A Survey of Temporal Extensions of Description Logics

Alessandro Artale [a] Enrico Franconi [b]

[a] *Department of Computation – UMIST, Manchester, UK.*
E-mail: artale@co.umist.ac.uk
[b] *Department of Computer Science – Univ. of Manchester, UK.*
E-mail: franconi@cs.man.ac.uk

This paper surveys the temporal extensions of Description Logics appeared in the literature. The analysis considers a large spectrum of approaches appeared in the Temporal Description Logics area: from the loosely coupled approaches – which comprise, for example, the enhancement of simple Description Logics with a constraint based mechanism – to the most principled ones – which consider a combined semantics for the abstract and the temporal domains. It will be shown how these latter approaches have a strict connection with temporal logics.

Advantages of using Temporal Description Logics are their high expressivity combined with desirable computational properties – such as decidability, soundness and completeness of deduction procedures. In this survey the computational properties of various families of Temporal Description Logics will be pointed out.

Keywords: Description Logics, Temporal Logics

AMS Subject classification: ???

1. Introduction

Description Logics[1] are formalisms designed for the logical reconstruction and the extension of representational tools such as object-oriented data models, semantic data models (e.g., Extended Entity/Relationship), frame-based ontology languages, and semantic networks, KL-ONE-like languages, type systems, and feature logics. Nowadays, Description Logics are also considered the most important unifying formalism for the many object-centered representation languages used in areas other than Knowledge Representation. Important characteristics of Description Logics are high expressivity together with decidability, which guarantee that reasoning algorithms always terminate with the correct answers.

Temporal extensions of modeling formalisms are relevant to capture the evolving behavior of dynamic domains, and they have been extensively considered in Artificial Intelligence. In the Description Logic literature, several approaches

[1] Description Logics have been also called *Frame-Based Description Languages*, *Term Subsumption Languages*, *Terminological Logics*, *Taxonomic Logics*, *Concept Languages* or KL-ONE-*like languages*.

for representing and reasoning with time dependent concepts have been proposed. These temporal extensions differ from each others in different ways. They differ on the ontology of time, whether they adopt an interval-based or a point-based notion of time. They differ on how the temporal dimension is handled, i.e., whether an *explicit* notion of time is adopted in which temporal operators are used to build new formulæ, or the temporal information is only *implicit* in the language by resorting to a state-change style of representation to denote sequences of events. In the case of an explicit representation of time, there is a further distinction – introduced by Finger and Gabbay [23] – between an *external* and an *internal* point of view. In the *external* method the very same individual can have different "snapshots" in different moments of time that describe the various states of the individual at these times. In this latter case, the representation language can be seen in a modular way where two different logics are combined: while an atemporal part of the language describes the "static" aspects, the temporal part relates the different snapshots – describing in such a way the "dynamic" aspects. In the *internal* method the different states of an individual are seen as different individual components: an individual is a collection of distinct temporal "parts" each one holding at a particular moment.

This paper is organized as follows. After introducing the main features of Description Logics in Section 2, and the interval temporal logic $\mathcal{HS}$ in Section 3, we propose to group the surveyed papers in four different areas: Sections 4 and 5 illustrate what we called the external method where an interval-based or a point-based temporal logic is combined with a Description Logic. Section 6 describes the languages using an *internal* representation of time, and Section 7 presents the extensions which avoid any explicit temporal representation – like the state-change based approaches. Section 8 concludes the paper.

2. Description Logics

In this section the formal framework of Description Logics is briefly presented. The presentation of the formal apparatus will strictly follow the $\mathcal{ALC}$ notation introduced by Schmidt-Schauss and Smolka [41] and summarized in [21,14]: in this perspective, Description Logics are considered as a *structured* fragment of predicate logic. $\mathcal{ALC}$ is the minimal Description Logic including full negation and disjunction – i.e., it is propositionally closed. In this Section, we will consider the language $\mathcal{ALCF}$ [31], extending $\mathcal{ALC}$ with functions. In the rest of the paper several variants of this basic Description Logic will be introduced.

The basic types of $\mathcal{ALCF}$ are *concepts*, *roles*, and *features*. A concept is a description gathering the common properties among a collection of individuals; from a logical point of view it is a unary predicate ranging over the domain of individuals. Properties are represented either by means of roles – which are interpreted as binary relations associating to individuals of a given class values

$$
\begin{array}{rll}
C, D \rightarrow & A \mid & \text{(atomic concept)} \\
& \top \mid & \text{(top)} \\
& \bot \mid & \text{(bottom)} \\
& \neg C \mid & \text{(complement)} \\
& C \sqcap D \mid & \text{(conjunction)} \\
& C \sqcup D \mid & \text{(disjunction)} \\
& \forall P . C \mid & \text{(universal quantifier)} \\
& \exists P . C \mid & \text{(existential quantifier)} \\
& p : C & \text{(selection)} \\
& p \downarrow q \mid & \text{(agreement)} \\
& p \uparrow q \mid & \text{(disagreement)} \\
& p \uparrow \mid & \text{(undefinedness)} \\
p, q \rightarrow & f \mid & \text{(atomic feature)} \\
& p \circ q & \text{(path)}
\end{array}
$$

Figure 1. Syntax rules for the $\mathcal{ALCF}$ Description Logic.

for that property – or by means of features – which are interpreted as functions associating to individuals of a given class a *single* value for that property.

According to the syntax rules of Figure 1, $\mathcal{ALCF}$ *concepts* (denoted by the letters C and D) are built out of *atomic concepts* (denoted by the letter A), *atomic roles* (denoted by the letter P), and *atomic features* (denoted by the letter f). The syntax rules are expressed following the tradition of Description Logics [8]: they can be read as, e.g., if C and D are concept expressions then $C \sqcap D$ is a concept expression, too.

Let us now consider the formal semantics. We define the *meaning* of concept expressions as sets of individuals – as for unary predicates – and the meaning of roles as sets of pairs of individuals – as for binary predicates. Formally, an *interpretation* is a pair $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consisting of a set $\Delta^{\mathcal{I}}$ of individuals (the *domain* of $\mathcal{I}$) and a function $\cdot^{\mathcal{I}}$ (the *interpretation function* of $\mathcal{I}$) mapping every concept to a subset of $\Delta^{\mathcal{I}}$, every role to a subset of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, and every feature to a partial function from $\Delta^{\mathcal{I}}$ to $\Delta^{\mathcal{I}}$, such that the equations of the left column in Figure 2 are satisfied. The semantics of the language can also be given by stating equivalences among expressions of the language and open First Order Logic formulæ. An atomic concept A, an atomic role P, and an atomic feature f, are mapped respectively to the open formulæ $A(\alpha)$, $P(\alpha, \beta)$, and $f(\alpha, \beta)$ – with f a functional relation, also written $f(\alpha) = \beta$, and α, β denoting the free variables. The rightmost column of Figure 2 gives the transformational semantics of $\mathcal{ALCF}$ expressions in terms of equivalent FOL well-formed formulæ. A concept C, a role P and a path p correspond to the FOL open formulæ$F_C(\alpha)$, $F_P(\alpha, \beta)$, and $F_p(\alpha, \beta)$, respectively. It is worth noting that, using the standard model-theoretic semantics, the extensional semantics of the left column can be derived from the transformational semantics of the right column.

