid \texttt{el(}\mathit{process_set}\texttt{)} \mid \texttt{-}\,\mathit{node} \mid \mathit{node}\;\texttt{+}\;\mathit{node} \mid \texttt{*} \\
\mathit{process_set} &::= \texttt{min(}\mathit{process}\texttt{)} \mid \texttt{sum(}\mathit{process}\texttt{)} \mid \texttt{\#} \\
\mathit{process} &::= \mathit{process}\;\texttt{+}\;\mathit{process} \mid \texttt{e(}\mathit{eqs}\texttt{)} \mid \texttt{r(}\mathit{trs}\texttt{)} \mid \texttt{c(}\mathit{trs}\texttt{)} \\
\mathit{eqs} &::= \texttt{sum(}\mathit{term_pair}\texttt{)} \mid \texttt{\#} \\
\mathit{trs} &::= \texttt{sum(}\mathit{term_pair}\texttt{)} \mid \texttt{cp(}\mathit{eqs}\texttt{)} \mid \texttt{\#} \\
\mathit{term_pair} &::= \texttt{smax} \mid \texttt{ssum}
\end{aligned}$$

The properties forming the basic elements of strategies are captured by integer values. The following paragraphs explain the different components.

- A *node property* of a node $n = \langle s : t, \ldots, E, \ldots \rangle$ can be its creation time (denoted by `*`), a property of the node's data $s : t$, or a process set property *pp* of its equation labels E, which is written as `el(`*pp*`)`. Node properties can also be added or inverted.

- A *process set property* takes either the sum or the minimum over a property of the involved processes, or may simply be the number of processes it contains, which is denoted by `#`.

- Given a current node set N, a *process property* of a process p may be an equation set property *ep* of its equation projection $E[N, p]$ or a rewrite system property *tp* of either its rule projection $R[N, p]$ or its constraint projection $C[N, p]$, which is expressed by writing `e(`*ep*`)`, `r(`*tp*`)` and `c(`*tp*`)`, respectively. Process properties can also be added.
- An *equation set property* of a set of equations $\mathcal{E}$ can be its cardinality (`#`) or the sum over a term pair property of the contained equations. A *rewrite system property* of a rewrite system $\mathcal{R}$ can additionally be a property of its set of critical pairs $\mathrm{CP}(\mathcal{R})$.
- A *term pair property* of $s : t$ can be the sum $|s| + |t|$ or maximum $\max\{|s|, |t|\}$ of the sizes of the involved terms, which is specified by writing `ssum` and `smax`.
- Finally, properties are combined to obtain selection strategies. The simplest possible strategy is `?`, which is implemented by picking a node randomly. Alternatively, a strategy may combine a node property *np* with another strategy *s* to a tuple `(`*np*`,`*s*`)`. By using this rule multiple times, a node property list of the form `(`np_1`,` $\ldots$ `(`np_k`,?)` $\ldots$ `)` can be constructed. To mix strategies, a strategy can also be of the shape r`(`s_1`:`s_2`)`, where r is assumed to be a rational value in the closed interval $[0, 1]$. Node property lists are evaluated by mapping each node to the corresponding tuple of integer values, its *cost*, and choosing the (lexicographic) minimum. In case of a mixed strategy r`(`s_1`:`s_2`)`, the strategy s_1 is applied with probability r, and s_2 is used in the remaining cases.

As selection measures have considerable impact, many different strategies for automated reasoning tools have been reported in the literature. For instance, Vampire [104] employs a size/age ratio when deciding on a fact to be processed next. If this ratio is (e.g.) $2 : 3$ then out of 5 selections 2 will pick the oldest and 3 the smallest node, i.e., the node where the sum of its term sizes $|s| + |t|$ is minimal. In mkbTT this approach can be achieved with the strategy

$$s_{\texttt{size/age}}(r) = r\texttt{((data(ssum),?):(*,?))}$$

where the parameter $r \in [0, 1]$ controls the ratio of size-determined selections, e.g., a size/age ratio of $2 : 3$ corresponds to $r = 0.6$.

The "best-first" selection approach applied in Slothrop [107] corresponds to advancing a process for which $|E[N, p]| + |\mathrm{CP}(R[N, p])| + |C[N, p]|$ is minimal. When combined with, for example, a size/age ratio, this is expressed as follows:

$$s_{\texttt{slothrop}}(r) = \texttt{(el(min(e(\#)+r(cp(\#))+c(\#))),}s_{\texttt{size/age}}(r)\texttt{)}$$

In mkbTT, the strategies $s_{\texttt{max}}$ and $s_{\texttt{sum}}$ turned out to be beneficial. These strategies first restrict attention to processes where the number of symbols in $E[N, p]$ and $C[N, p]$ is minimal, then select nodes with minimal data and finally go for a node which has the greatest number of processes in its equation label:

$$s_{\texttt{sum}} = \texttt{(el(min(e(sum(ssum))+c(sum(ssum)))),(data(ssum),(-el(\#),?))))}$$

The strategy $s_{\texttt{max}}$ differs from $s_{\texttt{sum}}$ only in that `ssum` is replaced by `smax`. To use mkbTT with other heuristics than those just described, a user-defined strategy can be specified via a command line option.

Certification

If mkbTT is run in standard completion mode and T$_\text{T}$T$_2$ is interfaced internally, it can produce a proof in CPF format. This output contains a termination proof (provided by T$_\text{T}$T$_2$) and a proof that the resulting rewrite rules are part of the equational theory induced by the input equations. Correctness of this proof can then be certified by CeTA [102], which verifies the output of mkbTT, checks joinability of all critical pairs as well as the input equations, and hence certifies convergence of the resulting TRS. Certifiable proofs produced by mkbTT can be obtained from the website.

8.1.2 Usage

Command Line Interface

The tool mkbTT is equipped with a simple command line interface. It expects an input problem in TPTP3 [100] or TPDB format, where in the latter case both the old textual and the newer XML format[2] are supported. If the input problem is in TPDB format then all its rewrite rules are actually treated as equations.

Several options allow to configure the tool. If the input problem is in the old textual format and the option `-po` is set then pre-orientation is activated, i.e., all strict rules in the (relative) input problem are preserved whereas all weak rules are considered as equations. Both a global and a local timeout in seconds can be specified using `-t` and `-T`. The termination prover is given as argument to the `-tp` option. Alternatively, if T$_\text{T}$T$_2$ is used internally a termination strategy can be supplied with the `-s` option. A selection strategy can be given with the option `-ss`, using the grammar described in Section 8.1.1.

Critical pair criteria can be applied by supplying `-cp` with appropriate arguments. For standard and ordered completion, the criteria of unblockedness, primality, connectedness and their combination are available by using the arguments `blocked`, `prime`, `connected`, and `all`, respectively. For normalized completion, primality and $\mathcal{S}$-reducibility can be used, where the latter is enabled with the argument `sreduce`.

For standard and ordered completion, isomorphism checks are to be specified via the option `-is` with optional arguments `rename`, `rename+`, `perm`, or `perm+`. With the suffix `+` we compare processes pairs in every iteration, otherwise checks are only performed when a process is split. By default mkbTT applies a heuristic to determine which isomorphism is potentially applicable.

Term indexing techniques used for rewriting and unification in standard and ordered completion may be selected with the options `-ix` and `-ui` together with one of `nv`, `pi`, `dt`, or—in the case of rewriting—`ct`, referring to naive search,

2 `http://www.termination-portal.org/`

path indexing, discrimination trees, and code trees respectively. For normalized completion, AC-discimination trees are used by default.

The option `-kp` expects a floating point value larger than 1 and allows to give a process filtering rate. For example, `-kp 1.2` deletes all processes that exceed the cost of the best process by 20%.

By default mkbTT applies standard completion. Ordered completion can be used with the option `-o`, and normalized completion with the option `-n`. In the latter case, a theory representation with respect to which completion is performed can be supplied via the option `-th` followed by a file containing the TRS $\mathcal{S}$. If no such argument is given then mkbTT detects an applicable theory automatically (currently ACU, groups and rings are supported, besides AC).

To control output, the flags `-ct`, `-st` and `-p` require mkbTT to print the completed system, statistics and a proof in case of success. Furthermore, the tool offers a checking mode where a file containing a rewrite system supplied via the option `-ch` is tested for termination, confluence and for allowing rewrite proofs for the the input equalities.

As an example, the call

```
mkbtt -t 600 -T 5 -tp aprove -cp prime WSW06_CGE2.trs
```

runs the tool on CGE_2 for at most 600 seconds using a script calling AProVE [42] for termination checks with a timeout of 5 seconds, and employs the critical pair criterion PCP.

Web Interface

Besides a command line interface, mkbTT can also be executed via a web interface. The screenshot in Figure 8.2 provides an impression. Various options may be configured by the user.

8.2 A Constrained Completion Tool

We furthermore improved and extended the implementation of constrained completion presented in [55] to the tool CC for standard and ordered completion based on the constrained equality framework. We outline how our framework was automated, give some implementation details and comment on the tool's usage.

8.2.1 Automation of the Constrained Equality Framework

For all of the procedures discussed in Chapter 7, our approach suggests a straightforward automation by repeatedly applying the respective reduction S to the initial constraint set. The remaining challenge is to find a suitable set of terminating TRSs $\mathfrak{R}(\mathcal{C})$ given a set of constrained equalities $\mathcal{C}$. To accomplish this task we exploit the fact that reduction orders can be encoded as satisfiability problems. Such encodings are well established nowadays [26, 37, 63, 114].

