In *Proc. of the* 4^{th} *International Conference on Principles in Knowledge Representation and Reasoning (KR94)*

A Computational Account for a Description Logic of Time and Action

Alessandro Artale* **and Enrico Franconi**
Knowledge Representation and Reasoning Lab.
IRST, I-38050 Povo TN, Italy
{artale|franconi}@irst.it

Abstract

A formal language for representing and reasoning about time and action is presented. We employ an action representation in the style of Allen, where an action is represented by describing the time course of events while the action occurs. In this sense, an action is defined by means of temporal constraints on the world states, which pertain to the action itself, and on other more elementary actions occurring over time. A distinction between action types and individual actions is supported by the formalism. Plans are seen as complex actions whose properties possibly change with time. The formal representation language used in this paper is a description logic, and it is provided with a well founded syntax, semantics and calculus. Algorithms for the subsumption and recognition tasks – forming the basis for action management – are provided.

1 INTRODUCTION

The goal of this work is to investigate a formal framework that permits dealing with time, actions and plans in a uniform way. As opposed to the most common approaches to modeling actions as state change – e.g., the formal models based on *situation calculus* [McCarthy and Hayes, 1969], the STRIPS-like planning systems [Lifschitz, 1987] – where actions are instantaneous and defined as functions from one state to another, we prefer to explicitly introduce the notion of time by admitting that actions take time, like in [Allen, 1991]. Allen builds a representation based on time, eliminating the notion of state as encoded in the STRIPS-like systems by means of *preconditions* – causes – and *postconditions* – effects. This formalism is not intended to capture any sense of causality, but it represents an action by describing the time course of events while the action occurs. Besides, unlike STRIPS-like systems, an action takes time: then, it is possible to define what is true while the action is occurring. Different actions can be concurrent or may overlap in time; effects of overlapping actions can be different from the sum of their individual effects; effects may not follow the action but more complex temporal relations may hold. Starting from a formal language able to express temporally related objects, actions are represented through temporal constraints on the world states, which pertain to the action itself, and on other more primitive actions occurring over time. With respect to [Allen, 1991], our formalism has a clear distinction between the language for expressing action types (the conceptual level) and the language for expressing individual actions (the assertional level). Plans are built by temporally relating action types in a compositional way using the temporal constructors available in the language. In this way, since the temporal relationships are proper operators of the basic language, the distinction between actions and plans disappears. As a matter of fact, we do not need distinct languages for objects and states representation, for time representation, for actions representation, and for plans representation.

The basic temporal language we propose is a *concept language* [Nebel, 1990], i.e. a description logic of the KL-ONE family[1], and it is inspired by the work of [Schmiedel, 1990]. The use of a concept language to represent directly action and plan descriptions allows us to exploit the ideas developed in the concept languages family, like procedures for subsumption, classification and instance recognition [Hollunder *et al.*, 1990; Nebel, 1990]. In this paper we present a calculus to check subsumption between actions types, and to recognize which type of action has taken place at a certain time interval from the observation of what is happening in the world. A plan taxonomy based on subsumption can be built, and it can play the role of a plan library to be used for plan retrieval and plan recognition tasks [Kautz, 1991].

Several temporal extensions of a concept language exist in the literature: Claudio Bettini in [Bettini, 1992] and [Bettini, 1993] proposes a variable-free extension

*Current address: Ladseb-CNR, I-35020 Padova PD, Italy

[1] Concept languages are also called Frame-Based Description Languages, Term Subsumption Languages, Terminological Logics, Taxonomic Logics or Description Logics.

with both existential and universal temporal quantification; [Devanbu and Litman, 1991] and [Weida and Litman, 1992] – and recently [Weida and Litman, 1994] – propose a loose hybrid integration between concept languages and respectively regular expressions and constraint networks; [Schmiedel, 1990] proposes a very expressive but undecidable language with variables and temporal quantifiers; [Schild, 1993] proposes the embedding of *point*-based tense operators in a propositionally closed concept language – his ideas have been applied in the BACK terminological representation system; [Lambrix and Rönnquist, 1993] study the combination of the temporal logic LITE, where the notion of object is revised from being an indivisible entity into being a temporal structure of *versions*, and a terminological logic. In [Song and Cohen, 1991], temporal constraints between actions and its decomposed subactions – in the context of hierarchical planning – are made explicit from the structure of the plan, in order to improve the results of plan recognition.

Our proposal reduces the expressivity of [Schmiedel, 1990] in the direction of [Weida and Litman, 1992]; in this way, we obtain a decidable logic with a sound and complete subsumption algorithm. However, while [Weida and Litman, 1992] use two different formalisms to represent actions and plans – a non temporal concept language for describing actions and a second formalism to compose plans by adding temporal information – we choose an extension of a description logic where time operators are available directly as term constructors. This view implies an integration of a temporal domain in the semantic structure where terms themselves are interpreted, giving the formal way both for a well-founded notion of subsumption and for proving soundness and completeness of the corresponding algorithm. Moreover, we are able to build *temporal structured* actions – as opposed to atomic actions – describing how the world state changes because of the occurrence of an action. In fact, our language allows for feature representation, as suggested in [Heinsohn *et al.*, 1992], in order to relate actions to states of the world.

The paper is organized as follows. The formal language is first introduced, by presenting its syntax and semantics at both the concept and individual levels. The subsumption and instance recognition problems are formally defined in this framework. Examples of application of the temporal language for action and plan representation and reasoning are presented in section 3. The calculus is finally briefly revealed, by looking first to a normal form for concept expressions, and then to the algorithms for subsumption and instance recognition.

2 THE TEMPORAL LANGUAGE

We introduce in this section the temporal language. Pursuing the ideas of [Schmiedel, 1990], an interval-based temporal extension of concept languages is investigated. A well founded syntax and semantics for the language is given and a formal definition of the subsumption and recognition reasoning tasks is devised.

C, D	$\rightarrow$	$E \mid$	(non-temporal concept)
		$C \sqcap D \mid$	(conjunction)
		$C \sqcup D \mid$	(disjunction)
		$C@X \mid$	(qualifier)
		$C[Y]@X \mid$	(substitutive qualifier)
		$\diamond(X^{+})\ Tc^{+}.C$	(existential quantifier)
E, F	$\rightarrow$	$A \mid$	(atomic concept)
		$\top \mid$	(top)
		$E \sqcap F \mid$	(conjunction)
		$E \sqcup F \mid$	(disjunction)
		$p \downarrow q \mid$	(agreement)
		$p : E$	(selection)
p, q	$\rightarrow$	$f \mid$	(atomic feature)
		$\star g \mid$	(atomic parametric feature)
		$p \circ q$	(feature chain)
Tc	$\rightarrow$	$(X\ (R)\ Y)$	(temporal constraint)
R, S	$\rightarrow$	$R\ ,\ S \mid$	(disjunction)
		$\mathsf{s} \mid \mathsf{mi} \mid \mathsf{f} \mid \ldots$	(Allen's relations)
X, Y	$\rightarrow$	$\sharp \mid \mathrm{x} \mid \mathrm{y} \mid \ldots$	(temporal variables)

Figure 1: Syntax rules for the temporal language.

Basic types of the language are *concepts*, *individuals*, *temporal variables* and *intervals*. A concept is a description gathering the common properties among a collection of individuals. Concepts can describe entities of the world, states, events. Temporal variables denote intervals bound by temporal constraints, by means of which abstract temporal patterns in the form of constraint networks are expressed. Concepts (resp. individuals) can be specified to hold at a certain interval variable (resp. value) defined by the constraint network. In this way, *action types* (resp. *individual actions*) can be represented in a uniform way by temporally related concepts (resp. individuals).

Concept expressions (denoted by C, D) are built out of *atomic concepts* (denoted by A), *atomic features* (denoted by f), *atomic parametric features* (denoted by $\star g$)[2] and constrained *interval variables* (denoted by X, Y) according to the abstract syntax rules of figure 1[3]. For the basic interval relations we use the same notation as in [Allen, 1991]: before (b), meets (m), during (d), overlaps (o), starts (s), finishes (f), equal ($=$), after (a), met-by (mi), contains (di), overlapped-by (oi), started-by (si), finished-by (fi).

Temporal variables are introduced by the temporal existential quantifier "$\diamond$". Variables appearing in temporal constraints should be declared within the same tempo-

[2]Names for atomic features and atomic parametric features are from the same alphabet of symbols; the $\star$ symbol is not intended as operator, but only as differentiating the two syntactic types.

[3]The syntax rules are expressed following the tradition of concept languages [Hollunder *et al.*, 1990]. It can be read as, e.g. if C is a concept expression and X is a temporal variable, then $C@X$ is a concept expression.

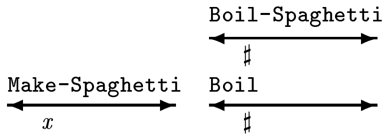


Figure 2: Temporal dependencies in the definition of the `Boil-Spaghetti` action.

ral quantifier, with the exception of the special variable $\sharp$. Temporal variables appearing at the right hand side of a "@" operator are called *bindable*. Concepts should not include *unbound* (a.k.a. *free*) bindable variables, with the exception of the special variable $\sharp$; a bindable variable is said to be bound in a concept if it is declared at the nearest temporal quantifier in the body of which it occurs. Moreover, in chained constructs of the form $((C[Y_1]@X_1)[Y_2]@X_2 \ldots)$ non bindable variables should not appear more than once, with the exception of the special variable $\sharp$.

In this language, unlike [Schmiedel, 1990; Bettini, 1992], it is not possible to express the negated of the existential temporal quantifier: it is argued that the dual of $\Diamond$ – i.e. the universal temporal quantifier $\Box$ – leads the satisfiability problem – and the subsumption – for the language to be undecidable [Halpern and Shoham, 1991; Venema, 1990; Bettini, 1993].

Concept expressions are interpreted in our logic over pairs of *temporal intervals* and *individuals* $\langle i, a\rangle$, meaning that the individual a is in the extension of the concept at the interval i. If a concept is intended to denote an action, then its interpretation can be seen as the set of individual actions of that type occurring at some interval.