$\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$	true
$\bot^{\mathcal{I}} = \emptyset$	false
$(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$	$\neg F_C(\alpha)$
$(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$	$F_C(\alpha) \wedge F_D(\alpha)$
$(C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$	$F_C(\alpha) \vee F_D(\alpha)$
$(\exists P.C)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \mid \exists b.(a,b) \in P^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\}$	$\exists x.F_P(\alpha,x) \wedge F_C(x)$
$(\forall P.C)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \mid \forall b.(a,b) \in P^{\mathcal{I}} \rightarrow b \in C^{\mathcal{I}}\}$	$\forall x.F_P(\alpha,x) \rightarrow F_C(x)$
$(p : C)^{\mathcal{I}} = \{a \in dom\, p^{\mathcal{I}} \mid p^{\mathcal{I}}(a) \in C^{\mathcal{I}}\}$	$\exists x.F_p(\alpha,x) \wedge F_C(x)$
$p \downarrow q^{\mathcal{I}} = \{a \in dom\, p^{\mathcal{I}} \cap dom\, q^{\mathcal{I}} \mid p^{\mathcal{I}}(a) = q^{\mathcal{I}}(a)\}$	$\big(\exists x.F_p(\alpha,x) \wedge F_q(\alpha,x)\big) \wedge$ $\big(\forall x,y.F_p(\alpha,x) \wedge F_q(\alpha,y) \rightarrow x = y\big)$
$p \uparrow q^{\mathcal{I}} = \{a \in dom\, p^{\mathcal{I}} \cap dom\, q^{\mathcal{I}} \mid p^{\mathcal{I}}(a) \neq q^{\mathcal{I}}(a)\}$	$\big(\exists x,y.F_p(\alpha,x) \wedge F_q(\alpha,y)\big) \wedge$ $\big(\forall x,y.F_p(\alpha,x) \wedge F_q(\alpha,y) \rightarrow x \neq y\big)$
$(p\uparrow)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus dom\, p^{\mathcal{I}}$	$\neg\exists x.F_p(\alpha,x)$
$(p \circ q)^{\mathcal{I}} = p^{\mathcal{I}} \circ q^{\mathcal{I}}$	$\exists x.F_p(\alpha,x) \wedge F_q(x,\beta)$

Figure 2. The extensional and transformational semantics in $\mathcal{ALCF}$.

Description Logics allow to express *Knowledge Bases* (KB) by means of two kinds of logical axioms. The *intensional* axioms express generic knowledge about the concepts and roles in the KB. The *extensional* axioms constrain the way concept and roles are instantiated. Intensional knowledge is conveyed through a *terminology* or a *TBox* expressed as finite set of *terminological axioms*. For a concept name A, and (possibly complex) concepts C, D, terminological axioms are of the form $A \doteq C$ (concept definition), $A \sqsubseteq C$ (primitive concept definition), $C \sqsubseteq D$ (general inclusion statement). An interpretation $\mathcal{I}$ satisfies $C \sqsubseteq D$ if and only if the interpretation of C is included in the interpretation of D, i.e., $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. It is clear that the last kind of axiom is a generalization of the first two: concept definitions of the type $A \doteq C$ – where A is an atomic concept – can be reduced to the pair of axioms $(A \sqsubseteq C)$ and $(C \sqsubseteq A)$. An interpretation $\mathcal{I}$ is a model for a *TBox* $\mathcal{T}$ if $\mathcal{I}$ satisfies all the terminological axioms in $\mathcal{T}$. Unless explicitly noted, in the following we will consider *acyclic simple* TBoxes only: there are only concept definitions, and a defined concept may appear at most once as the left-hand side of an axiom, and no terminological cycles are allowed, i.e., no defined concepts may occur – neither directly nor indirectly – within its own definition [37]. As an example of the expressive power of an $\mathcal{ALCF}$ *TBox*, we can consider the class denoting "happy fathers", defined using the atomic predicates `Man`, `Woman`, `Doctor`, `Rich`, `Famous` (concepts), `CHILD`, `FRIEND` (roles) and `WIFE` (feature):

$$\texttt{HappyFather} \doteq \texttt{Man} \sqcap (\texttt{WIFE} : \texttt{Woman}) \sqcap (\exists\texttt{CHILD}.\top) \sqcap \forall\texttt{CHILD}.(\texttt{Doctor} \sqcap \exists\texttt{FRIEND}.(\texttt{Rich} \sqcup \texttt{Famous}))$$

i.e., the men with a wife (exactly one!) whose children are doctors having some rich or famous friend.

Extensional knowledge is expressed by means of an *ABox* which is formed by a finite set of *assertional axioms*, i.e. predications on individual objects. Let $\mathcal{O}$ be the alphabet of symbols denoting *individuals*; an assertion is an axiom of the form $C(a)$, $R(a,b)$ or $p(a,b)$, where a and b denote individuals in $\mathcal{O}$. The interpretation $\mathcal{I}$ is extended over individuals in such a way 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). $C(a)$ is satisfied by an interpretation $\mathcal{I}$ iff $a^{\mathcal{I}} \in C^{\mathcal{I}}$, $P(a,b)$ is satisfied by $\mathcal{I}$ iff $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in P^{\mathcal{I}}$, and $p(a,b)$ is satisfied by $\mathcal{I}$ iff $p^{\mathcal{I}}(a^{\mathcal{I}}) = b^{\mathcal{I}}$. An interpretation $\mathcal{I}$ is a model for an *ABox* $\mathcal{A}$ if $\mathcal{I}$ satisfies all the assertional axioms in $\mathcal{A}$.

For example, the individual `john`, as defined by the following *ABox*, could be recognised as an `HappyFather`:

```
Man(john), WIFE(john,mary), CHILD(john,bill),
Doctor(bill), FRIEND(bill,peter), Rich(peter).
```

A *knowledge base* is a finite set Σ of terminological and assertional axioms (i.e., $\Sigma = \langle TBox, ABox \rangle$). An interpretation $\mathcal{I}$ is a *model* of a knowledge base Σ iff every axiom of Σ is satisfied by $\mathcal{I}$.

Let us describe now the basic reasoning services provided by a DL-system. From Σ does not *logically follows* that $C \doteq \bot$ (written $\Sigma \not\models C \doteq \bot$) if there exists a model $\mathcal{I}$ of Σ such that $C^{\mathcal{I}} \neq \emptyset$: we say that C is *satisfiable* and we indicate this reasoning problem as *concept satisfiability.* Σ *logically implies* $D \sqsubseteq C$ (written $\Sigma \models D \sqsubseteq C$) if $D^{\mathcal{I}} \subseteq C^{\mathcal{I}}$ for every model of Σ: we say that D is *subsumed* by C in Σ. The reasoning problem of checking whether D is *subsumed* by C in Σ is called *subsumption checking.* We write $\Sigma \not\models$ to indicate the problem of checking whether Σ has a model, a problem called *knowledge base consistency.* Σ *logically implies* $C(a)$ (written $\Sigma \models C(a)$) if $a^{\mathcal{I}} \in C^{\mathcal{I}}$ for every model of Σ: we say that a is an *instance* of C in Σ. The reasoning problem of checking whether a is an *instance* of C in Σ is called *instance checking.* Notice that for propositionally complete languages we have that $\Sigma \models D \sqsubseteq C$ if and only if $\Sigma \models D \sqcap \neg C \doteq \bot$, and $\Sigma \models C(a)$ if and only if $\Sigma \cup \{\neg C(a)\} \not\models$. In other words, subsumption can be reduced to satisfiability and instance checking to knowledge base consistency.

An acyclic simple TBox can be transformed into an *expanded* TBox having the same models, where no defined concepts make use in their definitions of any other defined concept. In this way, the interpretation of a defined concept in an expanded TBox does not depend from any other defined concept. It is easy to see that D is subsumed by C in Σ with an acyclic simple TBox if and only if the expansion of D (w.r.t. Σ) is subsumed by the expansion of C (w.r.t. Σ) in $\Sigma' = \langle \emptyset, ABox \rangle$ (i.e., the knowledge base with an the empty TBox). The expansion procedure recursively substitutes every defined concept occurring in a definition with its defining expression; such a procedure may generate a TBox exponential in size, but it was proved [36] that it could remain polynomial under reasonable restrictions. In the following we will interchangeably refer either to reasoning with respect to a TBox or to reasoning involving expanded concepts with respect to an empty TBox.