In our tool the TRSs $\mathfrak{R}(\mathcal{C})$ result from solving an optimization problem over SAT or SMT constraints in a similar way as in maximal completion, but by

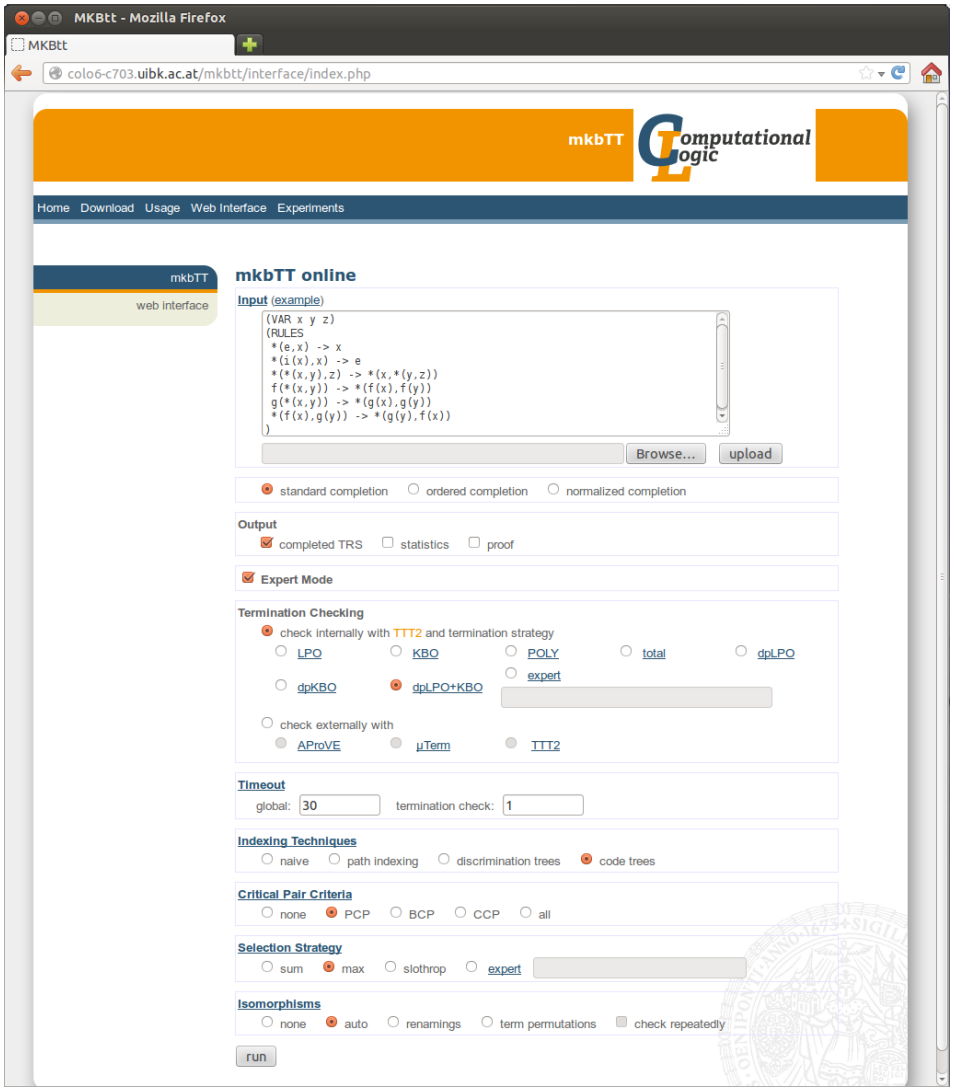


Figure 8.2: Web interface of mkbTT.

taking the termination constraints in the constrained equalities into account. We briefly sketch our approach to automation for the setting of standard completion.

We consider a signature $\mathcal{F}$. Any satisfiability encoding of a reduction order translates a TRS $\mathcal{R}$ over $\mathcal{F}$ into a formula $\phi_{\mathcal{R}}$ such that $\phi_{\mathcal{R}}$ is satisfiable by an assignment α if and only if $\ell \succ_\alpha r$ for all rules $\ell \to r \in \mathcal{R}$, where $\succ_\alpha$ is a reduction order on $\mathcal{T}(\mathcal{F}, \mathcal{V})$.

In the sequel we consider terms in $\mathcal{T}(\mathcal{F}, \mathcal{V})$ and an assignment α inducing a reduction order $\succ_\alpha$ on $\mathcal{T}(\mathcal{F}, \mathcal{V})$.

Definition 8.1. Let an *order constraint* be defined by the following BNF:

$$C ::= \ulcorner \ell > r \urcorner \mid x \mid \top \mid \bot \mid \neg\, C \mid C \vee C \mid C \wedge C$$

Let O be a mapping from termination constraints to order constraints which satisfies $O(\ell \to r) = \ulcorner \ell > r \urcorner$ and preserves the boolean structure.

For an assignment α and an order constraint ψ we write $\alpha \models \psi$ if α *satisfies* ψ. This relation is inductively defined such that $\alpha \models \ulcorner \ell > r \urcorner$ if and only if $\ell \succ_\alpha r$, and all boolean connectives have their usual semantics.

For a termination constraint C and an assignment α, let $R(C)$ denote the set of rewrite rules occurring in C. The TRS $\mathcal{R}_\alpha(C)$ is then defined as follows:

$$\mathcal{R}_\alpha(C) = \{\ell \to r \mid \ell \to r \in R(C) \text{ and } \ell \succ_\alpha r\}$$

Lemma 8.2 ([55])**.** *Let C be a termination constraint and α an assignment. If $\alpha \models O(C)$ then $\mathcal{R}_\alpha(C) \models C$.*

Let $\mathcal{C}$ be a CES. We assume that all termination constraints C occurring in $\mathcal{C}$ involve only terms in $\mathcal{T}(\mathcal{F}, \mathcal{V})$. We then write $\mathcal{R}_\alpha(\mathcal{C})$ for the TRS

$$\mathcal{R}_\alpha(\mathcal{C}) = \{\ell \to r \mid \langle \ell \simeq r, C\rangle \in \mathcal{C},\ \ell \succ_\alpha r \text{ and } \alpha \models O(C)\}$$

Lemma 8.3. *Let C be a termination constraint and α an assignment. If*

$$\alpha \models \bigwedge_{\langle s \approx t, C\rangle \in \mathcal{C}} \ulcorner s > t \urcorner \vee \ulcorner t > s \urcorner \vee \neg O(C)$$

then $\mathcal{C}{\downarrow_{\mathcal{R}_\alpha}}[\![\mathcal{R}_\alpha]\!] = \varnothing$.

Proof. Suppose $\alpha \models \ulcorner s > t \urcorner \vee \ulcorner t > s \urcorner \vee \neg O(C)$ for some $\langle s \approx t, C\rangle \in \mathcal{C}$. As $\langle s \approx t, C\rangle \in \mathcal{C}$ we have $\langle s \approx t, C \wedge \bigwedge \mathcal{R}_\alpha\rangle \in \mathcal{C}{\downarrow_{\mathcal{R}_\alpha}}$. If $\alpha \models \neg O(C)$ then $\alpha \not\models O(C \wedge \bigwedge \mathcal{R}_\alpha)$, so $s \approx t \notin \mathcal{C}{\downarrow_{\mathcal{R}_\alpha}}[\![\mathcal{R}_\alpha]\!]$. Otherwise, if $\alpha \models \ulcorner s > t \urcorner$ or $\alpha \models \ulcorner t > s \urcorner$ then $\mathcal{R}_\alpha(\mathcal{C}) \models s \to t$ or $\mathcal{R}_\alpha(\mathcal{C}) \models t \to s$ by Lemma 8.2, so $s \to t$ or $t \to s$ are in $\mathcal{R}_\alpha$ and hence $s \downarrow_{\mathcal{R}_\alpha} t$. Thus $s \approx t \notin \mathcal{C}{\downarrow_{\mathcal{R}_\alpha}}[\![\mathcal{R}_\alpha]\!]$. □

According to Theorem 7.13, the aim is to find a TRS $\mathcal{R}$ such that $\mathcal{C} = S^n_{\mathrm{KB}}(\mathcal{E}^\top)$ and $\mathcal{C}[\![\mathcal{R}]\!] = \varnothing$ for some $\mathcal{R} \in \mathfrak{R}(\mathcal{C})$. Recall that our transformers have the following general shape (cf. Definition 7.8):

$$S_{\mathcal{R}}(\mathcal{C}) = (\mathcal{C} \ominus \mathcal{R}) \ \cup\ \mathcal{C}{\downarrow_{\mathcal{R}}} \ \cup\ F(\mathcal{R})^\top{\downarrow_{\mathcal{R}}}$$
$$S(\mathcal{C}) = \bigcup_{\mathcal{R} \in \mathfrak{R}(\mathcal{C})} S_{\mathcal{R}}(\mathcal{C})$$

If $\mathcal{C} = S_{\mathcal{R}}(\mathcal{C}')$, we have $\mathcal{C}[\![\mathcal{R}]\!] = \varnothing$ if and only if $(\mathcal{C} \ominus \mathcal{R})[\![\mathcal{R}]\!] = \varnothing$, $\mathcal{C}{\downarrow_{\mathcal{R}}}[\![\mathcal{R}]\!] = \varnothing$, and $F(\mathcal{R})^{\top}{\downarrow_{\mathcal{R}}}[\![\mathcal{R}]\!] = \varnothing$. While $(\mathcal{C} \ominus \mathcal{R})[\![\mathcal{R}]\!] = \varnothing$ holds by Lemma 7.5 for any TRS $\mathcal{R}$, it is not clear how to determine in advance whether $F(\mathcal{R})^{\top}{\downarrow_{\mathcal{R}}}[\![\mathcal{R}]\!] = \varnothing$.