Within a concept, the special $\sharp$ variable refers to the generic interval at which the concept itself *holds*[4]; in the case of actions, it refers to the temporal interval at which the action itself *occurs*. A concept holds at an interval X if it is temporally qualified at X – written $C@X$; in this way, every occurrence of $\sharp$ embedded in C is interpreted as the X variable. Since any concept is implicitly temporally qualified at the special $\sharp$ variable, it is not necessary to explicitly qualify concepts at $\sharp$. The temporal existential quantifier introduces interval variables, related each other and possibly to the $\sharp$ variable in a way defined by the set of *temporal constraints*. The informal meaning of a concept with a temporal existential quantification can be understood with the following examples in the action domain [Weida and Litman, 1992].

$$\texttt{Boil-Spaghetti} \doteq \Diamond x\ (x\ \texttt{b}\ \sharp).\ (\texttt{Boil} \sqcap \texttt{Make-Spaghetti}@x)$$

`Boil-Spaghetti` denotes, by definition, any action occurring at some interval such that an event of `Boil`ing occurs at the same time and an event of type `Make-Spaghetti` occurs at some preceding interval. The $\sharp$ interval could be understood as the occurring time of the action type being defined: referring to it within the definition is an explicit way to temporally relate states and actions occurring in the world with respect to the occurrence of the action itself. The temporal constraint $(x\ \texttt{b}\ \sharp)$ states that the interval denoted by x should be before the interval denoted by $\sharp$, i.e. the occurrence interval of the action type `Boil-Spaghetti`. Figure 2 shows the temporal dependencies of the intervals in which the concept `Boil-Spaghetti` holds.

As a more complex example, let `Chef`, `Make-Spaghetti`, `Make-Fettuccine` and `Boil` be atomic concepts, $\star$`AGENT` be an atomic parametric feature and y be a temporal variable. We can describe the class of "*any action of a chef boiling pasta after having made spaghetti or fettuccini*" as:

$$\begin{aligned}&\texttt{Boil-Pasta} \doteq\\ &\quad \Diamond y\ (y\ \texttt{b}\ \sharp).\\ &\qquad \big(\star\texttt{AGENT} : \texttt{Chef} \sqcap \texttt{Boil} \sqcap\\ &\qquad\ (\texttt{Make-Spaghetti} \sqcup \texttt{Make-Fettuccine})@y\big)\end{aligned}$$

The parametric feature $\star$`AGENT` plays the role of *formal* parameter of the action type `Boil-Pasta`, mapping any instance of the action itself to its own agent, independently from time. The occurrence time of the disjunctive action type `Make-Spaghetti` $\sqcup$ `Make- Fettuccine` is bound to the y interval, while the occurrence times of $\star$`AGENT:Chef`, `Boil` and `Boil-Pasta` itself are implicitly bound to the $\sharp$ interval. Please note that, whereas the existence and identity of the $\star$`AGENT` of the action is independent from time, it can be qualified differently in different intervals of time, e.g the fact that it is a `Chef` is necessarily true only at the $\sharp$ interval.

The temporal substitutive qualifier $C[Y]@X$, renames the variable Y, within C, to X and it is a way of making coreference between two temporal variables in different scopes – i.e. declared in different temporal quantifiers. This is useful when using already defined concept names. As an example, `Boil-Pasta` could be redefined by simply renaming the temporal variable x within `Boil-Spaghetti`:

$$\begin{aligned}&\texttt{Boil-Pasta} \doteq\\ &\quad \Diamond y\ (y\ \texttt{b}\ \sharp).\ \big(\star\texttt{AGENT} : \texttt{Chef} \sqcap\\ &\qquad\qquad \big(\texttt{Boil-Spaghetti}[x]@y\ \sqcup\\ &\qquad\qquad\ (\texttt{Boil} \sqcap \texttt{Make-Fettuccine}@y)\big)\big)\end{aligned}$$

The assertion $\texttt{Boil-Pasta}(i, a)$ says that a is an individual action of types `Boil-Pasta` and `Boil` at the interval i, while it is either of type `Make-Spaghetti` or of type `Make-Fettuccine` at some interval j preceding i. Moreover, the same assertion implies that a is related to an $\star$`AGENT` , say b, which is of type `Chef` at the interval i:

$$\begin{aligned}&\texttt{Boil-Pasta}(i, a) \implies\\ &\quad \exists b.\ \texttt{Boil}(i, a) \land \star\texttt{AGENT}(a, b) \land \texttt{Chef}(i, b) \land\\ &\quad\ \exists j.\ \texttt{b}(j, i)\ \land (\texttt{Make-Spaghetti}(j, a) \lor\\ &\qquad\qquad\qquad \texttt{Make-Fettuccine}(j, a))\end{aligned}$$

[4]This variable is usually called *NOW*. We prefer not to adopt such a name, because it could be misleading.

$$
\begin{array}{rcl}
(\mathsf{s})^{\mathcal{E}} & = & \{\langle [u,v],[u_1,v_1]\rangle \in \mathcal{T}^\star_< \times \mathcal{T}^\star_< \mid \\
& & \quad u = u_1 \wedge v < v_1\} \\
(\mathsf{f})^{\mathcal{E}} & = & \{\langle [u,v],[u_1,v_1]\rangle \in \mathcal{T}^\star_< \times \mathcal{T}^\star_< \mid \\
& & \quad v = v_1 \wedge u_1 < u\} \\
(\mathsf{mi})^{\mathcal{E}} & = & \{\langle [u,v],[u_1,v_1]\rangle \in \mathcal{T}^\star_< \times \mathcal{T}^\star_< \mid \\
& & \quad u = v_1\} \\
& \ldots & \textit{(meaning of other Allen's relations)} \\
(R\ ,\ S)^{\mathcal{E}} & = & R^{\mathcal{E}} \cup S^{\mathcal{E}} \\
\langle \overline{X}, \overline{Tc}\rangle^{\mathcal{E}} & = & \{\mathcal{V} : \overline{X} \mapsto \mathcal{T}^\star_< \mid\ \forall (X\ (R)\ Y) \in \overline{Tc}. \\
& & \quad \langle \mathcal{V}(X), \mathcal{V}(Y)\rangle \in (R)^{\mathcal{E}}\}
\end{array}
$$

Figure 3: The temporal interpretation function.

2.1 THE FORMAL SEMANTICS

We assume a linear, unbounded and dense temporal structure $\mathcal{T} = (\mathcal{P}, <)$, where $\mathcal{P}$ is a set of time points and $<$ is a strict total order on $\mathcal{P}$. The *interval set* of a structure $\mathcal{T}$ is defined as the set $\mathcal{T}^\star_<$ of all closed intervals $[u,v] \doteq \{x \in \mathcal{P} \mid u \leq x \leq v, u \neq v\}$ in $\mathcal{T}$.

An *interpretation* $\mathcal{I} \doteq \langle \mathcal{T}^\star_<, \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}}\rangle$ consists of a set $\mathcal{T}^\star_<$ (the *interval set* of the selected temporal structure $\mathcal{T}$), a set $\Delta^{\mathcal{I}}$ (the *domain* of $\mathcal{I}$) and a function $\cdot^{\mathcal{I}}$ (the *primitive interpretation function* of $\mathcal{I}$). The primitive interpretation is a function giving a meaning to atomic concepts, features and parametric features:

$$
A^{\mathcal{I}} \subseteq \mathcal{T}^\star_< \times \Delta^{\mathcal{I}}\ ;
$$
$$
f^{\mathcal{I}} : (\mathcal{T}^\star_< \times \Delta^{\mathcal{I}}) \overset{partial}{\longmapsto} \Delta^{\mathcal{I}}\ ; \qquad \star g^{\mathcal{I}} : \Delta^{\mathcal{I}} \overset{partial}{\longmapsto} \Delta^{\mathcal{I}}
$$

Atomic parametric features are interpreted as partial functions; they differ from atomic features for being independent from time.

To give a meaning to complex temporal and conceptual expression, we introduce the *temporal interpretation function* and the *general interpretation function.*

The *temporal interpretation function* $\cdot^{\mathcal{E}}$ depends on the temporal structure $\mathcal{T}$, and it is defined in figure 3. The labeled directed graph $\langle \overline{X}, \overline{Tc}\rangle$, where $\overline{X}$ is a shorthand for a set of variables – representing the nodes – and $\overline{Tc}$ is a shorthand for a set of temporal constraints – representing the arcs, is called *temporal constraint network*; the temporal interpretation of a temporal constraint network is the set of all assignments of the variables which satisfy the temporal constraints. An *assignment of variables* is a function $\mathcal{V} : \overline{X} \mapsto \mathcal{T}^\star_<$ associating an interval value to each temporal variable. We will write $\langle \overline{X}, \overline{Tc}\rangle^{\mathcal{E}}_{\{x_1 \mapsto t_1, x_2 \mapsto t_2, \ldots\}}$ to denote the subset of $\langle \overline{X}, \overline{Tc}\rangle^{\mathcal{E}}$ where the variable x_i is mapped to the interval value t_i. A temporal constraint network is *consistent* if it admits a non empty interpretation.

A *general interpretation function* $\cdot^{\mathcal{I}}_{\mathcal{V},t,\mathcal{H}}$, based on a interpretation $\mathcal{I}$, an assignment of variables $\mathcal{V}$, an interval t and a set of constraints $\mathcal{H} = \{x_1 \mapsto t_1, \ldots\}$ over the assignments of inner variables, is defined in such a way that the equations of figure 4 are satisfied. The composition should be read from left to right, i.e.