2.1. Correspondence with Modal Logics

Schild [39] proved the correspondence between the Description Logic $\mathcal{ALC}$ and the propositional multi-modal logic $\boldsymbol{K_{(m)}}$ [25,27]. $\boldsymbol{K_{(m)}}$ is the simplest normal multi-modal logic interpreted over Kripke structures: there is more than one modal accessibility relation, each one independently behaving as a $\boldsymbol{K}$ accessibility relation. Informally, a concept corresponds to a propositional formula, and it is interpreted as the set of possible worlds over which the formula holds. The existential and universal quantifiers correspond to the possibility and necessity operators over different accessibility relations: $\Box_r C$ is interpreted as the set of all the possible worlds such that in every r-accessible world C holds; $\Diamond_r C$ is interpreted as the set of all the possible worlds such that in some r-accessible world C holds. Thus, roles are interpreted as the accessibility relations between worlds. A knowledge base includes also constraints on the Kripke structures through the ABox, by stating which are the necessary relations between worlds, and which are the formulæ necessarily holding in some world. Thus, we can speak of satisfiability of a formula ϕ of $K_{(m)}$ *with respect to* a set of world constraints Σ.

Starting from the work of Schild, the work presented in [14] analyses a very expressive modal logic, which extends the expressivity of *PDL* – i.e. the propositional dynamic modal logic [29] – with the converse operator and graded modalities. They have proved the correspondence with a very expressive description logic (called $\mathcal{ALCQI}_{reg}$ or $\mathcal{CIQ}$), which includes $\mathcal{ALC}$, qualified cardinality restrictions, inverse roles, and *regular expressions* over roles. Reasoning in $\mathcal{ALCQI}_{reg}$ is decidable, and it has been proved to be an EXPTIME-complete problem [14].

3. The Temporal Logic $\mathcal{HS}$

Since Description Logics are in a strict correspondence with propositional modal logics, the interval-based temporal extensions of Description Logics that follow an *external* approach can be seen as the combination of a propositional modal logic with the interval-based temporal logic introduced by Halpern and Shoham [28] – we indicate this temporal logic as $\mathcal{HS}$.

Well-formed formulæ of $\mathcal{HS}$ are built by augmenting the propositional calculus with the modal temporal operators corresponding to the Allen interval relations [1] (figure 3): 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). In practice, only the *starts* and *finishes* relation-

Relation	Abbr.	Inverse
before(i, j)	b	a
meets(i, j)	m	mi
overlaps(i, j)	o	oi
starts(i, j)	s	si
during(i, j)	d	di
finishes(i, j)	f	fi

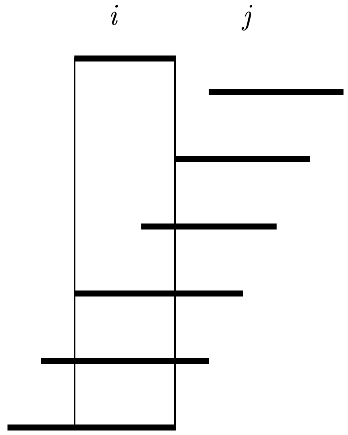

Figure 3. The Allen's interval relationships.

ships and their inverses are needed: $\langle$starts$\rangle$, $\langle$finishes$\rangle$, $\langle$started-by$\rangle$, $\langle$finished-by$\rangle$. The semantics of the language is based on Kripke structure whose domain is a set of intervals, and formulæ are interpreted as sets of intervals – we say that a formula holds at an interval if it is evaluated as true at that interval. In carrying on the interpretation process there is an implicit use of the reference interval, i.e., the actual evaluation interval – called *now*. The modal operators relate the *now* interval with other intervals. Intuitively, the meaning of an $\mathcal{HS}$ formula is the following:

$\langle$starts$\rangle\phi$ is true iff ϕ holds at some interval starting the reference interval
$\langle$finishes$\rangle\phi$ is true iff ϕ holds at some interval finishing the reference interval.

while the modal operators $\langle$started-by$\rangle$, $\langle$finished-by$\rangle$ introduce the inverse temporal relations. It is also possible to define the duals of these operators as usual: $[X] \doteq \neg\langle X\rangle\neg\phi$ – where X stands for a temporal relation. For example, [starts] ϕ says that ϕ is true at all beginning intervals.

It is worth noting that, the thirteen temporal relations can be simulated by using the above mentioned four modal operators provided that the temporal structure allows for point intervals [44][2]. Indeed, if both a *beginning point* and an *ending point* modal operators can be defined in the following way:

$$[\![BP]\!]\phi \doteq ([\mathsf{starts}]\bot \wedge \phi) \vee \langle\mathsf{starts}\rangle\,([\mathsf{starts}]\bot \wedge \phi)$$
$$[\![EP]\!]\phi \doteq ([\mathsf{finishes}]\bot \wedge \phi) \vee \langle\mathsf{finishes}\rangle\,([\mathsf{finishes}]\bot \wedge \phi)$$

which allows to define the *meets* and *met-by* modal operators as: $\langle\mathsf{meets}\rangle\phi \doteq [\![EP]\!]\langle\mathsf{started\text{-}by}\rangle\phi$ and $\langle\mathsf{met\text{-}by}\rangle\phi \doteq [\![BP]\!]\langle\mathsf{finished\text{-}by}\rangle\phi$. For the other temporal relations the following definitions hold:

[2] If the temporal structure allows only for proper intervals then the *meets* operator is also needed as a basic modal operator.

$$\begin{aligned}
\neg\phi^{\mathcal{I}} &= \mathcal{T}^{\star}_{<} \setminus \phi^{\mathcal{I}} \\
(\phi \wedge \psi)^{\mathcal{I}} &= \phi^{\mathcal{I}} \cap \psi^{\mathcal{I}} \\
(\langle\mathsf{starts}\rangle\phi)^{\mathcal{I}} &= \{[t_1,t_2] \in \mathcal{T}^{\star}_{<} \mid \exists u. t_1 \leq u < t_2 \text{ and } [t_1,u] \in \phi^{\mathcal{I}}\} \\
(\langle\mathsf{started\text{-}by}\rangle\phi)^{\mathcal{I}} &= \{[t_1,t_2] \in \mathcal{T}^{\star}_{<} \mid \exists u. t_2 < u \text{ and } [t_1,u] \in \phi^{\mathcal{I}}\} \\
(\langle\mathsf{finishes}\rangle\phi)^{\mathcal{I}} &= \{[t_1,t_2] \in \mathcal{T}^{\star}_{<} \mid \exists u. t_1 < u \leq t_2 \text{ and } [u,t_2] \in \phi^{\mathcal{I}}\} \\
(\langle\mathsf{finished\text{-}by}\rangle\phi)^{\mathcal{I}} &= \{[t_1,t_2] \in \mathcal{T}^{\star}_{<} \mid \exists u. u < t_1 \text{ and } [u,t_2] \in \phi^{\mathcal{I}}\}
\end{aligned}$$

Figure 4. $\mathcal{HS}$'s semantics

$$\begin{aligned}
\langle\mathsf{during}\rangle\phi &\doteq \langle\mathsf{starts}\rangle\langle\mathsf{finishes}\rangle\phi \\
\langle\mathsf{before}\rangle\phi &\doteq \langle\mathsf{meets}\rangle\langle\mathsf{meets}\rangle\phi \\
\langle\mathsf{overlaps}\rangle\phi &\doteq \langle\mathsf{starts}\rangle\langle\mathsf{finished\text{-}by}\rangle\phi
\end{aligned}$$

As an example, the notion of `Mortal` can be expressed in this logic as: `LivingBeing` $\wedge\, \langle\mathsf{met\text{-}by}\rangle\neg$`LivingBeing`, with the meaning of a `LivingBeing` who will not be alive in some interval met by the current interval.