But Lemma 8.3 suggests a criterion for $\mathcal{C}{\downarrow_{\mathcal{R}}}[\![\mathcal{R}]\!] = \varnothing$ which can be used to choose TRSs $\mathcal{R} \in \mathfrak{R}(\mathcal{C})$: Given a CES $\mathcal{C}$ and previously chosen TRSs $\mathcal{R}_1, \ldots, \mathcal{R}_{n-1}$, we try to find an assignment α_n such that α_n maximizes the number of satisfied clauses in the following formula:

$$\bigwedge_{\langle s \approx t, C\rangle \in \mathcal{C}} \ulcorner s \succ t \urcorner \vee \ulcorner t \succ s \urcorner \vee \neg C \tag{8.1}$$

If such an assignment α_n is found, we set $\mathcal{R}_n = \mathcal{R}_{\alpha_n}$. In order to achieve progress, we moreover require $\alpha_n \neq \alpha_i$ for all $1 \leqslant i < n$.

In the settings of ordered completion, the TRSs in $\mathfrak{R}(\mathcal{C})$ are selected in exactly the same way.

8.2.2 Implementation

The tool CC is based on the completion tool presented in [55], but we improved the standard completion module and implemented ordered completion to compare with mkbTT. Sources and binaries are available from the website of mkbTT.

CC is written in OCaml and uses Yices [36] as backend to solve maximal satisfiability problems. As reduction orders LPO and KBO are available.

Experiments showed that it is beneficial to change the maximization goal (8.1) by introducing weight factors $n, m \in \mathbb{N}$:

$$\bigwedge_{\langle s \approx t, C\rangle \in \mathcal{C}} (\ulcorner s \succ t \urcorner \vee \ulcorner t \succ s \urcorner) \cdot n \vee C \cdot m$$

Note that we obtain maximal completion if $m = 0$. In our experiments, $n = 2$ and $m = 1$ turned out to be the most successful.

8.2.3 Usage

The tool CC offers a simple command line interface. Input problems are expected to be in the textual TPDB format, where all rewrite rules in the input TRS are regarded as equations.

By default standard completion is applied. The option `-o` switches to ordered completion. The applied reduction order can be controlled with the options `-lpo`, `-kbo`, and `-lpokbo`. In the latter case, CC tries to find TRSs that are compatible with LPO or KBO. Finally, `-K` allows to control the numer of TRSs to be contained in $\mathfrak{R}(\mathcal{C})$.

8.3 Experiments

We ran experiments on a server equipped with eight dual-core AMD Opteron® processors 885 running at a clock rate of 2.6GHz with 64GB of main memory.

8.3.1 Standard Completion

We will first focus experimental results obtained with mkbTT, reporting on the influence of various ingredients. The section on standard completion will be concluded by comparing mkbTT with CC and other completion tools.

Our test set comprises 101 problems collected from the literature. In the following paragraphs we summarize results obtained for the whole test set, and illustrate our conclusions with selected examples from that database. For this purpose, systems with prefix TPTP refer to theories underlying unit equality problems in TPTP 3.6.0 [100], prefix SK90 refers to [95, Section 3], and WSW06 refers to the Slothrop [107]. The prefixes BGK94 and C89 mark systems stemming from [23] and [25], respectively. The full experimental data can be obtained from the website. All experiments described in the following tables featured a timeout of 600 seconds. If a successful completion could not be achieved within that period this is marked by ∞, whereas $\perp$ indicates failure. If not stated otherwise, in all of the following experiments the following default settings of mkbTT were used: We interface TTT2 internally with termination strategy `dp-lpo-kbo` and a termination timeout of two seconds, apply selection strategy $s_{\texttt{max}}$, and use the critical pair criterion PCP. We use only renaming isomorphisms, controlled by the `auto` heuristic. As term indexing techniques code trees and discrimination trees allow to retrieve encompassments and unifiable terms, respectively.

Termination

Tables 8.1 and 8.2 show results obtained with different termination strategies when interfacing TTT2 internally. In Table 8.1, the first three columns refer to plain LPO, KBO (with weights of two bits), and their combination. The strategy `tkbo` refers to transfinite KBO, i.e., KBO with ordinal weights and subterm coefficients [113]. The strategy `poly` corresponds to linear polynomial interpretations with coefficients of two bits.

Table 8.2 reports on results obtained with more sophisticated termination strategies: `dp-kbo`, `dp-lpo`, and `dp-lpo-kbo` combine dependency pairs, a dependency graph approximation, the subterm criterion and some simple counting techniques with reduction pair processors using KBO, LPO, and both, respectively. In both tables, columns (1) show the time required for completion and columns (2) the percentage of time spent on termination.

The use of plain reduction orders such as LPO or KBO often results in comparatively fast completions (as e.g. in the cases of SK90-3.04 and SK90-3.27 for LPO and TPTP-GRP493-1 for KBO) because little time is spent on termination checks as can be seen from the bottom line. On the other hand, plain reduction orders have comparatively limited power when it comes to orienting equations, which can prevent success as in case of WS06-proofreduct or the CGE systems. Overall, this results in fewer completions than obtained with more complex strategies that offer a higher flexibility. We could not find a system where polynomial interpretations are beneficial, and the bottom line shows that termination checks are very time consuming in this case. The overall success rate

	lpo		kbo		lpo + kbo		tkbo		poly	
	(1)	(2)	(1)	(2)	(1)	(2)	(1)	(2)	(1)	(2)
C89-A_3	∞		234.0	17	245.2	21	460.5	56	∞	
SK90-3.04	0.7	52	∞		0.7	48	8.5	72	∞	
SK90-3.27	5.6	21	23.4	7	23.9	11	23.4	58	446.8	31
TPTP-GRP493-1	∞		37.5	15	39.7	19	∞		∞	
TPTP-GRP496-1	66.0	15	60.0	19	56.6	25	∞		∞	
WS06-proofreduct	∞		∞		∞		∞		∞	
WSW06-CGE_3	∞		∞		∞		∞		∞	
successes	69		67		74		57		41	
average time	13.8		19.1		17.3		21.3		20.7	
termination %	18		23		29		69		60	

Table 8.1: Different reduction orders.

turned out to be best with `dp-lpo-kbo`, supposedly since a combined strategy can cope best with problems where LPO is beneficial and problems where KBO is preferable. There are systems that can be completed with a plain reduction order but not with a strictly more powerful termination strategy employing dependency pairs, like TPTP-GRP496-1 using KBO. This illustrates that more termination power does not necessarily result in a higher chance for success because many possibilities for orienting equations also induce many processes, which can deteriorate performance significantly.

Selection Strategies

Table 8.3 demonstrates the crucial impact of selection strategies in mkbTT. Columns (1) give the time required for completion and columns (2) the number of control loop iterations (i.e., selected nodes). In line with previous observations from the theorem proving literature, we found that the selection strategy is critical for the success of a run. While for some systems such as SK90-3.07, TPTP-GRP490-1 or the CGE examples it is beneficial to use $s_{\texttt{max}}$, there are also systems like BGK94-M_8 which can only be solved using $s_{\texttt{sum}}$, and problems like SK90-3.22 for which a mere size/age ratio works best. It thus seems impossible to determine a single best strategy. Since overall $s_{\texttt{max}}$ could complete most systems and is fastest on average, it is used by default in a specialized and faster implementation.

Critical Pair Criteria

Table 8.4 compares results obtained with mkbTT using the primality criterion PCP, the connectedness criterion CCP and the mixed criterion MCP. Columns (1) list the time required for completion, columns (2) the number of redundant

	dp-lpo		dp-kbo		dp-lpo-kbo	
	(1)	(2)	(1)	(2)	(1)	(2)
C89-A_3	65.0	48	71.8	50	74.8	51
SK90-3.04	2.6	53	∞		2.7	54
SK90-3.27	34.8	29	27.9	21	29.7	27
TPTP-GRP493-1	∞		92.4	29	93.1	32
TPTP-GRP496-1	∞		60.0	32	63.4	37
WS06-proofreduct	174.9	93	179.5	92	182.4	92
WSW06-CGE_3	43.7	80	42.5	80	44.3	81
successes	76		78		80	
average time	13.2		17.5		18.1	
termination %	55		49		51	

Table 8.2: Termination strategies involving dependency pairs.