$$
\begin{array}{rcl}
A^{\mathcal{I}}_{\mathcal{V},t,\mathcal{H}} & = & \{a \in \Delta^{\mathcal{I}} \mid \langle t, a\rangle \in A^{\mathcal{I}}\} \\
f^{\mathcal{I}}_{\mathcal{V},t,\mathcal{H}} & = & \hat{f}_t : \Delta^{\mathcal{I}} \overset{partial}{\longmapsto} \Delta^{\mathcal{I}} \mid \\
& & \quad \forall a.\ (a \in dom\, \hat{f}_t \leftrightarrow \\
& & \qquad \langle t, a\rangle \in dom\, f^{\mathcal{I}}) \wedge \\
& & \qquad \hat{f}_t(a) = f^{\mathcal{I}}(t,a) \\
\star g^{\mathcal{I}}_{\mathcal{V},t,\mathcal{H}} & = & \star g^{\mathcal{I}} \\
\top^{\mathcal{I}}_{\mathcal{V},t,\mathcal{H}} & = & \Delta^{\mathcal{I}} \\
(C \sqcap D)^{\mathcal{I}}_{\mathcal{V},t,\mathcal{H}} & = & C^{\mathcal{I}}_{\mathcal{V},t,\mathcal{H}} \cap D^{\mathcal{I}}_{\mathcal{V},t,\mathcal{H}} \\
(C \sqcup D)^{\mathcal{I}}_{\mathcal{V},t,\mathcal{H}} & = & C^{\mathcal{I}}_{\mathcal{V},t,\mathcal{H}} \cup D^{\mathcal{I}}_{\mathcal{V},t,\mathcal{H}} \\
(p \circ q)^{\mathcal{I}}_{\mathcal{V},t,\mathcal{H}} & = & p^{\mathcal{I}}_{\mathcal{V},t,\mathcal{H}} \circ q^{\mathcal{I}}_{\mathcal{V},t,\mathcal{H}} \\
(p : E)^{\mathcal{I}}_{\mathcal{V},t,\mathcal{H}} & = & \{a \in dom\, p^{\mathcal{I}}_{\mathcal{V},t,\mathcal{H}} \mid \\
& & \quad p^{\mathcal{I}}_{\mathcal{V},t,\mathcal{H}}(a) \in E^{\mathcal{I}}_{\mathcal{V},t,\mathcal{H}}\} \\
(p \downarrow q)^{\mathcal{I}}_{\mathcal{V},t,\mathcal{H}} & = & \{a \in dom\, p^{\mathcal{I}}_{\mathcal{V},t,\mathcal{H}} \cap dom\, q^{\mathcal{I}}_{\mathcal{V},t,\mathcal{H}} \mid \\
& & \quad p^{\mathcal{I}}_{\mathcal{V},t,\mathcal{H}}(a) = q^{\mathcal{I}}_{\mathcal{V},t,\mathcal{H}}(a)\} \\
(C@X)^{\mathcal{I}}_{\mathcal{V},t,\mathcal{H}} & = & (C[\sharp]@X)^{\mathcal{I}}_{\mathcal{V},t,\mathcal{H}} \\
(C[Y]@X)^{\mathcal{I}}_{\mathcal{V},t,\mathcal{H}} & = & \begin{cases}
C^{\mathcal{I}}_{\mathcal{V},t,\mathcal{H}} & \text{if } X = \sharp, Y = \sharp \\
C^{\mathcal{I}}_{\mathcal{V},\mathcal{V}(X),\mathcal{H}} & \text{if } X \neq \sharp, Y = \sharp \\
C^{\mathcal{I}}_{\mathcal{V},t,\mathcal{H}\cup\{Y \mapsto \mathcal{V}(\sharp)\}} & \text{if } X = \sharp, Y \neq \sharp \\
C^{\mathcal{I}}_{\mathcal{V},t,\mathcal{H}\cup\{Y \mapsto \mathcal{V}(X)\}} & \text{if } X \neq \sharp, Y \neq \sharp
\end{cases} \\
(\Diamond(\overline{X})\overline{Tc}.\ C)^{\mathcal{I}}_{\mathcal{V},t,\mathcal{H}} & = & \{a \in \Delta^{\mathcal{I}} \mid \\
& & \quad \exists \mathcal{W}.\ \mathcal{W} \in \langle \overline{X}, \overline{Tc}\rangle^{\mathcal{E}}_{\mathcal{H}\cup\{\sharp \mapsto t\}} \wedge \\
& & \quad a \in C^{\mathcal{I}}_{\mathcal{W},t,\emptyset}\}
\end{array}
$$

Figure 4: The general interpretation function.

$(p^{\mathcal{I}}_{\mathcal{V},t,\mathcal{H}} \circ q^{\mathcal{I}}_{\mathcal{V},t,\mathcal{H}})(a)$ means $q^{\mathcal{I}}_{\mathcal{V},t,\mathcal{H}}(p^{\mathcal{I}}_{\mathcal{V},t,\mathcal{H}}(a))$. The expression $dom\, f^{\mathcal{I}}_{\mathcal{V},t,\mathcal{H}}$ (respectively, $dom\, f^{\mathcal{I}}$) denotes the domain of the partial function $f^{\mathcal{I}}_{\mathcal{V},t,\mathcal{H}}$ ($f^{\mathcal{I}}$) – i.e., a subset of $\Delta^{\mathcal{I}}$ ($\mathcal{T}^\star_< \times \Delta^{\mathcal{I}}$) for which $f^{\mathcal{I}}_{\mathcal{V},t,\mathcal{H}}$ ($f^{\mathcal{I}}$) is defined. Intuitively, the general interpretation of a concept $C^{\mathcal{I}}_{\mathcal{V},t,\mathcal{H}}$ is the set of entities of the domain which are of type C at the time interval t, with the assignment for the free temporal variables in C given by $\mathcal{V}$, and with the constraints for the assignment of variables in the scope of the outermost temporal quantifiers given by $\mathcal{H}$.

In absence of free variables in the concept expression – with the exception of $\sharp$, we introduce the *natural* interpretation function $C^{\mathcal{I}}_t$ being equivalent to the general interpretation function $C^{\mathcal{I}}_{\mathcal{V},t,\mathcal{H}}$ with any $\mathcal{V}$ such that $\mathcal{V}(\sharp) = t$, and $\mathcal{H} = \emptyset$. The set of interpretations $\{C^{\mathcal{I}}_{\mathcal{V},t,\mathcal{H}}\}$ obtained by varying $\mathcal{I}, \mathcal{V}, t$ with a fixed $\mathcal{H}$ is maximal wrt set inclusion if $\mathcal{H} = \emptyset$, i.e. the set of natural interpretations includes any set of general interpretations with a fixed $\mathcal{H}$. In fact, since $\mathcal{H}$ represents a constraint in the assignment of variables, the unconstrained set is the larger one.

An interpretation $\mathcal{I}$ is a *model* for a concept C if $C^{\mathcal{I}}_t \neq \emptyset$ for every t. If a concept has a model, then it is *satisfiable*, otherwise it is *unsatisfiable.* A concept C is *subsumed* by a concept D (written $C \sqsubseteq D$) if $C^{\mathcal{I}}_t \subseteq D^{\mathcal{I}}_t$ for every interpretation $\mathcal{I}$ and every interval t.

Concept definitions are introduced by terminological axioms of the form $A \doteq C$. An interpretation $\mathcal{I}$ satisfies $A \doteq C$ iff $A^{\mathcal{I}}_t = C^{\mathcal{I}}_t$, for every t. A *terminology* (TBox) is a finite set of terminological axioms, with the restriction that an atomic concept may appear at most once in the left-hand side of an axiom in the terminology, and that the terminology may not contain cycles [Nebel, 1990]. An interpretation $\mathcal{I}$ is a model for a TBox iff it satisfies all terminological axioms in the TBox. An *expanded* TBox, i.e. a terminology where each definition does not make use of the other definitions, is obtained by applying an interpretation-preserving expansion procedure[5], i.e. substituting every defined concept occurring in a definition with its defining term.

Please note that concepts in this language are always satisfiable, with the proviso that the temporal constraints introduced by the existential quantifiers are consistent. This can be easily checked, after the reduction of the concept into a *normal form* (see section 4), by checking each resulting temporal constraint network using some standard algorithm [van Beek and Cohen, 1990]. In the following we will consider satisfiable concepts only.

It is interesting to notice that only the relations s, f, mi are really necessary in the concept language, because it is possible to express any temporal relationship between two distinct intervals using only these three relations and their transposes si, fi, m [Halpern and Shoham, 1991]. In fact, the following equivalences hold:

$\Diamond x\ (x\ \mathsf{a}\ \sharp).\ C@x \equiv \Diamond x\ (x\ \mathsf{mi}\ \sharp).\ (\Diamond y\ (y\ \mathsf{mi}\ \sharp).\ C@y)@x$
$\Diamond x\ (x\ \mathsf{d}\ \sharp).\ C@x \equiv \Diamond x\ (x\ \mathsf{s}\ \sharp).\ (\Diamond y\ (y\ \mathsf{f}\ \sharp).\ C@y)@x$
$\Diamond x\ (x\ \mathsf{o}\ \sharp).\ C@x \equiv \Diamond x\ (x\ \mathsf{s}\ \sharp).\ (\Diamond y\ (y\ \mathsf{fi}\ \sharp).\ C@y)@x$
$\Diamond x\ (x = \sharp).\ C@x \equiv C$

Other interesting equivalences are the following:

$C \equiv \Diamond()\ ().\ C$
$C \equiv C@\sharp$
$C@X \equiv C[\sharp]@X$
$(C[X]@\underline{Y})[\underline{Z}]@W \equiv (C[Z]@W)[X]@Y$
$\top \equiv \Diamond(\overline{X})\ \overline{Tc}.\ \top$

2.2 THE ASSERTIONAL LANGUAGE

We consider now *assertions*, i.e. predications on temporally qualified individual entities; usually, they are referred to as *ABox statements*. Let $\mathcal{O}$ be the alphabet of symbols denoting *individuals*; an assertion is a statement of one of the following forms $C(i,a)$, $p(i,a,b)$, $\star p(a,b)$, $R(i,j)$, where C is a concept, p is a feature, $\star p$ is a parametric feature, R is a temporal relation, a and b are in $\mathcal{O}$ and denote individuals, i and j denote intervals in $\mathcal{T}^{\star}_{<}$.

In order to assign a meaning to the assertions, the interpretation function $\cdot^{\mathcal{I}}$ is extended to individuals, so that $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ for each individual $a \in \mathcal{O}$ and $a^{\mathcal{I}} \neq b^{\mathcal{I}}$ if $a \neq b$ (Unique Name Assumption). Moreover, we intend $i^{\mathcal{E}}$ to be an element of $\mathcal{T}^{\star}_{<}$. The semantics of the assertions is the following: $C(i,a)$ is satisfied by an interpretation $\mathcal{I}$ iff $a^{\mathcal{I}} \in C^{\mathcal{I}}_{i^{\mathcal{E}}}$; $p(i,a,b)$ is satisfied by $\mathcal{I}$ iff $p^{\mathcal{I}}_{i^{\mathcal{E}}}(a^{\mathcal{I}}) = b^{\mathcal{I}}$; $\star p(a,b)$ is satisfied by $\mathcal{I}$ iff $\star p^{\mathcal{I}}(a^{\mathcal{I}}) = b^{\mathcal{I}}$; and $R(i,j)$ is satisfied by $\mathcal{I}$ iff $\langle i^{\mathcal{E}}, j^{\mathcal{E}}\rangle \in R^{\mathcal{E}}$.