For what concerns the semantics, this tense logic is provided with a Tarski-style semantics. A linear and unbounded temporal structure $\mathcal{T} = (\mathcal{P}, <)$ is assumed, where $\mathcal{P}$ is a set of time points and $<$ is a strict partial order on $\mathcal{P}$. The *interval set* of a structure $\mathcal{T}$ is defined as the set $\mathcal{T}^{\star}_{<}$ of all closed intervals $[t_1, t_2] \doteq \{x \in \mathcal{P} \mid t_1 \leq x \leq t_2\}$ in $\mathcal{T}$. An *interpretation* $\mathcal{I} \doteq \langle \mathcal{T}^{\star}_{<}, \cdot^{\mathcal{I}} \rangle$ consists of a set $\mathcal{T}^{\star}_{<}$ (the *interval set* of the selected temporal structure $\mathcal{T}$), and a function $\cdot^{\mathcal{I}}$ which maps each primitive proposition into a set of closed intervals where it is true – i.e., $p^{\mathcal{I}} \subseteq \mathcal{T}^{\star}_{<}$. The interpretation of generic formulæ in $\mathcal{HS}$ is inductively defined, as reported in Figure 4. As usual, a formula ϕ is said *satisfiable* if there exists an interpretation $\mathcal{I}$ such that $[t_1, t_2] \in \phi^{\mathcal{I}}$ for some $[t_1, t_2] \in \mathcal{T}^{\star}_{<}$. A formula ϕ is said *valid* with respect to a class of temporal structures $\mathcal{A}$ if $\neg\phi$ is not satisfiable in $\mathcal{A}$.

Halpern and Shoham prove many interesting complexity results concerning the validity and satisfiability problems. It is worth to note how these results depend from the underlying temporal structure. The validity problem ranges from decidable to Π^1_1-hard; for what concerns the satisfiability problem its complexity class can be obtained by observing that it is the complement of the validity problem. The bad result is summarized by the following statement:

> "One gets decidability only in very restricted cases, such as when the set of temporal models considered is a finite collection of structures, each consisting of a finite set of natural numbers (since in this case one can simply perform an exhaustive check on all structures)." [28]

To present the complexity results in a formal way we introduce the notion of an *infinitely ascending sequence*. A temporal structure is said to contain an *infinitely ascending sequence* if it contains an infinite sequence of points $t_0, t_1, \ldots$ such that $t_i < t_{i+1}$. The critical complexity result is stated by the following proposition.

Proposition 3.1. The validity problem for any class of temporal structures that contains an *infinitely ascending sequence* is r.e.-hard.

This was proved by constructing tense formulæ that encode the computation of a Turing machine. The next theorem summarizes the complexity results provided by Halpern and Shoham.

Theorem 3.2. The validity problem for all dense, linear and unbounded classes of temporal structures is r.e.-complete. The validity problem for $\mathcal{Q}$ is r.e.-complete. The validity problem for $\mathcal{N}$ is Π^1_1-complete. The validity problem for $\mathcal{R}$ is in Π^2_1.

4. Interval-based Temporal Description Logics

In this Section we present various proposals that extend Description Logics with an explicit interval-based notion of time following the *external* approach. The resulting logics are the combination of a static DL with a temporal logic. These logics have in the interval-based temporal logics introduced by [28] their natural ancestor.

4.1. Schmiedel's Formalism

Schmiedel [42] was the first to propose an extension of Description Logics with an interval-based temporal logic. The underlying Description Logic is the prefix version of the $\mathcal{FLENR}^-$ language[3] [20], while the new term-forming operators are the temporal qualifier at, the existential and universal temporal quantifiers sometime and alltime. The at operator specifies the time at which a concept holds while sometime and alltime are temporal quantifiers introducing temporal variables. Temporal variables are constrained by means of temporal relationships based on Allen's interval algebra extended with metric constraints in order to deal with durations, absolute times and granularities of intervals. Figure 5 shows the syntax of the temporal extension proposed by Schmiedel. To give an example of this temporal Description Logic, the concept of `Mortal` can be defined as:

$$\begin{aligned}\texttt{Mortal} \doteq\ & \texttt{LivingBeing}\ \textbf{and}\\ & (\textbf{sometime}\ (x)\ (\textsf{met-by}\ x\ now).(\textbf{at}\ x\ (\textbf{not}\ \texttt{LivingBeing})))\end{aligned}$$

[3] Note that $\mathcal{FLE}^-$ differs from $\mathcal{ALC}$ in that neither it contains the concepts $\top$ and $\bot$ nor it allows for complement or disjunction; the letter $\mathcal{N}$ stands for cardinality restrictions on roles, while $\mathcal{R}$ indicates the role conjunction operator.

$$
\begin{array}{rcl}
<\mathit{concept}> & ::= & <\mathit{atomic\text{-}concept}> \\
& | & (\textsf{and}\ <\mathit{concept}>^{+}) \\
& | & (\textsf{all}\ <\mathit{role}>\ <\mathit{concept}>) \\
& | & (\textsf{atleast}\ \mathit{min}\ <\mathit{role}>) \\
& | & (\textsf{atmost}\ \mathit{max}\ <\mathit{role}>) \\
& | & (\textsf{at}\ <\mathit{interval}>\ <\mathit{concept}>) \\
& | & (\textsf{sometime}\ (<\mathit{interval\text{-}variable}>^{+})\ <\mathit{time\text{-}net}>\,.\,<\mathit{concept}>) \\
& | & (\textsf{alltime}\ (<\mathit{interval\text{-}variable}>^{+})\ <\mathit{time\text{-}net}>\,.\,<\mathit{concept}>) \\
<\mathit{atomic\text{-}concept}> & ::= & \mathit{symbol} \\
<\mathit{role}> & ::= & <\mathit{atomic\text{-}role}> \\
& | & (\textsf{and}\ <\mathit{role}>^{+}) \\
& | & (\textsf{domain}\ <\mathit{concept}>) \\
& | & (\textsf{range}\ <\mathit{concept}>) \\
& | & (\textsf{at}\ <\mathit{interval}>\ <\mathit{role}>) \\
& | & (\textsf{sometime}\ (<\mathit{interval\text{-}variable}>^{+})\ <\mathit{time\text{-}net}>\,.\,<\mathit{role}>) \\
& | & (\textsf{alltime}\ (<\mathit{interval\text{-}variable}>^{+})\ <\mathit{time\text{-}net}>\,.\,<\mathit{role}>) \\
<\mathit{atomic\text{-}role}> & ::= & \mathit{symbol} \\
<\mathit{time\text{-}net}> & ::= & <\mathit{time\text{-}constraint}> \\
& | & (\textsf{and}\ <\mathit{time\text{-}constraint}>^{+}) \\
<\mathit{time\text{-}constraint}> & ::= & (<\mathit{interval\text{-}relation}>\ <\mathit{interval}>\ <\mathit{interval}>) \\
& | & (<\mathit{comparison}>\ <\mathit{interval}>\ <\mathit{duration\text{-}constant}>) \\
& | & (<\mathit{granularity}>\ <\mathit{interval}>) \\
<\mathit{interval\text{-}relation}> & ::= & \textsf{equal} \mid \textsf{meets} \mid \textsf{met-by} \mid \textsf{after} \mid \textsf{before} \\
& | & \textsf{overlaps} \mid \textsf{overlapped-by} \mid \textsf{starts} \mid \textsf{started-by} \\
& | & \textsf{finishes} \mid \textsf{finished-by} \mid \textsf{during} \mid \textsf{contains} \\
& | & (\textsf{or}\ <\mathit{interval\text{-}relation}>^{+}) \\
<\mathit{comparison}> & ::= & < \mid \leq \mid = \mid \geq \mid > \\
<\mathit{granularity}> & ::= & \mathit{sec} \mid \mathit{min} \mid \mathit{hour} \mid \ldots \\
<\mathit{interval}> & ::= & <\mathit{interval\text{-}variable}> \mid <\mathit{interval\text{-}constant}> \mid \mathit{now} \\
<\mathit{interval\text{-}variable}> & ::= & \mathit{symbol} \\
<\mathit{interval\text{-}constant}> & ::= & \mathit{symbol} \\
<\mathit{duration\text{-}constant}> & ::= & \mathit{symbol}
\end{array}
$$

Figure 5. Syntax rules for the Schmiedel proposal.

with the meaning of a `LivingBeing` at the reference interval *now*, who will not be alive at some interval x met by the reference interval *now*. A concept denotes a set of pairs, $\langle i, a\rangle$, composed by a temporal interval and an individual. With the use of the at temporal operator it is possible to bind the evaluation time of a concept to a particular interval of time: (at $'1993'$ `Student`) denotes the set of persons that were students during *1993*.