	$s_{\texttt{max}}$		$s_{\texttt{sum}}$		$s_{\texttt{slothrop}}$		$s_{\texttt{size/age}}$		$s_{\texttt{old}}$	
	(1)	(2)	(1)	(2)	(1)	(2)	(1)	(2)	(1)	(2)
BGK94-D_8	195.0	112	∞		370.4	161	352.2	352	158.8	149
BGK94-D_{16}	45.2	102	117.8	132	∞		∞		152.1	151
BGK94-M_8	∞		14.6	22	∞		∞		∞	
C89-A_2	28.1	108	165.6	270	∞		∞		138.8	109
SK90-3.07	86.2	161	∞		∞		∞		∞	
SK90-3.22	∞		∞		∞		25.0	99	∞	
TPTP-GRP490-1	130.9	218	∞		564.0	182	∞		∞	
WSW06-CGE_3	46.8	49	439.1	325	232.0	87	138.2	136	132.0	85
successes	79		68		68		71		68	
average time	22.6		31.2		61.8		52.8		35.7	

Table 8.3: Different selection strategies.

	none	PCP		CCP		MCP	
	(1)	(1)	(2)	(1)	(2)	(1)	(2)
BGK94-D_{12}	78.4	∞		23.6	24	23.1	38
C89-A_3	96.3	73.7	24	93.5	24	76.4	54
TPTP-GRP457-1	7.9	2.8	17	4.8	29	4.0	33
TPTP-GRP484-1	343.6	53.0	35	111.2	76	105.1	80
TPTP-GRP490-1	∞	130.8	67	80.0	155	90.4	174
TPTP-GRP496-1	80.3	77.4	60	65.3	100	64.8	104
WSW06-CGE_3	44.0	45.2	14	44.8	29	44.8	29
successes	79	78		80		80	
average time	21.6	23.5		18.7		18.4	
redundant CPs		834		1529		1605	
time to check		5.1		32.7		28.4	

Table 8.4: Different critical pair criteria.

critical pairs for the successful process and columns (3) the total number of created nodes.

We found a single system, TPTP-GRP490-1, which could only be completed when using PCP, CCP or MCP. For a number of systems the use of critical pair criteria results in a considerably smaller number of nodes and consequently some speedup, as in the cases of C89-A_3, TPTP-GRP457-1 or TPTP-GRP496-1. This is also reflected in the reduced average time for completion with CCP and MCP. However, there are also examples such as BGK94-D_{12} which can no longer be completed when using PCP (although CCP and MCP work), and examples such as TPTP-GRP484-1 where a less powerful criterion results in less control loop iterations and thus, a shorter completion time. In these cases the selection strategy $s_{\texttt{max}}$ seems to be influenced by the critical pair criterion in an unfortunate way: the effect of critical pair criteria for a certain system was generally found to depend on the selection strategy. When comparing the three criteria, it turns out that PCP detects the least number of critical pairs, but performs redundancy checks very fast (see the bottom line). When summing up all critical pairs filtered out for successful processes, CCP is twice as effective as PCP. The criterion BCP is a little less effective than PCP, relevant results can be obtained from the website. Overall MCP turned out to be most beneficial.

Term Indexing

Table 8.5 compares the term indexing techniques implemented in mkbTT to retrieve variants and encompassments. Here `nv` abbreviates naive filtering of the node database, `pi` refers to path indexing, `dt` refers to discrimination trees and `ct` to code trees. Columns (1) list the time required for completion while columns (2) and (3) give the percentage of time spent on retrieval and rewrite

	nv			pi			dt			ct		
	(1)	(2)	(3)	(1)	(2)	(3)	(1)	(2)	(3)	(1)	(2)	(3)
BGK94-D_8	173.9	6	19	180.8	3	21	160.1	1	15	151.8	1	13
C89-A_2	29.3	6	17	29.6	4	16	27.5	1	12	25.5	1	11
SK90-3.07	48.6	13	33	50.8	6	33	44.4	2	26	40.8	1	24
TPTP-GRP481-1	40.7	12	36	42.0	7	42	34.8	2	33	32.9	1	28
successes	79			79			79			79		
average time	21.1			21.1			19.0			17.9		
time/retrieval	104.8			51.5			11.5			6.8		
time/maintenance	0.8			1.2			0.8			0.5		

Table 8.5: Different term indexing techniques.

operations, respectively. While all indexing techniques allow to complete the same number of systems, the time consumed by retrieval operations can be reduced significantly when using discrimination trees or code trees. Table 8.5 singles out some examples where the gain is especially significant. When comparing the time required for rewrite steps, discrimination trees fall back behind code trees since the retrieved candidate nodes still have to be checked for subsuming the query term. This is not required when using a technique achieving perfect filtering such as code trees.

The bottom lines sum up indexing-related computation times over the whole database. It turns out that the retrieval time can be reduced by more than 90% using discrimination trees or code trees. As expected, maintenance operations such as insertion and removal consume hardly any time.

Concerning the retrieval of unifiable terms in deduce operations, the use of term indexing techniques turned out to be less influential. Compared to naive filtering, discrimination trees decrease the average share of time spent on retrieval from 1% to 0.3%.

Isomorphisms

Isomorphism checks can be performed either only on process splits, or repeatedly for every process pair. Table 8.6 compares both possibilities for renamings (`ren`) and argument permutations (`ap`) with the setting where no isomorphism checks are used (a `+` indicates repeated checks). The setting `auto` refers to mkbTT's default strategy, which determines at the beginning of a completion run whether the initial equations allow for a nontrivial renaming or argument permutation. In this case repeated checks are performed throughout the deduction. Columns (1) give the time required for completion in seconds, and columns (2) the number of processes emerging in the course of a run.

Renaming checks, especially when performed repeatedly, turned out to be useful for a number of problems, in particular the CGE systems. Also for some string systems like SK90-3.28 and SK90-3.29, and TPTP-GRP011-4 the number

	none		ren		ren+		ap		auto	
	(1)	(2)	(1)	(2)	(1)	(2)	(1)	(2)	(1)	(2)
C89-A_3	64.8	174	65.6	174	77.1	161	65.4	174	77.4	161
SK90-3.02	0.1	7	0.1	7	0.1	7	0.1	3	0.1	7
SK90-3.28	101.7	499	102.8	499	157.6	385	102.4	499	156.3	385
SK90-3.29	3.5	128	2.6	97	3.0	73	3.5	128	3.0	73
TPTP-GRP011-4	3.0	13	1.8	6	1.8	6	3.0	13	1.8	6
WSW06-CGE_2	37.7	50	37.6	50	7.0	11	37.5	50	7.0	11
WSW06-CGE_3	175.1	163	176.6	163	46.0	35	176.8	163	46.0	35
successes	80		80		80		80		80	
average time	18.3		18.3		23.1		18.3		17.8	

Table 8.6: Different isomorphisms.

of processes could be reduced, although this does not always result in faster completions due to the time required for checking. Argument permutations were useful for just two small systems in the benchmark set, one of which is SK90-3.02 where the number of processes could be halved, and repeated checks did not turn out to be useful at all.

On the other hand, especially repeated checks for isomorphisms can be costly if no isomorphic process pairs appear. This is for example the case for SK90-3.28 and the CGE systems when used with argument permutations. Overall the `auto` setting prevails concerning number of successes and average time, although the heuristic does not always go for the best choice.

Novel Completions

While Slothrop was the first completion tool to handle CGE_2 (in more than 200 seconds), mkbTT can also complete the systems CGE_3, CGE_4 and CGE_5 describing the theory of 3, 4 and 5 commuting group endomorphisms within 18, 145 and 35796 seconds, respectively (when using repeated isomorphism checks and decreasing the number of processes by using the `-kp` option). Our tool also produced the first convergent TRS for the proof reduction system WS06-proofreduct presented in [106].

Comparison with Other Tools

We now compare mkbTT with Maxcomp [56], Slothrop [107], and KBCV [98]. To this end we used the extension of our previous test set, which is available from the Maxcomp website and comprises 115 problems. In the following tables, the prefix KH11 marks systems stemming from [56].

Table 8.7 compares mkbTT with other implementations of completion with termination tools, namely Slothrop and KBCV. The table lists the time required for a successful completion in seconds. The last two lines give the number

	mkbTT	Slothrop	KBCV
BGK94-M_{12}	∞	38.8	6.0
SK90-3.26	∞	∞	20.9
SK90-3.28	223.8	436.6	∞
TPTP-GRP454-1	9.6	∞	6.2
WS06-proofreduct	237.9	208.2	∞
WSW06-equiv-proofs	7.3	33.5	∞
successes	87	76	87
average time	40.1	28.2	11.1

Table 8.7: Comparing mkbTT with Slothrop and KBCV.

	mkbTT			Maxcomp		CC	
	standard	LPO	KBO	LPO	KBO	LPO	KBO
BGK94-D_{10}	123.4	97.2	69.2	0.6	∞	2.2	∞
BGK94-D_{16}	128.0	304.8	53.3	34.0	∞	3.6	∞
OKW95-dt1	3.1	2.0	∞	40.8	∞	0.03	∞
SK90-3.22	∞	∞	∞	3.2	5.7	∞	1.6
WS06-proofreduct	237.9	∞	∞	$\perp$	∞	∞	∞
WSW06-equiv-proofs	7.3	$\perp$	$\perp$	∞	∞	∞	∞
successes	87	75	72	86	69	84	56
average time	40.1	19.1	30.2	3.6	8.6	1.0	3.6

Table 8.8: Comparison of mkbTT, Maxcomp, and CC.

of successes and the average time required to compute a convergent system, respectively. Note that for this comparison the Slothrop code was incorporated into mkbTT such that the same termination backend can be used. Overall, mkbTT solves about 15% more problems than Slothrop. Nevertheless, due to different selection strategies which are favorable for different problems there are also examples where Slothrop can produce a convergent system but our tool (with its default strategy) cannot, such as the system BGK94-M_{12}. The tool KBCV is based upon completion with termination tools as presented in [107], but does not feature multi-completion. It is more successful than mkbTT for some problems such as BGK94-M_{12} where mkbTT has to keep track of many processes. On the other hand, it does, e.g., not succeed in case of the CGE systems where mkbTT gains efficiency from the use of isomorphisms.