[5]The expansion procedure can be expensive [Nebel, 1990].

$$\texttt{Make-Spaghetti} \xrightarrow[\textit{(before)}]{} \texttt{Boil}$$

Figure 5: The graphical definition of the `Boil-Spaghetti` plan.

A set Σ of ABox statements and TBox axioms is called a *knowledge base*. An interpretation $\mathcal{I}$ is a *model* of Σ iff every assertion and every terminological axiom of Σ is satisfied by $\mathcal{I}$. If Σ has a model, then it is *satisfiable.* Σ *logically implies* an assertion α (written $\Sigma \models \alpha$) if α is satisfied by every model of Σ. Given a knowledge base Σ, an individual a in $\mathcal{O}$ is said to be an *instance* of a concept C *at the interval* i if $\Sigma \models C(i,a)$.

3 ACTIONS AND PLANS

We show in this section how the temporal language can be applied for action and plan representation using some common domains, like the *cooking* domain [Weida and Litman, 1992] and the *block world* domain [Allen, 1991]. While actions describe how the world is affected by their occurrence, plans are described as a collection of action types constrained by temporal relations. In this way, a plan can be graphically represented as a temporal constraint network, where nodes denote action types. At this level of representation, plans can be seen as *complex actions*: since actions composing a plan can be expanded, plans and actions are not structurally different. This distinction is further elaborated in [Artale and Franconi, 1995], where each action composing a plan is considered as a step referring to a different individual action, and an appropriate function relates a plan to its steps.

3.1 THE COOKING DOMAIN

The plan `Boil-Spaghetti` introduced in section 2 can be depicted as in figure 5.

`Boil-Spaghetti` $\doteq$
  $\Diamond x\ (x\ \mathsf{b}\ \sharp).\ (\texttt{Boil} \sqcap \texttt{Make-Spaghetti}@x)$

The definition employs the $\sharp$ interval to denote the occurrence time of the plan itself; in this way, it is possible to describe how different actions or states of the world concurring to the definition of the plan are related to it. This is why the variable $\sharp$ is explicitly present in the definition of `Boil-Spaghetti`, instead of a generic variable: the `Boil` action should take place at the same time of the plan itself.

The definition of a plan can be reused within the definition of other plans; the plan `Boil-Spaghetti` defined above is used in the definition of `Assemble-Spaghetti-Marinara` (see figure 6):

`Assemble-Spaghetti-Marinara` $\doteq$

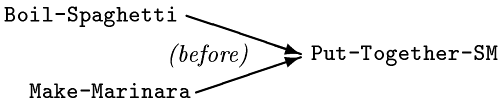


Figure 6: The graphical definition of the `Assemble-Spaghetti-Marinara` plan.

$$\Diamond(y\ z\ w)\ (y\ \mathsf{b}\ w)(z\ \mathsf{b}\ w).\ (\texttt{Boil-Spaghetti}@y\ \sqcap\ \texttt{Make-Marinara}@z\ \sqcap\ \texttt{Put-Together-SM}@w)$$

In this case, precise temporal relations between the two temporal constraint networks are asserted (figure 7): the action `Put-Together-SM` takes place after the `Boil` action. Observe that the occurrence interval of the plan `Assemble-Spaghetti-Marinara` does not appear in the figure, because it is not temporally related with any other interval.

A plan subsuming `Assemble-Spaghetti-Marinara` is the plan defined below `Prepare-Spaghetti`, supposing that the action `Make-Sauce` subsumes `Make-Marinara`. This means that among all the individual actions of the type `Prepare-Spaghetti` there are all the individual actions of type `Assemble-Spaghetti-Marinara`:

$$\texttt{Prepare-Spaghetti}\ \doteq\ \Diamond\ (y\ z)\ ().\ (\texttt{Boil-Spaghetti}@y\ \sqcap\ \texttt{Make-Sauce}@z)$$

Please note that `Boil-Spaghetti` does subsume neither `Prepare-Spaghetti` nor `Assemble-Spaghetti-Marinara`, even if the former is part of the definition of these latter. This could be better explained if we observe how the `Prepare-Spaghetti` plan is expanded:

$$\texttt{Prepare-Spaghetti}\ \doteq\ \Diamond\ (x\ y\ z)(x\ \mathsf{b}\ y).\ (\texttt{Boil}@y \sqcap \texttt{Make-Spaghetti}@x \sqcap \texttt{Make-Sauce}@z)$$

The effect of binding `Boil-Spaghetti` to the temporal variable y has been that the `Boil` action occurs at the interval y, which is possibly different from the occurring time of `Prepare-Spaghetti`; while `Boil-Spaghetti` and `Boil` actions take place *necessarily* at the same time. Subsumption between `Prepare-Spaghetti` and `Boil-Spaghetti` fails since different temporal relations between the actions composing the two plans and the plans themselves are specified. In particular, we can observe that the plan `Boil-Spaghetti` denotes a narrower class than the plan

$$\Diamond(x\ y)\ (x\ \mathsf{b}\ y).\ (\texttt{Boil}@y\ \sqcap \texttt{Make-Spaghetti}@x),$$

which subsumes each of `Prepare-Spaghetti`, `Assemble-Spaghetti-Marinara` and `Boil-Spaghetti`.

3.2 THE BLOCK WORLD DOMAIN

As a further example of the expressive power of the temporal language, we show how to represent the `Stack` action in the block world, as it is defined in [Allen, 1991]. A stacking action involves two blocks, which should be both clear at the beginning; the central part of the action consists of grasping one block; at the end, the blocks are one on top of the other, and the bottom one is no more clear (figure 8).

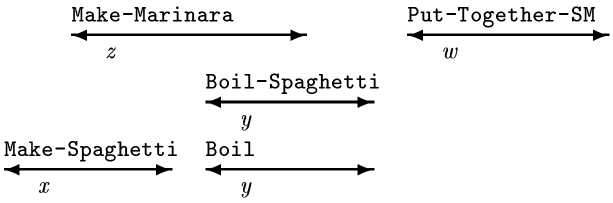


Figure 7: Temporal dependencies in the definition of `Assemble-Spaghetti-Marinara`.

Our representation borrows from the RAT system [Heinsohn *et al.*, 1992] the intuition of representing action parameters by means of partial functions mapping from the action itself to the involved action parameter. In the language, these functions are called *parametric features*. For example, the action `Stack` has the parameters $\star$`OBJECT1` and $\star$`OBJECT2`, representing in some sense the objects which are involved in the action independently from time. So, in the assertion "$\star\texttt{OBJECT1}(a, \textit{block-a})$", *block-a* denotes the first object involved in the action a at any interval. On the other hand, an assertion involving a (non-parametric) feature, e.g. "$\texttt{ON}(i, \textit{block-a}, \textit{block-b})$", does not imply anything about the truth value at intervals other than i.

The concept definition makes use of temporal qualified concept expressions, including feature *selections* and *agreements*. The expression $(\star\texttt{OBJECT2} : \texttt{Clear-Block})@x$ means that the second parameter of the action should be a `Clear-Block` at the interval denoted by x; $(\star\texttt{OBJECT1}\circ\texttt{ON} \downarrow \star\texttt{OBJECT2})@y$ indicates that at the interval y the object `on` which $\star$`OBJECT1` is placed is $\star$`OBJECT2`. The formal definition of the action `Stack` is:

$$\begin{aligned}\texttt{Stack}\ \doteq\ & \Diamond(x\ y\ z\ v\ w)\\ & \quad (x\ \mathsf{fi}\ \sharp)(y\ \mathsf{mi}\ \sharp)(z\ \mathsf{mi}\ \sharp)(v\ \mathsf{o}\ \sharp)\\ & \quad (w\ \mathsf{f}\ \sharp)(w\ \mathsf{mi}\ v).\\ & \quad \big((\star\texttt{OBJECT2} : \texttt{Clear-Block})@x \sqcap\\ & \quad (\star\texttt{OBJECT1}\circ\texttt{ON} \downarrow \star\texttt{OBJECT2})@y \sqcap\\ & \quad (\star\texttt{OBJECT1} : \texttt{Clear-Block})@v \sqcap\\ & \quad (\star\texttt{OBJECT1} : \texttt{Hold-Block})@w \sqcap\\ & \quad (\star\texttt{OBJECT1} : \texttt{Clear-Block})@z\big)\end{aligned}$$

The above concept does not state which properties are the prerequisites for the stacking action or which properties must be true whenever the action succeeds. What this action intuitively states is that $\star$`OBJECT1` will be on $\star$`OBJECT2` in a situation where both objects are clear at the start of the action. Note that the world state described at the intervals denoted by v, w, z is the result of an action of *grasping* a previously clear block:

$$\begin{aligned}\texttt{Grasp}\ \doteq\ & \Diamond(x\ w\ z)\ \ (x\ \mathsf{o}\ \sharp)(w\ \mathsf{f}\ \sharp)(w\ \mathsf{mi}\ x)(z\ \mathsf{mi}\ \sharp).\\ & \quad \big((\star\texttt{OBJECT1} : \texttt{Clear-Block})@x \sqcap\\ & \quad (\star\texttt{OBJECT1} : \texttt{Hold-Block})@w \sqcap\\ & \quad (\star\texttt{OBJECT1} : \texttt{Clear-Block})@z\big)\end{aligned}$$

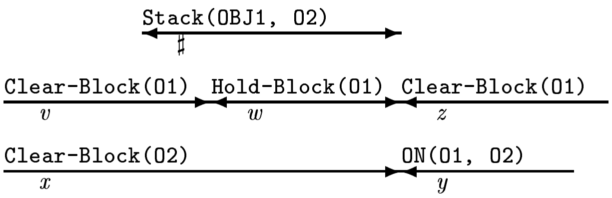


Figure 8: The definition of the `Stack` action.