The expressive power of the language is a direct consequence of the introduction of temporal variables constrained by temporal relations. In this way the logic is able to express abstract temporal patterns. Temporal variables are introduced by the quantifiers sometime and alltime together with a set of constraints – indicated as $<$ *time-net* $>$ in the syntax. There are three kinds of temporal constraints: *qualitative relations* between pairs of intervals by using the Allen algebra, *metric constraints* on a single interval, and *granularity constraints* where an interval is required to take values that are multiples of some time unit. The following example shows a $<$ *time-net* $>$ which makes use of the three kinds of constraints:

(and (day x)(= x $'24h'$)
(day y)(= y $'24h'$)
(meets x y)
(or starts finishes during) x *now*)
(or starts finishes during) y *now*))

where x and y are two consecutive days within *now*. The two constraints (day x) and (= x $'24h'$) restrict x to be coincident with a calendar day. Note that, without the constraint (day x), x could be any interval spanning 24 hours, but not necessarily a full day of the calendar. On the other hand, leaving away the duration constraint, x could take any value that is started and finished by a full day.

Let us show now the model-theoretic semantics. The author assumes a discrete temporal structure over the integers $\mathcal{T} = (\mathcal{Z}, <)$, where $<$ is a strict partial order over $\mathcal{Z}$. The *interval set* $\mathcal{T}^{\star}_{<}$ is defined as the set of closed intervals $[t_1, t_2] \doteq \{x \in \mathcal{Z} \mid t_1 \leq x \leq t_2, t_1 \neq t_2\}$ in $\mathcal{T}$ – in the following the letter i will be used to denote time intervals. For the temporal relations a fixed *temporal model* $\mathcal{M} \doteq \{\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3, \mathcal{M}_4, \mathcal{M}_5\}$ is assumed such that:

$$\begin{aligned}
&\mathcal{M}_1 : \text{Interval-constant} \mapsto \mathcal{T}^{\star}_{<} \\
&\mathcal{M}_2 : \text{Duration-constant} \mapsto 2^{\mathcal{T}^{\star}_{<}} \\
&\mathcal{M}_3 : \text{Comparison-operator} \mapsto 2^{\mathcal{T}^{\star}_{<} \times \mathcal{T}^{\star}_{<}} \\
&\mathcal{M}_4 : \text{Interval-relation} \mapsto 2^{\mathcal{T}^{\star}_{<} \times \mathcal{T}^{\star}_{<}} \\
&\mathcal{M}_5 : \text{Granularity-predicates} \mapsto 2^{\mathcal{T}^{\star}_{<}}
\end{aligned}$$

Such an interpretation preserves the intuitive meaning of the various temporal constructs – e.g., $\langle \mathcal{M}_1['August1990'], \mathcal{M}_1['September1990'] \rangle \in \mathcal{M}_4[\textsf{meets}]$ and $\mathcal{M}_1['3/12/1990'] \in \mathcal{M}_5[\textsf{day}]$. Since concepts can have temporal variables, a *variable assignment* function $\mathcal{V} : \overline{X} \mapsto \mathcal{T}^{\star}_{<}$, where $\overline{X}$ denotes a set of temporal variables, is introduced. To give a meaning to a *time-net*, the *temporal interpretation function* is introduced: $\langle \overline{X}, \overline{Tc} \rangle^{\mathcal{E}}$ – where $\overline{Tc}$ is a set of temporal constraints – is the set of all possibles variable assignments which satisfy the temporal relations

$$
\begin{aligned}
(\mathsf{and}\ C_1 \ldots C_n)^{\mathcal{I}}_{\mathcal{V},i} &= \bigcap_{i=1}^{n} (C_i)^{\mathcal{I}}_{\mathcal{V},i} \\
(\mathsf{all}\ R\ C)^{\mathcal{I}}_{\mathcal{V},i} &= \{a \in \Delta^{\mathcal{I}} \mid \forall b.(a,b) \in R^{\mathcal{I}}_{\mathcal{V},i} \Rightarrow b \in C^{\mathcal{I}}_{\mathcal{V},i}\} \\
(\mathsf{atleast}\ m\ R)^{\mathcal{I}}_{\mathcal{V},i} &= \{a \in \Delta^{\mathcal{I}} \mid \ \ |\{b \mid (a,b) \in R^{\mathcal{I}}_{\mathcal{V},i}\}| \geq m\} \\
(\mathsf{atmost}\ m\ R)^{\mathcal{I}}_{\mathcal{V},i} &= \{a \in \Delta^{\mathcal{I}} \mid \ \ |\{b \mid (a,b) \in R^{\mathcal{I}}_{i}\}| \leq m\} \\
(\mathsf{at}\ x\ C)^{\mathcal{I}}_{\mathcal{V},i} &= \begin{cases} C^{\mathcal{I}}_{\mathcal{V},\mathcal{V}(x)} & \text{if } x \neq now \\ C^{\mathcal{I}}_{\mathcal{V},i} & \text{if } x = now \\ C^{\mathcal{I}}_{\mathcal{V},\mathcal{M}(x)} & \text{if } x \text{ is a constant} \end{cases} \\
(\mathsf{sometime}\ (\overline{X})\ \overline{Tc}.C)^{\mathcal{I}}_{\mathcal{V},i} &= \{a \in \Delta^{\mathcal{I}} \mid \exists \mathcal{W}.\mathcal{W} \in \langle \overline{X}, \overline{Tc} \rangle^{\mathcal{E}}_{\sharp \mapsto i} \wedge a \in C^{\mathcal{I}}_{\mathcal{W},i}\} \\
(\mathsf{alltime}\ (\overline{X})\ \overline{Tc}.C)^{\mathcal{I}}_{\mathcal{V},i} &= \{a \in \Delta^{\mathcal{I}} \mid \forall \mathcal{W}.\mathcal{W} \in \langle \overline{X}, \overline{Tc} \rangle^{\mathcal{E}}_{\sharp \mapsto i} \Rightarrow a \in C^{\mathcal{I}}_{\mathcal{W},i}\} \\
(\mathsf{domain}\ C)^{\mathcal{I}}_{\mathcal{V},i} &= C^{\mathcal{I}}_{\mathcal{V},i} \times \Delta^{\mathcal{I}} \\
(\mathsf{range}\ C)^{\mathcal{I}}_{\mathcal{V},i} &= \Delta^{\mathcal{I}} \times C^{\mathcal{I}}_{\mathcal{V},i}
\end{aligned}
$$

Figure 6. Semantics for composed terms.

in $\overline{Tc}$. As an example, let $\overline{X} = \{x, y\}$ and $\overline{Tc} = \{(\mathsf{meets}\ x\ y)\}$, then $\forall \mathcal{V} \in \langle \overline{X}, \overline{Tc} \rangle^{\mathcal{E}}$, $\langle \mathcal{V}(x), \mathcal{V}(y) \rangle \in \mathcal{M}_4[\mathsf{meets}]$. Furthermore, $\langle \overline{X}, \overline{Tc} \rangle^{\mathcal{E}}_{x \mapsto i}$ denotes the set of interpretations of a time-net where x is mapped to i.