Table 8.8 compares with Maxcomp [56] and CC. Even with its more complex standard termination strategy, mkbTT can complete only one problem more than Maxcomp, although Maxcomp of course fails on problems like WSW06-equiv-proofs, WS06-proofreduct, or also the CGE problems which cannot be com-

	mkbTT	Maxcomp with LPO	Maxcomp with KBO
successes	1109	821	812

Table 8.9: Comparison of mkbTT and Maxcomp on a subset of TPDB.

pleted using plain LPO or KBO. Also, Maxcomp is decidedly more efficient. The tool CC yields similar results as Maxcomp, although it turned out to be a bit less efficient, lagging two successes behind.

The difference between mkbTT and Maxcomp grows when a benchmark set requiring more sophisticated termination techniques is considered, as shown in Table 8.9. Here the 3061 problems in TPDB version 7 were considered which are not already confluent and could be completed by at least one of the tools within 600 seconds.

8.3.2 Ordered Completion

As testbed we used 138 equational systems which comprise the 115 systems used for standard completion, plus all theories underlying TPTP [100] unit equality problems, plus examples collected from the literature [6, 44, 78]. All experiments described in the following tables featured a timeout of 600 seconds. If a successful completion could not be achieved within that period this is marked by ∞, whereas $\perp$ indicates failure. If not stated otherwise we used the following default settings of mkbTT: We interface $\mathsf{T_TT_2}$ internally with a termination timeout of two seconds and strategies which guarantee total termination. Apart from the termination strategy we use the same default settings as for standard completion.

Table 8.10 gives an overview of results obtained with different termination strategies. The first three strategies apply plain LPO and KBO (with weights of two bits), and transfinite KBO. Besides plain reduction orders, we also experimented with termination strategies combining multiple techniques. The strategy `lpo + kbo` iteratively removes rules using LPO and KBO in parallel, and `total` additionally applies linear polynomial interpretations. Columns (1) show the time required for completion and columns (2) the percentage of time spent on termination.

We briefly comment on some issues that seem noteworthy. Again some problems such as TPTP-GRP445-1 cannot be completed with LPO, but any strategy involving KBO works. For others such as KH11-fib KBO fails (or requires much more time, as in KH11-rl-theory) but any strategy with LPO works. The combined strategies are obviously more likely to succeed in both cases, and are hence most powerful. Nevertheless there are also problems like TPTP-GRP487-1 where LPO and KBO succeed, but `total` does not, presumably since the combined strategy gives rise to an enlarged search space resulting in an increased number of processes and hence worse performance. One problem (Example 5.44) could only be solved with the total strategy and `tkbo`, whereas for TPTP-GRP452-1 only `tkbo` was successful. As was to be expected, these total termination

	lpo		kbo		tkbo		lpo + kbo		total	
	(1)	(2)	(1)	(2)	(1)	(2)	(1)	(2)	(1)	(2)
KH11-fib	1.8	86	∞		∞		2.6	90	15.7	98
KH11-rl-theory	4.3	16	244.9	2	293.2	17	4.5	51	10.5	79
TPTP-GRP445-1	∞		5.8	14	11.4	55	5.5	28	11.7	67
TPTP-GRP452-1	∞		∞		192.1	30	∞		∞	
TPTP-GRP487-1	59.5	7	80.7	24	∞		95.0	35	∞	
Example 5.44	∞		∞		0.2	64	∞		0.1	
WSW06-CGE$_3$	∞		∞		∞		∞		∞	
successes	89		88		81		96		90	
average time	13.4		20.9		17.1		22.4		27.1	
termination %	12		16		57		30		83	

Table 8.10: Ordered completion using different termination strategies.

	mkbTT		CC	
	lpo	kbo	lpo	kbo
successes	89	88	55	47
average time	13.4	18.5	2	6.6

Table 8.11: Ordered completion with mkbTT and CC.

techniques are too weak to complete problems such as WSW06-CGE$_3$.

Table 8.12 shows theorem proving results obtained with mkbTT. Examples stem from the unit equality division of TPTP 3.6.0 [100]. The test set e consists of 215 problems rated *easy*, d contains 565 problems classified as *difficult*. The columns list (1) the number of successes, (2) the average time for a successful run in seconds (given a timeout of 600 seconds), and (3) the percentage of time spent on termination checks. Both Waldmeister [72] and E [92] solve about 200 problems in e and more than 400 of the d set.

	total			kbo			lpo			dp+lpo		
	(1)	(2)	(3)	(1)	(2)	(3)	(1)	(2)	(3)	(1)	(2)	(3)
e	149	43.9	82	163	16.6	8	164	24.3	14	138	49.9	80
d	116	66.0	64	148	64.8	4	152	50.6	6	121	55.0	17

Table 8.12: Performance of mkbTT on TPTP UEQ problems.

	mkbTT						C*i*ME
	AC		AG		auto		
	(1)	(2)	(1)	(2)	(1)	(2)	(1)
G94-abelian groups (AG)	1.6	77	0.1	5	0.1	5	0.05
AG + homomorphism	181.7	928	4.8	104	4.8	104	0.05
LS96-G0	1.9	82	0.1	8	0.1	8	?
LS96-G1	∞		12.4	49	12.5	49	?
G94-arithmetic	14.9	503	–		13.8	483	?
G94-AC-ring with unit	22.9	501	7.2	301	0.1	9	0.1
MU04-binary arithmetic	2.9	199	–		3.0	185	?
MU04-ternary arithmetic	18.1	816	–		17.3	781	?
Example 6.47	0.3	26	–		0.3	26	?
Example 6.59	∞		15.4	486	15.2	486	?
Example 6.60	∞		216.7	457	145.1	400	?
G94-semiring	3.3	209	–		3.5	193	0.1
K00-sum	1.4	4	–		1.4	4	?
completed systems	10		7		13		4

Table 8.13: Normalized completion with mkbTT.

8.3.3 Normalized Completion

Normalized completion experiments with mkbTT were run on problems collected from the literature. The prefixes attached to problems indicate their source: G94 refers to [40] and MU04 refers to [77], LS96 is associated with finite group representations in [71], and K00 refers to [64].

In Table 8.13 we compare mkbTT results with C*i*ME [28]. In the first two settings for mkbTT, the theory $\mathcal{T}$ was chosen to be AC and the theory of abelian groups (AG), respectively, whereas automatic theory detection was applied for the last setting. Termination checks were done with MuTerm [1], and the primality critical pair criterion was used. The global timeout and the timeout for each termination check were set to 300 and 2 seconds, respectively. Columns (1) list the total time in seconds while columns (2) give the number of nodes created during the run. The symbol ∞ marks a timeout, and – indicates that the theory is not applicable. In line with [75], we observed that completion with respect to larger theories $\mathcal{T}$ is typically more efficient, although completion modulo ACU yields only slight improvements compared to AC. As expected, C*i*ME is much faster if an appropriate reduction order is supplied as input. But as already mentioned, such a reduction order is hard to determine in advance, and in some cases no usable AC-RPO or polynomial interpretation exists. This is e.g. the case for commuting group actions (cf. Example 6.59, where mkbTT is able to find an ACU-convergent system in a bit more than one hour. In case of a system decribing commuting ring homomorphisms (cf. Example 6.59), our tool succeeds using normalized completion modulo group theory/ring theory in 216.7/145.1 seconds producing 457/400 nodes, respectively.

Concerning critical pair criteria, we found that the primality criterion de-

creased the total number of nodes by nearly 40%, which reduces the computation time by about 25%. $\mathcal{S}$-reducibility does not filter out any critical pairs if completion modulo ACU is performed. For normalized completion modulo group theory, very few redundant critical pairs are detected. The connectedness criterion was found to be comparatively expensive, and also the combined criterion could not achieve the same performance gain as the simpler primality criterion due to the additional effort of testing the criterion.

Chapter 9

Conclusion

This thesis explored fully automatic procedures for variants of Knuth-Bendix completion. To this end two complementary approaches were discussed:

(1) *Multi-completion with termination tools* is based upon a classical inference system but develops a reduction order *en route* using a termination prover. It keeps track of multiple branches of the search tree in parallel and shares inferences to gain efficiency.

(2) Completion-like procedures based on the *constrained equality framework* phrase a completion process as a maximal satisfiability problem.

We summarize our findings and contributions for each of the discussed completion methods.