We can redefine the `Stack` action by making use of the `Grasp` action:

$$\begin{aligned}\texttt{Stack} \;\doteq\;& \diamond(x\ y\ u\ v)\ \ (x \ \mathsf{fi}\ \sharp)(y\ \mathsf{mi}\ \sharp)(u\ \mathsf{f}\ \sharp)(v\ \mathsf{o}\ \sharp).\\ & \big((\star\texttt{OBJECT2} : \texttt{Clear-Block})@x \sqcap\\ & (\star\texttt{OBJECT1}\circ\texttt{ON}\downarrow\star\texttt{OBJECT2})@y \sqcap\\ & (\texttt{Grasp}[x]@v)@u\big)\end{aligned}$$

The temporal substitutive qualifier $(\texttt{Grasp}[x]@u)$ *renames* within the defined `Grasp` action the variable x to v and it is a way of making coreference between two temporal variables, while the temporal constraints peculiar to the renamed variable x are inherited by the substituting interval v. The effect of temporally qualifying the grasping action at u is that the $\sharp$ variable associated to the grasping action itself is bound to the interval denoted by u – remember that the variable $\sharp$ used inside an action refers to the occurrence time of the action itself. Because of this binding on the occurrence time of the grasping action, the $\sharp$ variable in the grasping action and the $\sharp$ variable in the stacking action denote different time intervals, so that the grasping action occurs at an interval finishing the occurrence time of the stacking action.

Now we show how from a series of observations in the world we can make action recognition, an inference service which computes if an *individual action* is an instance of an *action type* at a certain interval. Given the following ABox, describing a world where blocks can be clear, grasped and/or on each other and where a generic individual action a is taking place at time interval i_a having the blocks *block-a* and *block-b* as its parameters:

$\star\texttt{OBJECT1}(a, \textit{block-a}),\ \star\texttt{OBJECT2}(a, \textit{block-b}),$
$\mathsf{o}(i_1, i_a),\ \texttt{Clear-Block}(i_1, \textit{block-a}),$
$\mathsf{fi}(i_2, i_a),\ \texttt{Clear-Block}(i_2, \textit{block-b}),$
$\mathsf{mi}(i_3, i_1),\ \mathsf{f}(i_3, i_a),\ \texttt{Hold-Block}(i_3, \textit{block-a}),$
$\mathsf{mi}(i_4, i_a),\ \texttt{Clear-Block}(i_4, \textit{block-a}),$
$\mathsf{mi}(i_5, i_a),\ \texttt{ON}(i_5, \textit{block-a}, \textit{block-b})$

then the system recognizes that in the context of a knowledge base Σ, composed by the above ABox and the definition of the `Stack` concept in the TBox, the individual action a is an instance of the concept `Stack` at the time interval i_a, i.e. $\Sigma \models \texttt{Stack}(i_a, a)$.

4 THE CALCULUS

This section presents a calculus for the temporal concept language. We first look for a *normal form* of concepts, which will be useful for the subsumption and instance recognition algorithms. Since the dimension of the normal form of a concept can be exponential in presence of concept disjunction, a way to compute an effective normal form in the special case of absence of concept disjunction is also devised. Section 4.1 considers the language without concept disjunction; section 4.2 considers the full language.

- $C@X \sqcap D@X \quad\rightarrow\quad (C \sqcap D)@X$
- $(C@X_1)@X_2 \quad\rightarrow\quad \begin{cases} C@X_1 & \text{if } X_1 \neq \sharp \\ C@X_2 & \text{if } X_1 = \sharp \end{cases}$
- $p : (q : C) \quad\rightarrow\quad (p \circ q) : C$
- $\diamond(\overline{X}_1)\ \overline{Tc}_1.\ \big(C \sqcap (\diamond(\overline{X}_2)\ \overline{Tc}_2.\ D)@X\big) \quad\rightarrow$
  $\diamond(\overline{X}_1 \uplus \overline{X}_2)\ \overline{Tc}_1 \uplus \overline{Tc}_2[X/\sharp].\ (C \sqcap D_+[X/\sharp])$
- $C \sqcap \diamond(\overline{X})\ \overline{Tc}.\ D \quad\rightarrow\quad \diamond(\overline{X})\ \overline{Tc}.\ (C \sqcap D)$
  if C doesn't contain variables

Comment: $\overline{X} \uplus \overline{Y}$ returns the union of the sets $\overline{X}$ and $\overline{Y}$, where all the elements of $\overline{Y}$ occurring in $\overline{X}$ are renamed, except for $\sharp$; Z_+ is intended to be the expression Z where the same renaming has taken place.

Figure 9: Nondeterministic rewrite rules to transform an arbitrary concept into an equivalent existential concept.

4.1 NORMAL FORM AND SUBSUMPTION

Let us consider in this section the restricted language without concept disjunction. Every concept of the restricted language in an expanded TBox can be reduced into an equivalent *existential concept* of the form: $\diamond(\overline{X})\ \overline{Tc}.\ (Q^1@X^1 \sqcap \ldots \sqcap Q^n@X^n)$, where each Q^j is a non-temporal concept, i.e. it does not contain neither temporal quantifiers nor temporal qualifiers – nor concept disjunctions. Figure 9 presents a set of rules for reducing a concept C into the existential form $(\mathit{ef}\,C)$, once concept names and substitutive qualifiers have been expanded in C. A concept in existential form can be seen as a *conceptual temporal constraint network*, i.e. a labeled directed graph $\langle \overline{X}, \overline{Tc}, \overline{Q@X} \rangle$ where arcs are labeled with a set of arbitrary temporal relationships – representing their disjunction – and nodes are labeled with nontemporal concepts.

Proposition 1 (Existential Form) *Every concept can be reduced into an equivalent existential concept by applying the rules of figure 9, i.e. given a concept C, $C^{\mathcal{I}}_{\mathcal{V},t} = (\mathit{ef}\,C)^{\mathcal{I}}_{\mathcal{V},t}$ for every interpretation $\mathcal{I}$, every assignment $\mathcal{V}$ and every interval t.*

Given a concept in existential form, the temporal completion of the constraint network is computed:

Definition 1 (Temporal Completion) The temporal completion of a concept in existential form – the Completed Existential Form, CEF – is obtained by sequentially applying the following steps:

• *(Closure)* The transitive closure of the Allen temporal relations in the conceptual temporal constraint network is computed (see e.g. [van Beek and Cohen, 1990]).

• *(parameter introduction)* New information is added to each node because of the presence of parameters, as the following rules show (the $\rightsquigarrow$ symbol is intended in such a way that each time the concept expression in the left hand side appears at a top level conjunction in some node of the temporal constraint network then the right hand side represents the concept expression that must be conjunctively added to all the other nodes):

$$\star g_1 \circ \ldots \circ \star g_n [\circ\, f\, [\circ\, p]] : C$$
$$\quad \rightsquigarrow\ \star g_1 \circ \ldots \circ \star g_n : \top.$$
$$\star g_1^1 \circ \ldots \circ \star g_n^1 \downarrow \star g_1^2 \circ \ldots \circ \star g_m^2$$
$$\quad \rightsquigarrow\ \star g_1^1 \circ \ldots \circ \star g_n^1 \downarrow \star g_1^2 \circ \ldots \circ \star g_m^2.$$
$$\star g_1^1 \circ \ldots \circ \star g_n^1 \circ\ f^1 \left[\circ\, p^1\right] \downarrow \star g_1^2 \circ \ldots \circ \star g_m^2 \circ\ f^2 \left[\circ\, p^2\right]$$
$$\quad \rightsquigarrow\ \star g_1^1 \circ \ldots \circ \star g_n^1 : \top\ \sqcap\ \star g_1^2 \circ \ldots \circ \star g_m^2 : \top.$$

• *(= introduction)* New temporal constraints with the "=" relation are introduced, if they are not already present, for every variable declared in the constraint network and for the $\sharp$ variable, by applying the following rewrite rule:

$$\Diamond(\overline{X})\ \overline{Tc}.\ C\ \rightarrow$$
$$\quad \Diamond(\overline{X})\ \overline{Tc}\ (\sharp\ =\ \sharp)\ (x\ =\ x)\ (y\ =\ y) \cdots .\ C$$

• *(= collapsing)* For each equality temporal constraint, collapse the equal nodes by applying the following rewrite rule:

$$\Diamond(\overline{X})\ \overline{Tc}\ (x\ =\ y).\ C\ \rightarrow\ \Diamond(\overline{X} \setminus \{y\})\ \overline{Tc}_{[x/y]}.\ C_{[x/y]}$$
$$\text{if } x \neq y \text{ and } y \neq \sharp \qquad \Box$$

Proposition 2 (Equivalence of CEF) *Every concept in existential form can be reduced into an equivalent completed existential concept by applying the above procedure.*

The most relevant properties of a concept in CEF is that all the admissible interval temporal relations are explicit and the concept expression in each node is no more refinable without changing the overall concept meaning; this is stated by the following proposition.

Proposition 3 (Node Independence of CEF) *Let* $\langle \overline{X}, \overline{Tc}, \overline{Q@X} \rangle$ *be a conceptual temporal constraint network in its completed form (CEF); for all* $Q \in \overline{Q}$ *and for all* $C \not\sqsupseteq Q$ *then* $\langle \overline{X}, \overline{Tc}, \overline{(Q \sqcap C)@X} \rangle^{\mathcal{I}}_{\mathcal{V},t} \neq \langle \overline{X}, \overline{Tc}, \overline{Q@X} \rangle^{\mathcal{I}}_{\mathcal{V},t}$ *for every interpretation* $\mathcal{I}$*, every assignment* $\mathcal{V}$ *and every interval* t*.*

Proof. The proposition states that the information in each node of the CEF is independent from the information in the other nodes. In fact, $\langle \overline{X}, \overline{Tc}, \overline{(Q \sqcap C)@X} \rangle^{\mathcal{I}}_{\mathcal{V},t} = \langle \overline{X}, \overline{Tc}, \overline{Q@X} \rangle^{\mathcal{I}}_{\mathcal{V},t}$ if the concept expression in one node implies new information in some other node, since, for nontemporal concepts, adding information means restricting the concept in some way. We examine the only two cases in which the information stated in a node adds new information in some other node, and we show that these cases are covered by the completion rules.

i. *Nodes related only by means of the equal relations.* The *(= collapsing)* rule provides to collapse two contemporary nodes conjoining the concept expressions of each of them. Note that, thanks to the *(Closure)* rule, all the possible equal temporal relations are made explicit.

ii. *Time-invariant information.* Every time-invariant information should spread over all the nodes. Only parametric features and the $\top$ concept have a time-invariant semantics: by induction, we prove that the only time-invariant concepts are $\top$, $\star g_1 \circ \ldots \circ \star g_n : \top$, $\star g_1 \circ \ldots \circ \star g_n \downarrow \star p_1 \circ \ldots \circ \star p_m$ (with $n, m \geq 1$) or an arbitrary conjunction of these terms. The *(parameter introduction)* rule considers all the possible syntactical cases of deduction concerning time-invariant concept expressions. $\Box$