The interpretation of temporal conceptual expressions is a triple $\mathcal{I} \doteq \langle \mathcal{T}^{\star}_{<}, \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$, with the interval domain $\mathcal{T}^{\star}_{<}$, a generic non empty individual domain $\Delta^{\mathcal{I}}$ and an interpretation function $\cdot^{\mathcal{I}}$ which fixes the extension of atomic concepts and roles – denoted with the letters A and P respectively – in such a way that:

$$A^{\mathcal{I}} \subseteq \mathcal{T}^{\star}_{<} \times \Delta^{\mathcal{I}} \qquad P^{\mathcal{I}} \subseteq \mathcal{T}^{\star}_{<} \times \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$$

Note that, with respect to the temporal logic $\mathcal{HS}$ the interpretation of a temporal DL adds an object domain $\Delta^{\mathcal{I}}$ to the interval domain $\mathcal{T}^{\star}_{<}$. To interpret generic concept and role expressions the interpretation function has to satisfy the equations showed in Figure 6 – the equations for the analogous operators on roles are left to the intuition of the reader. Thus, each concept (role) is mapped to a function that assigns a set of individuals (of pairs of individuals) to each time interval. The notation $C^{\mathcal{I}}_{\mathcal{V},i}$ ($R^{\mathcal{I}}_{\mathcal{V},i}$) stands for the set of individuals (of pairs of individuals) of the domain which are of type C (R) at the time interval i, with the assignment to the free temporal variables in C (R) given by $\mathcal{V}$. Thus, the interpretation of a generic expression depends both on a given time interval i and on an assignment $\mathcal{V}$ for the free variables.

If we consider just closed concept expressions the interpretation does not depend on $\mathcal{V}$. An interpretation $\mathcal{I}$ is a *model* for a closed concept C if $C^{\mathcal{I}}_{i} \neq \emptyset$, for some interval i. A concept C is subsumed by the concept D ($C \sqsubseteq D$) if $C_i^{\mathcal{I}} \subseteq D_i^{\mathcal{I}}$ for all interpretation $\mathcal{I}$ and all time intervals i.

Schmiedel's work does not propose any algorithm for computing subsumption for this temporal variant of Description Logics, but only some preliminary hints are given. Schmiedel's temporal Description Logic can be formally related to the interval-based temporal logic $\mathcal{HS}$ proposed by Halpern and Shoham. As Bettini [9] shows (see Lemma 4.3 in the next Section), Schmiedel's logic when closed under complementation contains the $\mathcal{HS}$ logic as a proper fragment. Schmiedel's logic is argued to be undecidable, sacrificing the main benefit of Description Logics, i.e., the possibility to have decidable inference techniques (see the next Section for more details).

4.2. The Undecidable Realm

Bettini [9,10] suggests a variable-free extension with both existential and universal temporal quantification. He gives undecidability results for the proposed class of temporal languages – resorting to the undecidability results of Halpern and Shoham's temporal logic – and investigates approximated reasoning algorithms. Starting from the language $\mathcal{ALCN}$ two concept constructors are introduced: $\Diamond TE.\ C$ and $\Box TE.\ C$. The $\Diamond$ and $\Box$ operators are respectively the existential and universal temporal quantifiers, but, unlike Schmiedel's formalism, they do not allow for explicit interval variables. The temporal expression, TE, is a set of temporal constraints on two implicit intervals: the reference interval and the current one. This makes the language very close to the $\mathcal{HS}$ logic if we note that each TE can be simulated by an appropriate combination of temporal operators. Bettini presents a hierarchy of temporal expressions TE_i, with $i = 1 \ldots 5$, with higher temporal expressiveness. TE_1 allows to express single basic temporal relations, i.e., *meets*, *starts*, *finishes* and their converses. They are called basic since the other Allen's relations can be expressed by a combination of them. TE_2 includes the thirteen Allen's relations. TE_3 allows all those combination of temporal relations which give rise to the pointisable interval set. TE_4 allows for an arbitrary disjunction of temporal relations. TE_5 extends TE_4 with metric constraints by means of either duration intervals or specific constant intervals – reaching in this latter case an expressive power close to the one of the *time-net* of Schmiedel. Every language is indicated by prefixing the name of the non-temporal Description Logic composing it with the letter $\mathcal{T}$ and by adding a numerical subscript which denotes the kind of temporal expressions allowed. For example, $\mathcal{TFL}_5^-$ is the temporal extension of $\mathcal{FL}^-$ allowing TE_5 as temporal expressions in $\Diamond TE.\ C$ and $\Box TE.\ C$ operators. In this framework the concept of `Mortal` can be defined as:

$$\texttt{Mortal} \doteq \texttt{LivingBeing} \sqcap \Diamond(\textsf{met-by}).\ \neg\texttt{LivingBeing}$$

Depending on the underlying non-temporal Description Logic, there are some expressiveness equivalence between apparently different languages due to the interaction between temporal and non-temporal operators. As an example, when the languages are built upon $\mathcal{ALC}$ the following equivalences hold: $\mathcal{TALC}_1 \equiv \mathcal{TALC}_2 \equiv \mathcal{TALC}_3 \equiv \mathcal{TALC}_4$. These equivalences can be easily proved by recovering to the equivalences showed by Halpern and Shoham on the reducibility of the Allen's relations to a combination of the basic ones, and by noting that:

$$\diamond(rel_1, rel_2).\ C \equiv \big(\diamond(rel_1).\ C \ \sqcup\ \diamond(rel_2).\ C\big)$$
$$\square(rel_1, rel_2).\ C \equiv \big(\square(rel_1).\ C \ \sqcap\ \square(rel_2).\ C\big)$$

whenever you read (rel_1, rel_2) as the disjunction ($\mathsf{or}\ \ rel_1\ rel_2$).

As in the case of Schmiedel's formalism, the time is part of the semantic structure. A concept denotes a set of pairs of temporal intervals and individuals $\langle i, a\rangle$. Intuitively, a given individual belongs to the extension of a concept at certain time intervals. The temporal operators allow to relate the current interval to other intervals. The expression $\diamond TE.\ C$ denotes the set of pairs $\langle i, a\rangle$ such that the individual a belongs to the extension of C at the interval i' which satisfies $i'\,TE\,i$. For example, $\langle \mathsf{1990}, a_1\rangle$ belongs to the set $\diamond\mathsf{after}.\ \mathtt{Engineer}$ if there exists an interval that is after $\mathsf{1990}$ in which the individual a_1 is an $\mathtt{Engineer}$ – this will be the case if $\langle \mathsf{1991}, a_1\rangle$ would be an instance of the concept $\mathtt{Engineer}$. The $\square$ operator works in a similar way, but it universally qualifies the implicit temporal variable that satisfies the temporal constraint.

Let us now briefly introduce the model-theoretic semantics. Given an unbound and linear temporal structure, $\langle \mathcal{P}, <\rangle$, the domain of temporal intervals, $\mathcal{T}^\star_<$, is defined, as usual, as the set of pairs of points in $\mathcal{P}$. A fixed temporal model, $\mathcal{M}$, is assumed in the same spirit of Schmiedel, while $\cdot^{\mathcal{E}}$ is the temporal interpretation function that maps a temporal expression TE and a reference interval i into a set of intervals in $\mathcal{T}^\star_<$:

$$[rel]^{\mathcal{E}}_i = \{i' \mid \langle i', i\rangle \in \mathcal{M}_4[rel]\}$$
$$[rel_1, \ldots, rel_n]^{\mathcal{E}}_i = \textstyle\bigcup_{j=1}^n [rel_j]^{\mathcal{E}}_i$$

In addition there are other equations taking into account the interaction between qualitative and metric temporal constraints. An interpretation structure is a triple $\mathcal{I} \doteq \langle \mathcal{T}^\star_<, \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}}\rangle$, where $\Delta^{\mathcal{I}}$ is a set of individuals, $\mathcal{T}^\star_<$ is the time interval set and $\cdot^{\mathcal{I}}$ an interpretation function. It assigns a meaning to generic concept expressions by mapping each atomic concept into a set of pairs $\langle i, a\rangle$, and each role into a set of triples $\langle i, a, b\rangle$ – only the time-dependent constructs are reported here:

$$(\diamond TE.\ C)^{\mathcal{I}}_{\mathcal{V},i} = \{a \in \Delta^{\mathcal{I}} \mid \exists i'.\ i' \in (TE)^{\mathcal{E}}_i \wedge a \in C^{\mathcal{I}}_{i'}\}$$
$$(\square TE.\ C)^{\mathcal{I}}_{\mathcal{V},i} = \{a \in \Delta^{\mathcal{I}} \mid \forall i'.\ i' \in (TE)^{\mathcal{E}}_i \rightarrow a \in C^{\mathcal{I}}_{i'}\}$$

An interpretation $\mathcal{I}$ is a *model* for a concept C if $C^{\mathcal{I}}_i \neq \emptyset$, for some $i \in \mathcal{T}^\star_<$.