For standard completion, a combined approach of multi-completion and the use of termination tools was presented in Chapter 4. Isomorphisms were proposed as a means to detect superfluous processes and hence, restrict the search tree, and critical pair criteria as a means to confine the number of equational consequences. Multi-completion with termination tools was implemented in the tool mkbTT. Our experiments (see Chapter 8) proved mkbTT to be competitive with other state-of-the-art tools, and gave rise to several novel completions. We also evaluated the usefulness of isomorphisms, critical pair criteria, term indexing techniques, and selection strategies.

A completion procedure based on the constrained equality framework was recalled in Section 7.2. Although already presented in [55], the implementation in the tool CC was considerably improved. When comparing mkbTT with this enhanced version of CC both tools could handle about equally many problems. Problems which require termination power beyond plain reduction orders could only be handled by mkbTT, since CC is inherently limited to reduction orders for which orientability can be encoded as a satisfiability problem. But CC turned out to be more efficient for problems where mkbTT runs into a timeout due to an unfortunate selection sequence or too many possibilities for orientation. Judging from our test set, we conclude that for many common input problems standard reduction orders are actually sufficient.

Ordered completion was discussed in Chapter 5, and its well-known inference system slightly simplified for finite runs. Critical pair criteria were shown sound for both settings. Refutational theorem proving results were given as well. We then discussed how termination tools—instead of fixed reduction orders—can

be used in ordered completion procedures. We showed that, in contrast to standard completion, only termination techniques guaranteeing total termination can be used. Hence the wide variety of termination techniques offered by modern termination tools can hardly be exploited. Moreover, extended critical pairs need to be over-approximated. We also covered ordered multi-completion with termination tools. This approach was implemented in our tool mkbTT as well, which thus, to the best of our knowledge, constitutes the first ordered completion tool that does not require the input of a suitable reduction order.

In Section 7.3 we developed an ordered completion procedure based on the constrained equality framework, which is also covered by CC. Our experiments comparing mkbTT with CC showed that CC behaves much worse than mkbTT in the setting of ordered completion. One reason for this difference lies in the fact that CC includes all equations in critical pair computations whereas mkbTT includes only unorientable equations. For refutational theorem proving, mkbTT turned out to lag behind modern equational theorem provers such as Waldmeister or E. We conclude that for ordered completion the reduction order has less impact than in standard completion because ordered completion cannot fail.

Normalized completion—being the most recent and widely applicable approach for completion modulo theories—was considered in Chapter 6. We revisited the entire underlying theory and could improve the central ingredient of a normalizing pair. We gave detailed completeness and uniqueness proofs, and verified soundness of critical pair criteria in the setting of normalized completion. Moreover, correctness of a slightly simplified inference system for finite runs was proven. We also presented normalized completion and normalized multi-completion with AC-termination tools. A respective implementation was added to mkbTT, which hence constitutes the first normalized completion tool that does not require a reduction order as input. In our experiments we compared mkbTT with C*i*ME and could produce novel completions for several input problems concerned with algebraic structures. In Section 7.4 we also formalized a normalized completion procedure based on the constrained equality framework.

As future work, it could be worth investigating whether further equational reasoning techniques may benefit from the use of termination techniques and/or multi-completion, e.g., abstract congruence closure methods, different paramodulation calculi and inductive theorem proving procedures.

However, for abstract congruence closure the gain might be limited as a favorable reduction order can always be determined in advance. Concerning paramodulation, it seems worth noting that, as for ordered completion, applicable termination techniques are limited to total termination. Moreover, the order plays a less critical role than, e.g., in standard completion as a wrong choice does not cause failure. For rewriting induction, the use of termination tools [3] and a multi-completion like approach tracking multiple contexts [89] were already investigated.

Publications

For completeness all publications emerging from my PhD studies are collected in the following list (in order of appearance).

- H. Sato, S. Winkler, M. Kurihara, and A. Middeldorp. Multi-completion with Termination Tools (System Description). In *Proc. of the 4th International Joint Conference on Automated Reasoning (IJCAR 2008)*, volume 5195 of *LNCS*, pages 306–312, 2008.
- C. Sternagel, R. Thiemann, S. Winkler, and H. Zankl. CeTA—A Tool for Certified Termination Analysis. In *Proc. of the 10th International Workshop on Termination (WST 2009)*, pages 84–87, 2009.
- H. Sato, S. Winkler, M. Kurihara, and A. Middeldorp. Constraint-Based Multi-Completion Procedures for Term Rewriting Systems. In *IEICE Transactions on Information and Systems* E92-D (2), pages 220–234, 2009
- S. Winkler, H. Sato, A. Middeldorp, and M. Kurihara. Optimizing mkbTT (System Description). In *Proc. of the 21st International Conference on Rewriting Techniques and Applications (RTA 2010)*, LIPIcs 13, pages 373–384, 2010.
- S. Winkler and A. Middeldorp. Termination Tools in Ordered Completion. In *Proc. of the 5th International Joint Conference on Automated Reasoning (IJCAR 2010)*, volume 6173 of *LNCS*, pages 518–532, 2010.
- S. Winkler and A. Middeldorp. AC Completion with Termination Tools. In *Proc. of the 23rd International Conference on Automated Deduction (CADE-23)*, volume 6803 of *LNCS*, pages 492–498, 2011
- H. Zankl, S. Winkler, and A. Middeldorp. Automating Ordinal Interpretations. In *Proc. of the 12th International Workshop on Termination (WST 2012)*, pages 94–98, 2012.
- S. Winkler, H. Zankl, and A. Middeldorp. Ordinals and Knuth-Bendix Orders. In *Proc. of the 18th International Conference on Logic for Programming, Artificial Intelligence and Reasoning (LPAR 2012)*, volume 7180 of *LNCS*, pages 420–434, 2012.
- S. Winkler, H. Sato, A. Middeldorp, and M. Kurihara. Multi-Completion with Termination Tools. *Journal of Automated Reasoning*, 2013, to appear.

Bibliography

[1] B. Alarcón, R. Gutiérrez, J. Iborra, and S. Lucas. Proving termination of context-sensitive rewriting with MU-TERM. In *6th PROLE*, volume 188 of *ENTCS*, pages 105–115, 2007.

[2] B. Alarcón, S. Lucas, and J. Meseguer. A dependency pair framework for $A \vee C$-termination. In *Proc. 8th International Workshop on Rewriting Logic and its Applications (WRLA 2010)*, volume 6381 of *LNCS*, pages 35–51, 2010.

[3] T. Aoto. Rewriting induction using termination checker. In *Proc. of the 24th Annual Meeting of the Japan Society for Software Science and Technology*, pages 61–74, 2007.

[4] F. Baader and T. Nipkow. *Term Rewriting and All That*. Cambridge University Press, 1998.

[5] F. Baader and W. Snyder. Unification theory. In *Handbook of Automated Reasoning*, pages 445–532. Elsevier, 2001.

[6] L. Bachmair. *Canonical Equational Proofs*. Progress in Theoretical Computer Science. Birkhäuser, 1991.

[7] L. Bachmair. Associative-commutative reduction orderings. *Information Processing Letters*, 43(1):21–27, 1992.

[8] L. Bachmair, T. Chen, and I. Ramakrishnan. Associative-commutative discrimination nets. In *Proc. of the 5th International Joint Conference on Theory and Practice of Software Development (TAPSOFT 1993)*, volume 668 of *LNCS*, pages 61–74, 1993.

[9] L. Bachmair and N. Dershowitz. Critical pair criteria for completion. *Journal of Symbolic Computation*, 6(1):1–18, 1988.

[10] L. Bachmair and N. Dershowitz. Completion for rewriting modulo a congruence. *Theoretical Computer Science*, 67(2-3):173–201, 1989.

[11] L. Bachmair and N. Dershowitz. Equational inference, canonical proofs, and proof orderings. *Journal of the ACM*, 41(2):236–276, 1994.

[12] L. Bachmair, N. Dershowitz, and J. Hsiang. Orderings for equational proofs. In *Proc. of the 1st Annual IEEE Symposium on Logic in Computer Science (LICS 1986)*, pages 346–357. IEEE Computer Society, 1986.

[13] L. Bachmair, N. Dershowitz, and D. A. Plaisted. Completion without failure. In H. Aït Kaci and M. Nivat, editors, *Resolution of Equations in Algebraic Structures*, volume 2: Rewriting Techniques of *Progress in Theoretical Computer Science*, pages 1–30. Academic Press, 1989.

[14] L. Bachmair and H. Ganzinger. Rewrite-based equational theorem proving with selection and simplification. *Journal of Logic and Computation*, 4(3):217–247, 1994.

[15] L. Bachmair, H. Ganzinger, C. Lynch, and W. Snyder. Basic paramodulation. *Information and Computation*, 121(2):172–192, 1995.

[16] L. Bachmair and D. Plaisted. Termination orderings for associative-commutative rewriting systems. *Journal of Symbolic Computation*, 1(4):329–349, 1985.

[17] L. Bachmair, A. Tiwari, and L. Vigneron. Abstract congruence closure. *Journal of Automated Reasoning*, 31(2):129–168, 2003.

[18] T. Baird, G. Peterson, and R. Wilkerson. Complete sets of reductions modulo associativity, commutativity and identity. In *Proc. of the 7th International Conference on Rewriting Techniques and Applications (RTA 1996)*, volume 355 of *LNCS*, pages 29–44, 1989.

[19] A. Ben Cherifa and P. Lescanne. Termination of rewriting systems by polynomial interpretations and its implementation. *Science of Computer Programming*, 9(2):137–159, 1987.