As an example, we show the completed existential form of the previously introduced `Stack` action:

$$
\begin{aligned}
&\texttt{Stack}\ \doteq\\
&\ \Diamond(x\ y\ v\ w\ z)\\
&\quad (x\ \mathsf{fi}\ \sharp)(y\ \mathsf{mi}\ \sharp)(v\ \mathsf{mi}\ \sharp)(w\ \mathsf{f}\ \sharp)(z\ \mathsf{o}\ \sharp)(y\ \mathsf{mi}\ x)\\
&\quad (v\ \mathsf{mi}\ x)(w\ \mathsf{f}\ x)(z\ (\mathsf{o},\mathsf{d},\mathsf{s})\ x)(v\ (=,\mathsf{s},\mathsf{si})\ y)\\
&\quad (w\ \mathsf{m}\ y)(z\ \mathsf{b}\ y)(w\ \mathsf{m}\ v)(z\ \mathsf{b}\ v)(w\ \mathsf{mi}\ z)\\
&\quad (\sharp = \sharp)(x = x)(y = y)(v = v)(w = w)(z = z).\\
&\ \ \big((\star\texttt{OBJECT2} : \texttt{Clear-Block} \sqcap \star\texttt{OBJECT1} : \top)@x \sqcap\\
&\quad (\star\texttt{OBJECT1} : \texttt{Clear-Block} \sqcap \star\texttt{OBJECT2} : \top)@y \sqcap\\
&\quad (\star\texttt{OBJECT1} \circ \texttt{ON} \downarrow \star\texttt{OBJECT2})@v \sqcap\\
&\quad (\star\texttt{OBJECT1} : \texttt{Hold-Block} \sqcap \star\texttt{OBJECT2} : \top)@w \sqcap\\
&\quad (\star\texttt{OBJECT1} : \texttt{Clear-Block} \sqcap \star\texttt{OBJECT2} : \top)@z\big)
\end{aligned}
$$

As we have seen in section 2.1, a concept subsumes another just in case every possible instance of the second is also an instance of the first, for every time interval. Concept subsumption in the temporal language is reduced to concept subsumption between nontemporal concepts and to subsumption between temporal constraint networks. A similar general procedure was first presented in [Weida and Litman, 1992], where the language for nontemporal concepts is less expressive – it does not include features nor parametric features. Algorithms to compute subsumption between nontemporal concepts – $E_1 \sqsubseteq E_2$ – are well known, see e.g. [Hollunder *et al.*, 1990].

Definition 2 (Variable mapping) A *variable mapping* $\mathcal{M}$ is a total function $\mathcal{M} : \overline{X} \mapsto \overline{X}$ such that $\mathcal{M}(\sharp) = \sharp$. We write $\mathcal{M}(\overline{X})$ to intend $\{\mathcal{M}(X) \mid X \in \overline{X}\}$, and $\mathcal{M}(\overline{Tc})$ to intend $\{(\mathcal{M}(X)\ (R)\ \mathcal{M}(Y)) \mid (X\ (R)\ Y) \in \overline{Tc}\}$. $\Box$

Definition 3 (Temporal Constraint subsumption) A temporal constraint $(X_1 (R_1) Y_1)$ is said to *subsume* a temporal constraint $(X_2 (R_2) Y_2)$ under a generic variable mapping $\mathcal{M}$ – written $(X_1 (R_1) Y_1) \sqsupseteq_{\mathcal{M}} (X_2\ (R_2)\ Y_2)$ – if $\mathcal{M}(X_1) = X_2$, $\mathcal{M}(Y_1) = Y_2$ and $(R_1)^{\mathcal{E}} \supseteq (R_2)^{\mathcal{E}}$ for every temporal interpretation $\mathcal{E}$. $\Box$

Proposition 4 $(X_1(R_1)Y_1) \sqsupseteq_{\mathcal{M}} (X_2(R_2)Y_2)$ *if and only if* $\mathcal{M}(X_1) = X_2$, $\mathcal{M}(Y_1) = Y_2$ *and the disjuncts in* R_1 *are a superset of the disjuncts in* R_2.

Proof. Follows from the observation that the 13 temporal relations are mutually disjoint and their union covers the whole interval pairs space. □

Definition 4 (Temporal Constraint Network subsumption) A temporal constraint network $\langle \overline{X}_1, \overline{Tc}_1 \rangle$ subsumes a temporal constraint network $\langle \overline{X}_2, \overline{Tc}_2 \rangle$ under a variable mapping $\mathcal{M} : \overline{X}_1 \mapsto \overline{X}_2$, written $\langle \overline{X}_1, \overline{Tc}_1 \rangle \sqsupseteq_{\mathcal{M}} \langle \overline{X}_2, \overline{Tc}_2 \rangle$, if $\langle \mathcal{M}(\overline{X}_1), \mathcal{M}(\overline{Tc}_1) \rangle^{\mathcal{E}} \supseteq \langle \overline{X}_2, \overline{Tc}_2 \rangle^{\mathcal{E}}$ for every temporal interpretation $\mathcal{E}$. □

Proposition 5 $\langle \overline{X}_1, \overline{Tc}_1 \rangle \sqsupseteq_{\mathcal{M}} \langle \overline{X}_2, \overline{Tc}_2 \rangle$ *iff there exist a variable mapping* $\mathcal{M} : \overline{X}_1 \mapsto \overline{X}_2$ *such that for all* $X_1^i, Y_1^j \in \overline{X}_1$ *exist* $X_2^m, Y_2^n \in \overline{X}_2$ *which satisfy* $(X_1^i\ (R_1^{i,j})\ Y_1^j) \sqsupseteq_{\mathcal{M}} (X_2^m\ (R_2^{m,n})\ Y_2^n)$.

Proof. "$\Leftarrow$" From the definition of interpretation of a temporal constraint network, it is easy to see that each assignment of variables $\mathcal{V}$ in the interpretation of $\langle \overline{X}_1, \overline{Tc}_1 \rangle$ is also an assignment in the interpretation of $\langle \mathcal{M}(\overline{X}_2), \mathcal{M}(\overline{Tc}_2) \rangle$, since for all i, j $(R_1^{i,j})^{\mathcal{E}} \supseteq (R_2^{m,n})^{\mathcal{E}}$.
"$\Rightarrow$" Suppose that we are not able to find such a mapping; then, by hypotheses, for each possible variable mapping there exist some i, j such that $(R_1^{i,j})^{\mathcal{E}} \not\supseteq (R_2^{m,n})^{\mathcal{E}}$. So, for each variable mapping we can build an interpretation $\mathcal{E}^*$ and an assignment $\mathcal{V}^*$ such that $\mathcal{V}^* \in \langle \overline{X}_2, \overline{Tc}_2 \rangle^{\mathcal{E}^*}$ and $\mathcal{V}^* \notin \langle \mathcal{M}(\overline{X}_1), \mathcal{M}(\overline{Tc}_1) \rangle^{\mathcal{E}^*}$. But this contradicts the assumption that $\langle \overline{X}_1, \overline{Tc}_1 \rangle \sqsupseteq_{\mathcal{M}} \langle \overline{X}_2, \overline{Tc}_2 \rangle$. □

Definition 5 (S-mapping) A *s-mapping* from a conceptual temporal constraint network $\langle \overline{X}_1, \overline{Tc}_1, \overline{Q@X}_1 \rangle$ to a conceptual temporal constraint network $\langle \overline{X}_2, \overline{Tc}_2, \overline{Q@X}_2 \rangle$ is a variable mapping $\mathcal{S} : \overline{X}_1 \mapsto \overline{X}_2$ such that the nontemporal concept labeling each node in $\overline{X}_1$ subsumes the nontemporal concept labeling the corresponding node in $\mathcal{S}(\overline{X}_1)$, and $\langle \overline{X}_1, \overline{Tc}_1 \rangle \sqsupseteq_{\mathcal{S}} \langle \overline{X}_2, \overline{Tc}_2 \rangle$. □

The last normalization procedure reduces the graph by eliminating nodes with redundant information. This final normalization step ends up with the concept in the *essential graph form*, that will be the normal form used for checking concept subsumption.

Definition 6 (Essential graph) The subgraph of the CEF of a conceptual temporal constraint network $T = \langle \overline{X}, \overline{Tc}, \overline{Q@X} \rangle$ obtained by deleting the nodes labeled only with time-invariant concept expressions – with the exception of the $\sharp$ node – is called *essential graph* of T: $(\text{ess}\, T)$. □

Proposition 6 (Essential Graph Reduction) *Every conceptual temporal constraint network in completed existential form can be reduced into an equivalent essential graph, i.e. given a conceptual temporal constraint network* T, $T^{\mathcal{I}}_{\mathcal{V},t} = (\text{ess}\, T)^{\mathcal{I}}_{\mathcal{V},t}$ *for every interpretation* $\mathcal{I}$, *every assignment* $\mathcal{V}$ *and every interval* t.

Definition 7 (Redundant Node) A node K in a conceptual temporal constraint network $\langle \overline{X}, \overline{Tc}, \overline{Q@X} \rangle$ is *redundant* if the network resulting by deleting it is equivalent to the original one: $\langle \overline{X} \setminus K, \overline{Tc}|_{\overline{X} \setminus K}, \overline{Q@X}|_{\overline{X} \setminus K} \rangle^{\mathcal{I}}_{\mathcal{V},t} = \langle \overline{X}, \overline{Tc}, \overline{Q@X} \rangle^{\mathcal{I}}_{\mathcal{V},t}$ for every interpretation $\mathcal{I}$, every assignment $\mathcal{V}$ and every interval t. □

Definition 8 (Mapping-Redundant Set) A set $\overline{K}$ of nodes in a conceptual temporal constraint network $\langle \overline{X}, \overline{Tc}, \overline{Q@X} \rangle$ is a mapping-redundant set if there exists a s-mapping $\mathcal{S}$ from $\langle \overline{K}, \overline{Tc}|_{\overline{K}}, \overline{Q@X}|_{\overline{K}} \rangle^{\mathcal{I}}_{\mathcal{V},t}$ to $\langle \overline{X} \setminus \overline{K}, \overline{Tc}|_{\overline{X} \setminus \overline{K}}, \overline{Q@X}|_{\overline{X} \setminus \overline{K}} \rangle^{\mathcal{I}}_{\mathcal{V},t}$, such that $\forall K_i \in \overline{K}.\ \mathcal{S}(K_i) = X_j \to \exists Tc \in \overline{Tc}.\ Tc \sqsupseteq_{\mathsf{id}} (K_i = X_j)$.