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 $C_i^{\mathcal{I}} \subseteq D_i^{\mathcal{I}}$ for every interpretation $\mathcal{I}$, and every $i \in \mathcal{T}_<^\star$.

Let us comment now on the temporal expressivity of this family of languages. The absence of explicit temporal variables weakens the temporal structure of a concept since arbitrary relationships between more than two intervals can not be represented anymore. For example, it is not possible to describe the situation where two concept expressions, say C and D, hold at two meeting intervals (say x,y) with the first interval starting and the second finishing the reference interval (i.e., the temporal pattern $(x\ \mathsf{meets}\ y)(x\ \mathsf{starts}\ \sharp)(y\ \mathsf{finishes}\ \sharp)$ cannot be represented). More precisely, it is not possible to represent temporal relations between more than two intervals if they are not derivable by the temporal propagation of the constraints imposed on pairs of variables. Although the use of explicit variables is against the general trust of Description Logics, the gained expressive power together with the observation that the variables are limited only to the temporal part of the language are the main motivation for using them.

This limited temporal expressive power is motivated by the need to study the computational properties of such Description Logics extended with temporal operators on intervals. Bettini provides a set of equivalence preserving reductions between $\mathcal{HS}$ formulæ and $\mathcal{TALC}_i$ concept expressions.

The following theorem easily follows from the above reductions, from the results provided for $\mathcal{HS}$, and by observing that in a propositionally complete language subsumption reduces to unsatisfiability.

Theorem 4.1. The problem of determining the satisfiability of terms in $\mathcal{TALC}_i(\mathcal{P}, \leq)$ is co-r.e.-hard, for all i with $1 \leq i \leq 5$. The subsumption problem in $\mathcal{TALC}_i(\mathcal{P}, \leq)$ is r.e.-hard.

These complexity results were obtained for $\mathcal{HS}$ but they were limited to temporal structures allowing for durationless intervals. Bettini extends these results to temporal structures which allow only for proper intervals. The following theorem considers the realm of integer numbers without durationless intervals.

Theorem 4.2. The satisfiability and subsumption problems for the languages $\mathcal{TALC}_i(\mathcal{Z}, <)$ with $i = 1, \ldots, 5$ are undecidable. In particular, they belong to the classes Σ_1^1-*hard* and Π_1^1-*hard*, respectively.

The author then shows that the Schmiedel's formalism, indicated as $\mathcal{TB}$, is strictly related to the class of languages $\mathcal{TALC}_i$ (and so to the modal logic $\mathcal{HS}$). In particular, he considers the language $\mathcal{TB}$ extended with the complement operator, called $\mathcal{TB}_{\text{neg}}$.

Lemma 4.3. (Correspondence between $\mathcal{TALC}_1$ and $\mathcal{TB}_{\text{neg}}$) Every concept expression C in $\mathcal{TALC}_1(\mathcal{Z},<)$ can be translated into a concept expression in $\mathcal{TB}_{\rm neg}$, in such a way that C is satisfiable if and only if its translation is a $\mathcal{TB}_{\rm neg}$ satisfiable concept.

¿From this equivalence it follows that for the language $\mathcal{TB}_{\rm neg}$ the very same complexity results of theorem 4.2 apply. Bettini's analysis leaves some important open problems, as declared by the author himself. The decidability of satisfiability and subsumption remains an open problem when considering temporal structures which allow only for *proper* intervals and which are different from the structure of integer numbers. An example of such a structure is the one that interprets intervals on the rational numbers, Q. A critical point that will deserve a deep investigation is the decidability of satisfiability and subsumption with respect to languages without negation. Since in this case the two problems are no more each other reducible they might belong to different complexity classes. In particular, it remains an open problem whether reasoning in the language $\mathcal{TB}$, as presented by Schmiedel, is decidable.

4.3. Towards Decidable Logics

Artale and Franconi [2,3] consider a class of interval-based temporal Description Logics by reducing the expressivity of [42]. While Schmiedel's work lacks of a computational machinery, and Halpern and Shoham's logic is undecidable, Artale and Franconi present different decidable logics, providing for them sound, complete and terminating reasoning algorithms.

The most expressive language, $\mathcal{TL}$-$\mathcal{ALCF}$, is presented here. $\mathcal{TL}$-$\mathcal{ALCF}$ is composed by the temporal Logic $\mathcal{TL}$ – able to express temporally quantified terms – and the non-temporal Description Logic $\mathcal{ALCF}$. *Concept expressions* (denoted by C,D) are built following the syntax rules of Figure 7. Temporal concepts (C,D) are distinct from non-temporal concepts (E,F). Names for atomic features and atomic parametric features are from the same alphabet of symbols; the $\star$ symbol is not intended as an operator, but only as differentiating the two syntactic types. For the basic interval relations the Allen notation [1] is adopted. Temporal variables are introduced by the temporal existential quantifier "$\Diamond$" – excluding the special temporal variable $\sharp$, usually called *now*, and intended as the reference interval.

The intended *meaning* of the terms of the language $\mathcal{TL}$-$\mathcal{ALCF}$ is similar to the meaning of concept and role terms found in Schmiedel's and Bettini's logics. Concept expressions are interpreted over pairs of *temporal intervals* and *individuals* $\langle i,a\rangle$ (roles are interpreted as triples $\langle i,a,b\rangle$), meaning that the individual a is in the extension of the concept (is related to b via a role) at the interval i. Thus, a concept interpretation can be seen as the set of individuals of that concept type at some interval. Within a concept expression, the special "$\sharp$" variable denotes the current interval of evaluation. The temporal existential quantifier

$\mathcal{TL}$	$C, D \rightarrow$	$E \mid$	(non-temporal concept)
		$C \sqcap D \mid$	(conjunction)
		$C@X \mid$	(qualifier)
		$C[Y]@X \mid$	(substitutive qualifier)
		$\diamond(\overline{X})\ \overline{\mathit{Tc}}.C$	(existential quantifier)
	$\mathit{Tc} \rightarrow$	$(X\ (R)\ Y) \mid$	(temporal constraint)
		$(X\ (R)\ \sharp) \mid$	
		$(\sharp\ (R)\ Y)$	
	$\overline{\mathit{Tc}} \rightarrow$	$\mathit{Tc} \mid \mathit{Tc}\ \overline{\mathit{Tc}}$	
	$R, S \rightarrow$	$R\ ,\ S \mid$	(disjunction)
		$\mathsf{starts} \mid \mathsf{finishes} \mid \mathsf{met\text{-}by} \mid \ldots$	(Allen's relations)
	$X, Y \rightarrow$	$\mathrm{x} \mid \mathrm{y} \mid \mathrm{z} \mid \ldots$	(temporal variables)
	$\overline{X} \rightarrow$	$X \mid X\ \overline{X}$	
$\mathcal{ALCF}$	$E, F \rightarrow$	$A \mid$	(atomic concept)
		$\top \mid$	(top)
		$\bot \mid$	(bottom)
		$\neg E \mid$	(complement)
		$E \sqcap F \mid$	(conjunction)
		$E \sqcup F \mid$	(disjunction)
		$\forall P.E \mid$	(universal quantifier)
		$\exists P.E \mid$	(existential quantifier)
		$p : E$	(selection)
		$p \downarrow q \mid$	(agreement)
		$p \uparrow q \mid$	(disagreement)
		$p \uparrow$	(undefinedness)
	$p, q \rightarrow$	$f \mid$	(atomic feature)
		$\star g \mid$	(atomic parametric feature)
		$p \circ q$	(path)

Figure 7. Syntax rules for the interval Description Logic $\mathcal{TL}$-$\mathcal{ALCF}$

introduces interval variables, related to each other and possibly to the $\sharp$ variable in a way defined by the set of *temporal constraints*. In order to evaluate a concept at an interval X, different from the current one, we need to temporally qualify it at X – written $C@X$; in this way, every occurrence of $\sharp$ embedded within the concept expression C is interpreted as the X variable[4]. The informal meaning of a concept with a temporal existential quantification can be understood with the following examples in the action domain.