[20] M. Bofill, G. Godoy, R. Nieuwenhuis, and A. Rubio. Paramodulation and Knuth–Bendix completion with nontotal and nonmonotonic orderings. *Journal of Automated Reasoning*, 30(1):99–120, 2003.

[21] R. Bryant. Symbolic boolean manipulation with ordered binary-decision diagrams. *ACM Computing Surveys*, 24(3):293–318, 1992.

[22] B. Buchberger. Ein algorithmisches Kriterium fr die Lösbarkeit eines algebraischen Gleichungssystems. *Aequationes mathematicea*, 4(3):374–383, 1970.

[23] R. Bündgen, M. Göbel, and W. Küchlin. A fine-grained parallel completion procedure. In *Proc. of the International Symposium on Symbolic and Algebraic Computation (ISSAC 1994)*, pages 269–277, 1994.

[24] G. Butler and D. Lankford. Experiments with computer implementations of procedures which often derive decision algorithms for the word problem in abstract algebras. Technical Report MTP-7, Louisiana Technical University, 1980.

[25] J. Christian. Fast Knuth-Bendix completion: Summary. In *Proc. of the 3rd International Conference on Rewriting Techniques and Applications (RTA 1992)*, volume 355 of *LNCS*, pages 551–555, 1989.

[26] M. Codish, V. Lagoon, and P. Stuckey. Solving partial order constraints for LPO termination. *Journal of Satisfiability, Boolean Modeling and Computation*, 5:193–215, 2008.

[27] H. Comon, P. Narendran, R. Nieuwenhuis, and M. Rusinowitch. Deciding the confluence of ordered term rewrite systems. *ACM Transactions on Computational Logic*, 4(1):33–55, 2003.

[28] E. Contejean and C. Marché. C*i*ME: Complet*i*on modulo *E*. In *Proc. of the 7th International Conference on Rewriting Techniques and Applications (RTA 1996)*, volume 1103 of *LNCS*, pages 416–419, 1996.

[29] J. Denzinger, M. Kronenburg, and S. Schulz. DISCOUNT – a distributed and learning equational prover. *Journal of Automated Reasoning*, 18(1):189–198, 1997.

[30] N. Dershowitz. Orderings for term rewriting systems. *Theoretical Computer Science*, 17(3):279–301, 1982.

[31] N. Dershowitz. Termination of rewriting. *Journal of Symbolic Computation*, 3(1,2):69–116, 1987.

[32] N. Dershowitz, J. Hsiang, A. Josephson, and D. Plaisted. Associative-commutative rewriting. In *Proc. of the 8th International Joint Conference on Artificial Intelligence (IJCAI 1983)*, pages 940–944, 1983.

[33] N. Dershowitz, L. Marcus, and A. Tarlecki. Existence, uniqueness, and construction of rewrite systems. *SIAM Journal of Computing*, 17:629–639, 1988.

[34] N. Dershowitz and D. Plaisted. Rewriting. In *Handbook of Automated Reasoning*, pages 535–610. Elsevier, 2001.

[35] J. Dick, J. Kalmus, and U. Martin. Automating the Knuth-Bendix ordering. *Acta Informatica*, 28:95–119, 1990.

[36] B. Dutertre and L. D. Moura. A fast linear-arithmetic solver for DPLL(T). In *Proc. of the 18th International Conference on Computer Aided Verification (CAV 2006)*, volume 4144 of *LNCS*, pages 81–94, 2006.

[37] J. Endrullis, J. Waldmann, and H. Zantema. Matrix interpretations for proving termination of term rewriting. *Journal of Automated Reasoning*, 40(2-3):195–220, 2008.

[38] M. Fay. First order unification in equational theories. In *Proc. 4th Workshop on Automated Deduction (CADE-4)*, volume 87 of *LNCS*, pages 161–167, 1979.

[39] M. Ferreira and H. Zantema. Total termination of term rewriting. *Applicable Algebra in Engineering, Communication and Computing*, 7(2):133–162, 1996.

[40] W. Gehrke. Detailed catalogue of canonical term rewrite systems generated automatically. Technical report, RISC Linz, 1992.

[41] A. Geser. An improved general path order. *Applicable Algebra in Engineering, Communication and Computing*, 7(6):469–511, 1996.

[42] J. Giesl, P. Schneider-Kamp, and R. Thiemann. AProVE 1.2: Automatic termination proofs in the dependency pair framework. In *Proc. of the 3rd International Joint Conference on Automated Reasoning (IJCAR 2006)*, volume 4130 of *LNAI*, pages 281–286, 2006.

[43] G. Godoy and R. Nieuwenhuis. Paramodulation with built-in abelian groups. In *Proc. of the 15th Annual IEEE Symposium on Logic in Computer Science (LICS 2000)*, pages 413–424. IEEE Computer Society, 2000.

[44] J. Hsiang and M. Rusinowitch. On word problems in equational theories. Technical report, INRIA Lorraine, 1987. Unpublished manuscript.

[45] G. Huet. Confluent reductions: Abstract properties and applications to term rewriting systems. *Journal of the ACM*, 27(4):797–821, 1980.

[46] G. Huet. A complete proof of correctness of the Knuth-Bendix completion algorithm. *Journal of Computer and System Sciences*, 23(1):11–21, 1981.

[47] J.-P. Jouannaud. Confluent and coherent equational term rewriting systems: Application to proofs in abstract data types. In *Proc. of the 8th Colloquium on Trees in Algebra and Programming (CAAP 1983)*, volume 59 of *LNCS*, pages 269–283, 1983.

[48] J.-P. Jouannaud and H. Kirchner. Completion of a set of rules modulo a set of equations. *SIAM Journal of Computation*, 15(4):1155–1194, 1986.

[49] J.-P. Jouannaud and C. Marché. Termination and completion modulo associativity, commutativity and identity. *Theoretical Computer Science*, 104(1):29–51, 1992.

[50] S. Kamin and J. Lévy. Two generalizations of the recursive path ordering. Unpublished manuscript, University of Illinois, 1980.

[51] D. Kapur, D. Musser, and P. Narendran. Only prime superpositions need be considered in the Knuth-Bendix completion procedure. *Journal of Symbolic Computation*, 6(1):19–36, 1988.

[52] D. Kapur, P. Narendran, and F. Otto. On ground-confluence of term rewriting systems. *Information and Computation*, 86(1):14–31, 1990.

[53] D. Kapur, G. Sivakumar, and H. Zhang. A new method for proving termination of AC-rewrite systems. In *Proc. of the 10th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 1990)*, volume 472 of *LNCS*, pages 133–148, 1990.

[54] C. Kirchner and H. Kirchner. Constrained equational reasoning. In *Proc. of the International Symposium on Symbolic and Algebraic Computation (ISSAC 1989)*, pages 382–389, 1989.

[55] D. Klein. *Equational Reasoning and Completion.* PhD thesis, Japan Advanced Institute of Science and Technology, 2012.

[56] D. Klein and N. Hirokawa. Maximal completion (system description). In *Proc. of the 22nd International Conference on Rewriting Techniques and Applications (RTA 2011)*, volume 10 of *LIPIcs*, pages 71–80, 2011.

[57] D. Knuth and P. Bendix. Simple word problems in universal algebras. In J. Leech, editor, *Computational Problems in Abstract Algebra*, pages 263–297. Pergamon Press, 1970.

[58] H. Koike and Y. Toyama. Inductionless induction and rewriting induction. *Computer Software*, 17(6):1–12, 2000. In Japanese.

[59] K. Korovin and A. Voronkov. Orienting rewrite rules with the Knuth-Bendix order. *Information and Computation*, 183(2):165–186, 2003.

[60] M. Korp, C. Sternagel, H. Zankl, and A. Middeldorp. Tyrolean termination tool 2. In *Proc. of the 20th International Conference on Rewriting Techniques and Applications (RTA 2009)*, volume 5595 of *LNCS*, pages 295–304, 2009.

[61] W. Küchlin. A confluence criterion based on the generalised Newman lemma. In *Proc. of the 2nd European Conference on Computer Algebra (EUROCAL 1983)*, volume 204 of *LNCS*, pages 390–399, 1985.

[62] M. Kurihara and H. Kondo. Completion for multiple reduction orderings. *Journal of Automated Reasoning*, 23(1):25–42, 1999.

[63] M. Kurihara and H. Kondo. Efficient BDD encodings for partial order constraints with application to expert systems in software verification. In *Proc. of the 17th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE 2004)*, volume 3029 of *LNAI*, pages 827–837, 2004.

[64] K. Kusakari. *AC-Termination and Dependency Pairs of Term Rewriting Systems.* PhD thesis, Japan Advanced Institute of Science and Technology, 2000.

[65] D. Lankford and A. Ballantyne. Decision procedures for simple equational theories with associative-commutative axioms: Complete sets of associative-commutative reductions. Technical Report ATP-39, University of Texas, 1977.

[66] D. Lankford and A. Ballantyne. Decision procedures for simple equational theories with commutative axioms: Complete sets of commutative reductions. Technical Report ATP-35, University of Texas, 1977.

[67] D. Lankford and A. Ballantyne. Decision procedures for simple equational theories with permutative axioms: Canonical sets of permutative reductions. Technical Report ATP-37, University of Texas, 1977.