Proposition 7 *A node in an essential graph is redundant if and only if it is in the maximal – wrt set inclusion – mapping-redundant set of the graph. Moreover, the only other way to add redundant nodes to an essential graph is to add time-invariant nodes.*

The following theorem states that subsumption is decidable and provides a sound and complete procedure to compute it: first reduce the subsumer and the subsumee in essential graph form, then look for a s-mapping between the essential graphs by exhaustive search.

Theorem 1 (Concept subsumption) *A conceptual temporal constraint network* $T_1 = \langle \overline{X}_1, \overline{Tc}_1, \overline{Q@X}_1 \rangle$ *subsumes a conceptual temporal constraint network* $T_2 = \langle \overline{X}_2, \overline{Tc}_2, \overline{Q@X}_2 \rangle$ – $T_1 \sqsupseteq T_2$ – *iff there exists a s-mapping from the essential graph of* T_1 *to the essential graph of* T_2.

Proof. "$\Leftarrow$" Follows from the soundness of TCN subsumption, from the soundness of the algorithm for computing the subsumption between nontemporal concepts and from the semantics of the conceptual temporal constraint networks.

"$\Rightarrow$" Suppose that such a s-mapping does not exist. We can distinguish two main cases.

i) There is not a mapping $\mathcal{M}$ such that $\langle \overline{X}_1, \overline{Tc}_1 \rangle \sqsupseteq_{\mathcal{M}} \langle \overline{X}_2, \overline{Tc}_2 \rangle$. By adding redundant nodes to T_2, we obtain an equivalent conceptual temporal constraint network $T_2^* = \langle \overline{X}_2^*, \overline{Tc}_2^*, \overline{Q@X}_2^* \rangle$. Let us consider the extended network in a way that there exists a variable mapping $\mathcal{M}^*$ such that $\langle \overline{X}_1, \overline{Tc}_1 \rangle \sqsupseteq_{\mathcal{M}^*} \langle \overline{X}_2^*, \overline{Tc}_2^* \rangle$. Now, for all possible $\mathcal{M}^*$, there is a node $X_1^i \in \overline{X}_1$ such that $\mathcal{M}^*(X_1^i) = X_2^j$ with $X_2^j \notin \overline{X}_2$ and $Q_1^i \not\sqsupseteq Q_2^j$, since either Q_1^i is not in a time-invariant node – whereas Q_2^j

- $(C \sqcup D)@X \quad \rightarrow \quad C@X \sqcup D@X$
- $p : (C \sqcup D) \quad \rightarrow \quad p : C \sqcup p : D$
- $(C_1 \sqcup C_2) \sqcap D \quad \rightarrow \quad (C_1 \sqcap D) \sqcup (C_2 \sqcap D)$
- $\diamond(\overline{X})\ \overline{Tc}.\ (C \sqcup D) \quad \rightarrow \quad \diamond(\overline{X})\ \overline{Tc}.\ C \sqcup \diamond(\overline{X})\ \overline{Tc}.\ D$

Figure 10: Nondeterministic rewrite rules for computing the disjunctive form.

is in a time-invariant node – or Q_2^j is in the maximal mapping-redundant set of T_2^* – contradicting the hypothesis that the mapping $\mathcal{M}$ does not exist. Since the construction of $\mathcal{M}^*$ allows for the existence of a unique $\mathcal{V}$[6] for both networks, then we can build an instance of T_2^* which is not an instance of T_1.

ii) For each possible mapping $\mathcal{M}$ such that $\langle \overline{X}_1, \overline{Tc}_1 \rangle \sqsupseteq_{\mathcal{M}} \langle \overline{X}_2, \overline{Tc}_2 \rangle$ there will be always two nodes X_1^i and X_2^j such that $\mathcal{M}(X_1^i) = X_2^j$ and $Q_1^i \not\sqsupseteq Q_2^j$. Since, from proposition 3, the concept expression in each node is independent from any concept expression of the other nodes, we cannot refine the concept expression Q_2^j, looking for a subsumption relationships with Q_1^i. Besides, Q_1^i can not be eliminated – i.e. generalized to $\top$ – from T_1, since the conditions of proposition 7 hold. So, we can build an instance of T_2 which is not an instance of T_1.

Both cases contradict the assumption that T_1 subsumes T_2. □

4.2 DISJUNCTIVE CONCEPTS

In this section we introduce the disjunction operator and we show how to modify the calculus in order to check subsumption. In the way of computing subsumption, we need a normal form for concepts. The normalization procedure is essentially the same as the one reported in section 4.1. The figure 10 shows the rules dealing explicitly with disjunction. Those, in addition to the rules introduced in figure 9, reduce every concept into an equivalent *disjunctive concept* of the form:

$$\big(\diamond(\overline{X}_1)\ \overline{Tc}_1.\ G_1\big) \sqcup \cdots \sqcup \big(\diamond(\overline{X}_n)\ \overline{Tc}_n.\ G_n\big) \sqcup H_1 \sqcup \cdots \sqcup H_m$$

where G_i are conjunctions of concepts of the form $H^j@X^k$, and each H does not contain neither temporal quantifiers, nor temporal qualifiers, nor disjunctions.

Given a concept in *disjunctive form*, applying the *temporal completion* rules showed in section 4.1 to each disjunct, we end up with an equivalent concept in *completed disjunctive form* where the *node independence* property is preserved. Then the *essential graph form* is computed, obtaining a concept where each disjunct does not contain time-invariant nodes. At this point, we are able to compute the *disjunctive normal form* (dnf C).

Definition 9 (Disjunctive Normal Form) The *disjunctive normal form* of a concept is computed starting from its *essential graph form* and applying the following nondeterministic rewrite rules to each disjunct:

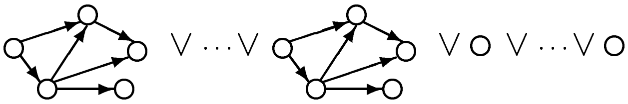
- *($\sqcup$ introduction)* Transform the conceptual temporal constraint network into an equivalent disjunction of conceptual temporal constraint networks containing only basic temporal relationships:

$$\diamond(\overline{X})\,(X_1\ (R,S)\ X_2)\,\overline{Tc}.C \rightarrow \diamond(\overline{X})(X_1\ R\ X_2)\overline{Tc}.C\ \sqcup \diamond(\overline{X})(X_1\ S\ X_2)\overline{Tc}.C$$

- *($\sqcup$ elimination)* If the disjunct is unsatisfiable – i.e. the temporal constraint network associated with it is inconsistent – then eliminate it. □

Proposition 8 (Equivalence of DNF) *Every disjunctive essential graph can be reduced into an equivalent disjunctive normal concept by applying the above procedure.*

A concept in *disjunctive normal form* can be seen as the disjunction of several *basic conceptual temporal constraint networks*, where arcs are labeled with basic temporal relationships and nodes are labeled with non-temporal non-disjunctive concepts:



Each basic conceptual temporal constraint network – i.e. a disjunct of the normal form – has some interesting properties, which are crucial for the proofs of the theorem: temporal constraints are always explicit, i.e. any two intervals are related by a basic temporal relation; there is no disjunction, neither implicit nor explicit, neither in the conceptual part nor in the temporal part; the information in each node is independent from the information in the other nodes. The following theorem reduces subsumption between concepts in disjunctive normal form into subsumption of disjunction-free concepts, such that the results of theorem 1 can be applied.

Theorem 2 (Concept subsumption) *Let $C = C_1 \sqcup \cdots \sqcup C_m$ and $D = D_1 \sqcup \cdots \sqcup D_n$ be concepts in disjunctive normal form; then $C \sqsubseteq D$ if and only if $\forall i \exists j.\ C_i \sqsubseteq D_j$.*

Proof. [7] Since it is easy to show that

$$C_1 \sqcup \ldots \sqcup C_n \sqsubseteq D \quad \text{iff} \quad \forall i. C_i \sqsubseteq D$$

we need only to prove the restricted thesis

$$C_i \sqsubseteq D_1 \sqcup \cdots \sqcup D_n \quad \text{iff} \quad C_i \sqsubseteq D_1 \vee \ldots \vee C_i \sqsubseteq D_n$$

[6] Since subsumption is computed with respect to a fixed evaluation time, $\mathcal{V}$ maps the different occurrences of $\sharp$ to the same interval; this justifies the choice that $\mathcal{M}(\sharp) = \sharp$.

[7] The proof of this theorem comes from an idea of Werner Nutt.

Every conceptual expression C_i corresponds to an existential quantified formula with one free variable. Moreover, the matrices of such formulas are conjunctions of positive predicates. Let us denote the formula corresponding to a concept C_i as $C'_i(x)$. The functionality of feature concept expressions can be expressed with a set F of definite Horn clauses. The restricted thesis holds – in the binary case – if and only if $F \cup \{C_i"(a)\} \models D'_1(a) \vee D'_2(a)$, where a is a constant substituting the free variable x and $C_i"(a)$ is obtained by skolemizing the existential quantified variables. $F \cup \{C_i"(a)\}$ is equivalent to a set of definite Horn clauses, which are characterized by a *minimal Herbrand model*, say $\mathcal{H}_{\mathcal{B}}$. Then:

$$F \cup \{C_i"(a)\} \models D'_1(a) \vee D'_2(a) \quad \text{iff}$$
$$\mathcal{H}_{\mathcal{B}} \models D'_1(a) \vee D'_2(a)$$

Since we are talking of a single model, $D'_1(a) \vee D'_2(a)$ is valid in $\mathcal{H}_{\mathcal{B}}$ if and only if either $D'_1(a)$ or $D'_2(a)$ is valid in $\mathcal{H}_{\mathcal{B}}$. □

4.3 INSTANCE RECOGNITION

Given a knowledge base Σ, an interval i, an individual a and a concept C, the *instance recognition problem* is to test whether $\Sigma \models C(i,a)$, i.e. the inference task checking if the individual is an instance of the concept at the given time. We will only sketch here the procedure for the instance recognition problem. The algorithm first computes the *most specialized concept expression* – $\mathtt{MSC}^a_i$ – for the individual a at the reference interval i. The $\mathtt{MSC}^a_i$ is a concept such that it instantiates the individual at the interval i, i.e. $\Sigma \models \mathtt{MSC}^a_i(i,a)$, and it is the *most* specific one, i.e. for every concept expression D of the language satisfying $\Sigma \models D(i,a)$ then $\mathtt{MSC}^a_i \sqsubseteq D$. The procedure computing the $\mathtt{MSC}^a_i$ is a variant of the well known abstraction procedure of concept languages – see e.g. [Nebel, 1990; Lenzerini and Schaerf, 1991; Donini and Era, 1992]. Even if we conjecture the completeness of the procedure computing the $\mathtt{MSC}^a_i$, no formal proof is available.