[68] P. Le Chenadec. *Canonical forms in finitely presented algebras*. Pitman, 1986.

[69] I. Lepper. Derivation lengths and order types of Knuth-Bendix orders. *Theoretical Computer Science*, 269(1-2):433–450, 2001.

[70] P. Lescanne. REVE: A rewrite rule laboratory. In *Proc. of the 4th International Symposium on Theoretical Aspects of Computer Science (STACS 1987)*, volume 247 of *LNCS*, pages 482–483, 1987.

[71] S. Linton and D. Shand. Some group theoretic examples with completion theorem provers. *Journal of Automated Reasoning*, 17(2):145–169, 1996.

[72] B. Löchner and T. Hillenbrand. A phytography of Waldmeister. *AI Communications*, 15(2,3):127–133, 2002.

[73] C. Marché. *Réécriture modulo une théorie présentée par un système convergent et décidabilité du problème du mot dans certaines classes de théories équationnelles*. PhD thesis, Université Paris-Sud, 1993.

[74] C. Marché. Normalised rewriting and normalised completion. In *Proc. of the 9th Annual IEEE Symposium on Logic in Computer Science (LICS 1994)*, pages 394–403. IEEE Computer Society, 1994.

[75] C. Marché. Normalized rewriting: An alternative to rewriting modulo a set of equations. *Journal of Symbolic Computation*, 21(3):253–288, 1996.

[76] C. Marché. Normalized rewriting: An unified view of Knuth-Bendix completion and Gröbner bases computation. In *Symbolic Rewriting Techniques*, volume 15 of *Progress in Computer Science and Applied Logic*, pages 193–208. Birkhäuser, 1998.

[77] C. Marché and X. Urbain. Modular and incremental proofs of AC-termination. *Journal of Symbolic Computation*, 38(1):873–897, 2004.

[78] U. Martin and T. Nipkow. Ordered rewriting and confluence. In *Proc. of the 10th International Conference on Automated Deduction (CADE-10)*, volume 449 of *LNCS*, pages 366–380, 1990.

[79] W. McCune. Experiments with discrimination-tree indexing and path indexing for term retrieval. *Journal of Automated Reasoning*, 9(2):147–167, 1992.

[80] Y. Métivier. About the rewriting systems produced by the Knuth-Bendix completion algorithm. *Information Processing Letters*, 16(1):31–34, 1983.

[81] A. Middeldorp and H. Zantema. Simple termination of rewrite systems. *Theoretical Computer Science*, 175(1):127–158, 1997.

[82] R. Nieuwenhuis and A. Rubio. Paramodulation-based theorem proving. In *Handbook of Automated Reasoning*, pages 371–443. Elsevier, 2001.

[83] G. Peterson and M. Stickel. Complete sets of reductions for some equational theories. *Journal of the ACM*, 28(2):233–264, 1981.

[84] D. Plaisted. Equational reasoning and term rewriting systems. In *Handbook of logic in artificial intelligence and logic programming*, volume 1, pages 274–364. Oxford University Press, 1993.

[85] D. Plaisted and A. Sattler-Klein. Proof lengths for equational completion. *Information and Computation*, 125(2):154–170, 1996.

[86] U. Reddy. Term rewriting induction. In *Proc. of the 10th International Conference on Automated Deduction (CADE-10)*, volume 449 of *LNCS*, pages 162–177, 1990.

[87] A. Rubio. A fully syntactic AC-RPO. *Information and Computation*, 178(2):515–533, 2002.

[88] A. Rubio and R. Nieuwenhuis. A total AC-compatible ordering based on RPO. *Theoretical Computer Science*, 142(2):209–227, 1995.

[89] H. Sato and M. Kurihara. Multi-context rewriting induction with termination checkers. *IEICE Transactions*, 93-D(5):942–952, 2010.

[90] H. Sato, S. Winkler, M. Kurihara, and A. Middeldorp. Multi-completion with termination tools (system description). In *Proc. of the 4th International Joint Conference on Automated Reasoning (IJCAR 2008)*, volume 5195 of *LNAI*, pages 306–312, 2008.

[91] A. Sattler-Klein. About changing the ordering during Knuth-Bendix completion. In *Proc. of the 11th International Symposium on Theoretical Aspects of Computer Science (STACS 1994)*, volume 775 of *LNCS*, pages 175–186, 1994.

[92] S. Schulz. The E Equational Theorem Prover, 2009. Available from `http:// www.eprover.org`.

[93] R. Sekar, I. V. Ramakrishnan, and A. Voronkov. Term indexing. In *Handbook of Automated Reasoning*, pages 1853–1964. Elsevier, 2001.

[94] J. Steinbach. AC-termination of rewrite systems: A modified Knuth-Bendix ordering. In *Proc. of the 2nd Conference on Algebraic and Logic Programming (ALP 1990)*, pages 372–386, 1990.

[95] J. Steinbach and U. Kühler. Check your ordering – termination proofs and open problems. Technical Report SR-90-25, Universität Kaiserslautern, 1990.

[96] J. Steinbach and M. Zehnter. Vademecum of polynomial orderings. Technical Report SR-90-03, Universität Kaiserslautern, 1990.

[97] T. Sternagel, R. Thiemann, H. Zankl, and C. Sternagel. Recording completion for finding and certifying proofs in equational logic. In *Proc. of the 1st International Workshop on Confluence (IWC 2012)*, pages 31–36, 2012.

[98] T. Sternagel and H. Zankl. KBCV - Knuth-Bendix completion visualizer. In *Proc. of the 5th International Joint Conference on Automated Reasoning (IJCAR 2012)*, volume 7364 of *LNCS*, pages 530–536, 2012.

[99] A. Stump and B. Löchner. Knuth-Bendix completion of theories of commuting group endomorphisms. *Information Processing Letters*, 98(5):195–198, 2006.

[100] G. Sutcliffe. The TPTP problem library and associated infrastructure: The FOF and CNF parts, v3.5.0. *Journal of Automated Reasoning*, 43(4):337–362, 2009.

[101] Terese. *Term Rewriting Systems*, volume 55 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2003.

[102] R. Thiemann and C. Sternagel. Certification of termination proofs using CeTA. In *Proc. of the 22nd International Conference on Theorem Proving in Higher-Order Logics*, volume 5674 of *LNCS*, pages 452–468, 2009.

[103] A. Voronkov. The anatomy of Vampire. *Journal of Automated Reasoning*, 15(2):237–265, 1995.

[104] A. Voronkov. Algorithms, datastructures, and other issues in efficient automated deduction. In *Proc. of the 1st International Joint Conference on Automated Reasoning (IJCAR 2001)*, volume 2083 of *LNCS*, pages 13–28, 2001.

[105] U. Waldmann. Cancellative abelian monoids and related structures in refutational theorem proving (parts I and II). *Journal of Symbolic Computation*, 33(6):777–861, 2002.

[106] I. Wehrman and A. Stump. Mining propositional simplification proofs for small validating clauses. In *Proc. of the 3rd Workshop on Pragmatics of Decision Procedures in Automated Reasoning (PDPAR 2005)*, volume 144 of *ENTCS*, pages 79–91, 2005.

[107] I. Wehrman, A. Stump, and E. Westbrook. Slothrop: Knuth-Bendix completion with a modern termination checker. In *Proc. of the 17th International Conference on Rewriting Techniques and Applications (RTA 2006)*, volume 4098 of *LNCS*, pages 287–296, 2006.

[108] A. Weiermann. Termination proofs for term rewriting systems by lexicographic path orderings imply multiply recursive derivation lengths. *Theoretical Computer Science*, 139(1-2):355–362, 1995.

[109] F. Winkler. Reducing the complexity of the Knuth-Bendix completion-algorithm: A "unification" of different approaches. In *Proc. of the 2nd European Conference on Computer Algebra (EUROCAL 1983)*, volume 204 of *LNCS*, pages 378–389, 1985.

[110] S. Winkler and A. Middeldorp. Termination tools in ordered completion. In *Proc. of the 6th International Joint Conference on Automated Reasoning (IJCAR 2010)*, volume 6173 of *LNAI*, pages 518–532, 2010.

[111] S. Winkler and A. Middeldorp. AC completion with termination tools. In *Proc. of the 23rd International Conference on Automated Deduction (CADE-23)*, volume 6803 of *LNAI*, pages 492–498, 2011.

[112] S. Winkler, H. Sato, A. Middeldorp, and M. Kurihara. Multi-completion with termination tools. *Journal of Automated Reasoning*, 2013. Accepted for publication. Available from the mkbTT website.

[113] S. Winkler, H. Zankl, and A. Middeldorp. Ordinals and Knuth-Bendix orders. In *Proc. of the 18th International Conference on Logic for Programming, Artificial Intelligence and Reasoning (LPAR 2012)*, volume 7180 of *LNCS*, pages 420–434, 2012.

[114] H. Zankl, N. Hirokawa, and A. Middeldorp. KBO orientability. *Journal of Automated Reasoning*, 43(2):173–201, 2009.

[115] H. Zankl and A. Middeldorp. Satisfying KBO constraints. In *Proc. of the 18th International Conference on Rewriting Techniques and Applications (RTA 2007)*, volume 4533 of *LNCS*, pages 389–403, 2007.

Index