Given the most specialized concept of every individual, the instance recognition problem can be reduced to a subsumption test: $\Sigma \models C(i,a)$ if and only if $\mathtt{MSC}^a_i \sqsubseteq C$.

For example, the most specialized concept for the individual action a at the reference interval i_a with respect to the ABox defined in section 3 is the following:

$$\begin{array}{lll} \mathtt{MSC}^a_{i_a} & \doteq & \Diamond(x_1\ x_2\ x_3\ x_4\ x_5) \\ & & (x_1\ \mathsf{o}\ \sharp)(x_2\ \mathsf{fi}\ \sharp)(x_3 \mathsf{mi}\ x_1) \\ & & (x_3\ \mathsf{f}\ \sharp)(x_4\ \mathsf{mi}\ \sharp)(x_5\ \mathsf{mi}\ \sharp). \\ & & \big((\star\mathtt{OBJECT1} : \mathtt{Clear\text{-}Block})@x_1\ \sqcap \\ & & (\star\mathtt{OBJECT1} : \mathtt{Hold\text{-}Block})@x_3\ \sqcap \\ & & (\star\mathtt{OBJECT1} : \mathtt{Clear\text{-}Block})@x_4\ \sqcap \\ & & (\star\mathtt{OBJECT2} : \mathtt{Clear\text{-}Block})@x_2\ \sqcap \\ & & (\star\mathtt{OBJECT1} \circ \mathtt{ON} \downarrow \star\mathtt{OBJECT2})@x_5\big) \end{array}$$

Since $\mathtt{MSC}^a_{i_a} \sqsubseteq \mathtt{Stack}$, the individual action a is an instance of the concept $\mathtt{Stack}$ at the time interval i_a.

5 CONCLUSIONS

We have shown in this paper a formalism for representing time, actions and plans in a uniform way. The proposed temporal concept language allows for the representation of actions in the style of Allen: an action can have parameters, which are the ties with the temporal evolution of the world, and an action is associated over time with other actions. An action taxonomy based on subsumption can be set up, and it can play the role of a plan library for plan retrieval tasks. From the observation of the evolution of the world state, the type of the involved actions can be understood, for plan recognition purposes.

Currently, the language is able to express a plan as an action having possibly different properties through time – i.e. as a *complex activity*. In [Artale and Franconi, 1995] a decomposition operator is introduced to distinguish the different actions composing a plan. A plan can be viewed as a hierarchical structure whose constituent actions could be seen as its distinct decomposing steps. Further research work within this approach includes the treatment of temporally *homogeneous*, *concatenable* and *countable* concepts[8]. Homogeneity is useful to characterize the temporal behavior of world states [Artale *et al.*, 1994]. The language can be successfully extended in order to cope with problems characterized by *inertial* – or *persistent* – properties. In this larger framework, states can be represented as simple non-temporal homogeneous and persistent concepts [Artale and Franconi, 1995].

Acknowledgments

This paper is a revised and extended version of a working paper presented at two IJCAI-93 workshops [Artale and Franconi, 1993]. This work has been partially supported by the Italian National Research Council (CNR), projects "Sistemi Informatici", "Pianificazione Automatica" and "Robotica". This research was partly done while the first author was visiting IRST from University of Florence, Italy. We would like to thank Claudio Bettini, Paolo Bresciani, Alfonso Gerevini, Werner Nutt, Andrea Schaerf, Luciano Serafini, Achille C. Varzi and two anonymous referees. All the errors of the paper are, of course, our own.

References

[Allen, 1991] James F. Allen. Temporal reasoning and planning. In James F. Allen, Henry A. Kautz, Richard N. Pelavin, and Josh D. Tenenberg, editors, *Reasoning about Plans*, chapter 1, pages 2–68. Morgan Kaufmann, 1991.

[8] A concept is *homogeneous* if its instances at some interval are also instances at the subintervals; a concept is *concatenable* if its instances at two meeting intervals are also instances at the union interval; a concept is *countable* if its instances at some interval are not instances at the overlapping intervals.

[Artale and Franconi, 1993] Alessandro Artale and Enrico Franconi. A unified framework for representing time, actions and plans. In F. D. Anger, H. W. Guesgen, and J. van Benthem, editors, *Workshop Notes of the IJCAI-93 Workshop on Temporal and Spatial Reasoning*, pages 193–217, Chambery, France, August 1993. Also in the Workshop Notes of the IJCAI-93 Workshop on Object-Based Representation System, pages 32–44; a shorter version appears in the Proceedings of the Italian Planning Workshop 1993 (IPW'93), pages 167–172, Roma Italy, September 1993.

[Artale and Franconi, 1995] Alessandro Artale and Enrico Franconi. Time, actions and plans representation in a description logic. *International Journal of Intelligent Systems*, 1995. To appear.

[Artale *et al.*, 1994] Alessandro Artale, Claudio Bettini, and Enrico Franconi. Homogeneous concepts in a temporal description logic. In *Proceedings of the International Workshop on Description Logics*, pages 36–41. DFKI, Saarbrücken, Bonn, Germany, May 1994.

[Artale, 1994] Alessandro Artale. *Rappresentazione di Tempo ed Azioni e Ragionamento Tassonomico per Object-Oriented Databases nel Contesto dei Linguaggi Terminologici.* PhD thesis, University of Florence, Italy, February 1994. (in italian).

[Bettini, 1992] Claudio Bettini. A formalization of interval-based temporal subsumption in first order logic. In *Workshop Notes of the ECAI Workshop on Theoretical Foundations of Knowledge Representation and Reasoning*, Vienna, Austria, August 1992.

[Bettini, 1993] Claudio Bettini. *Temporal Extensions of Terminological Languages.* PhD thesis, Computer Science Department, University of Milan, Italy, 1993.

[Devanbu and Litman, 1991] Premkumar T. Devanbu and Diane J. Litman. Plan-based terminological reasoning. In *Proc. of the 2 nd International Conference on Principles of Knowledge Representation and Reasoning (KR-91)*, pages 128–138, Cambridge, MA, May 1991.

[Donini and Era, 1992] Francesco M. Donini and Angelo Era. Most specific concepts for knowledge bases with incomplete information. In *Proc. of CIKM-92*, pages 545–551, 1992.

[Halpern and Shoham, 1991] J. Y. Halpern and Y. Shoham. A propositional modal logic of time intervals. *Journal of ACM*, 38(4):935–962, 1991.

[Heinsohn *et al.*, 1992] Jochen Heinsohn, Daniel Kudenko, Bernhard Nebel, and Hans-Jürgen Profitlich. RAT: representation of actions using terminological logics. Technical report, DFKI, Saarbrücken, Germany, November 1992.

[Hollunder *et al.*, 1990] B. Hollunder, W. Nutt, and M. Schmidt-Schauß. Subsumption algorithms for concept description languages. In *Proc. of the 9 th ECAI*, pages 348–353, Stockholm, Sweden, 1990.

[Kautz, 1991] Henry A. Kautz. A formal theory of plan recognition and its implementation. In James F. Allen, Henry A. Kautz, Richard N. Pelavin, and Josh D. Tenenberg, editors, *Reasoning about Plans*, chapter 2, pages 69–126. Morgan Kaufmann, 1991.

[Lambrix and Rönnquist, 1993] Patrick Lambrix and Ralph Rönnquist. Terminological logic involving time and evolution: A preliminary report. In *Proceedings of ISMIS-93*, 1993.

[Lenzerini and Schaerf, 1991] M. Lenzerini and A. Schaerf. Concept languages as query languages. In *Proc. of AAAI-91*, pages 471–476, Anaheim, CA, 1991.

[Lifschitz, 1987] Vladimir Lifschitz. On the semantics of STRIPS. In *The 1986 Workshop on Reasoning about Actions and Plans*, pages 1–10. Morgan Kaufman, 1987.

[McCarthy and Hayes, 1969] J. McCarthy and P. J. Hayes. Some philosophical problems from the standpoint of Artificial Intelligence. In B. Meltzer and D. Michie, editors, *Machine Intelligence*, volume 4, pages 463–502, Edinburgh, UK, 1969. Edinburgh University Press.

[Nebel, 1990] B. Nebel. *Reasoning and Revision in Hybrid Representation Systems*, volume 422 of *Lecture Notes in Artificial Intelligence.* Springer-Verlag, Berlin, Heidelberg, New York, 1990.

[Schild, 1993] Klaus D. Schild. Combining terminological logics with tense logic. In *Proceedings of the 6^{th} Portuguese Conference on Artificial Intelligence, EPIA'93*, October 1993.

[Schmiedel, 1990] A. Schmiedel. A temporal terminological logic. In *Proc. of AAAI-90*, pages 640–645, Boston, MA, 1990.

[Song and Cohen, 1991] Fei Song and Robin Cohen. Temporal reasoning during plan recognition. In *Proc. of AAAI-91*, pages 247–252, Anaheim, CA, 1991.

[van Beek and Cohen, 1990] P. van Beek and R. Cohen. Exact and approximate reasoning about temporal relations. *Computational Intelligence*, 6:132–144, 1990.

[Venema, 1990] Yde Venema. Expressiveness and completeness of an interval tense logic. *Notre Dame Journal of Formal Logic*, 31(4):529–547, Fall 1990.

[Weida and Litman, 1992] Robert Weida and Diane Litman. Terminological reasoning with constraint networks and an application to plan recognition. In *Proc. of the 3 rd International Conference on Principles of Knowledge Representation and Reasoning (KR-92)*, pages 282–293, Cambridge, MA, October 1992.

[Weida and Litman, 1994] Robert Weida and Diane Litman. Subsumption and recognition of heterogeneous constraint networks. In *Proceedings of CAIA-94*, 1994.