[4] Since any concept is implicitly temporally qualified at the special $\sharp$ variable, it is not necessary to explicitly qualify concepts at $\sharp$.

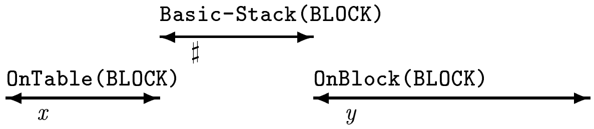

Figure 8. Temporal dependencies in the definition of the `Basic-Stack` action.

$$\texttt{Basic-Stack} \doteq \diamond(x\ y)\ (x \ \mathsf{meets}\ \sharp)(\sharp\ \mathsf{meets}\ y).\\ \big((\star\texttt{BLOCK} : \texttt{OnTable})@x \ \sqcap\ (\star\texttt{BLOCK} : \texttt{OnBlock})@y\big)$$

Figure 8 shows the temporal dependencies of the intervals in which the concept `Basic-Stack` holds. `Basic-Stack` denotes, according to the definition (a terminological axiom), any action occurring at some interval involving a $\star$`BLOCK` that was once `OnTable` and then `OnBlock`. 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 constraints ($x\ \mathsf{m}\ \sharp$) and ($\sharp\ \mathsf{m}\ y$) state that the interval denoted by x should meet the interval denoted by $\sharp$ – the occurrence interval of the action type `Basic-Stack` – and that $\sharp$ should meet y. The parametric feature $\star$`BLOCK` plays the role of *formal* parameter of the action, mapping any individual action of type `Basic-Stack` to the block to be stacked, independently from time. Please note that, whereas the existence and identity of the $\star$`BLOCK` of the action is time invariant, it can be qualified differently in different intervals of time, e.g., the $\star$`BLOCK` is necessarily `OnTable` only during the interval denoted by x.
In this framework, the concept defining a `Mortal` is:

$$\texttt{Mortal} \doteq \diamond(x)\ (\sharp\ \mathsf{meets}\ x).\ \texttt{LivingBeing} \sqcap\ \neg\texttt{LivingBeing}@x$$

$\mathcal{TL}$-$\mathcal{ALCF}$ is provided with a Tarski-style extensional semantics. A linear, unbounded, and dense temporal structure $\mathcal{T} = (\mathcal{P}, <)$ is assumed, where $\mathcal{P}$ is a set of time points and $<$ is a strict linear order on $\mathcal{P}$. The *interval set* of a structure $\mathcal{T}$ is defined as the set $\mathcal{T}^{\star}_{<}$ of all closed proper intervals $[t_1, t_2] \doteq \{x \in \mathcal{P} \mid t_1 \leq x \leq t_2, t_1 \neq t_2\}$ in $\mathcal{T}$. A *primitive 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 $\mathcal{I}$) which gives a meaning to atomic concepts, roles, features and parametric features:

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

Atomic parametric features are interpreted as partial functions; they differ from atomic features for being independent from time[5]. In order to give a meaning to temporal expressions present in generic concept expressions, Figure 9 defines

[5] Parametric Features can be seen as a form of *global roles* (see Section 5.2)

$$
\begin{aligned}
(\mathsf{starts})^{\mathcal{E}} &= \{\langle [u,v],[u_1,v_1]\rangle \in \mathcal{T}^{\star}_{<} \times \mathcal{T}^{\star}_{<} \mid u = u_1 \wedge v < v_1\} \\
(\mathsf{finishes})^{\mathcal{E}} &= \{\langle [u,v],[u_1,v_1]\rangle \in \mathcal{T}^{\star}_{<} \times \mathcal{T}^{\star}_{<} \mid v = v_1 \wedge u_1 < u\} \\
(\mathsf{met\text{-}by})^{\mathcal{E}} &= \{\langle [u,v],[u_1,v_1]\rangle \in \mathcal{T}^{\star}_{<} \times \mathcal{T}^{\star}_{<} \mid u = v_1\} \\
& \ldots \textit{(meaning of the other Allen temporal relations)} \\
(R\ \textbf{,}\ S)^{\mathcal{E}} &= R^{\mathcal{E}} \cup S^{\mathcal{E}} \\
\langle \overline{X}, \overline{Tc}\rangle^{\mathcal{E}} &= \{\mathcal{V} : \overline{X} \mapsto \mathcal{T}^{\star}_{<} \mid\ \forall (X\ (R)\ Y) \in \overline{Tc}.\ \langle \mathcal{V}(X), \mathcal{V}(Y)\rangle \in (R)^{\mathcal{E}}\}
\end{aligned}
$$

Figure 9. The temporal interpretation function.

the *temporal interpretation function*. The *temporal interpretation function* $\cdot^{\mathcal{E}}$ depends only on the temporal structure $\mathcal{T}$. The labelled directed graph $\langle \overline{X}, \overline{Tc}\rangle$ – where $\overline{X}$ is the set of variables representing the nodes, and $\overline{Tc}$ is the set of temporal constraints representing the arcs – is called *temporal constraint network.* The interpretation of a temporal constraint network is a set of variable assignments which satisfy the temporal constraints. A *variable assignment* is a function $\mathcal{V} : \overline{X} \mapsto \mathcal{T}^{\star}_{<}$ associating an interval value to a temporal variable. A temporal constraint network is *consistent* if it admits a non empty interpretation. The notation, $\langle \overline{X}, \overline{Tc}\rangle^{\mathcal{E}}_{\{x_1 \mapsto i_1, x_2 \mapsto i_2, \ldots\}}$, used to interpret concept expressions, denotes the subset of $\langle \overline{X}, \overline{Tc}\rangle^{\mathcal{E}}$ where the variable x_j is mapped to the interval value i_j.

At this point we are able to interpret generic concept expressions. An *interpretation function* $\cdot^{\mathcal{I}}_{\mathcal{V},i,\mathcal{H}}$, based on a variable assignment $\mathcal{V}$, an interval t and a set of constraints $\mathcal{H} = \{x_1 \mapsto i_1, \ldots\}$ over the assignments of inner variables, extends the primitive interpretation function in such a way that the equations of the Figure 10 are satisfied – we do not report the constructors that can be obtaind by complementation. Intuitively, the interpretation of a concept $C^{\mathcal{I}}_{\mathcal{V},i,\mathcal{H}}$ is the set of entities of the domain which are of type C at the time interval i, with the assignment for the free temporal variables in C given by $\mathcal{V}$ – $(C@X)^{\mathcal{I}}_{\mathcal{V},i,\mathcal{H}}$ – and with the constraints for the assignment of variables in the scope of the outermost temporal quantifiers given by $\mathcal{H}$. Notice that $\mathcal{H}$ interprets the variable renaming due to the temporal substitutive qualifier – $(C[Y]@X)^{\mathcal{I}}_{\mathcal{V},i,\mathcal{H}}$ – and it takes effect during the choice of a variable assignment, as the equation $(\Diamond(\overline{X})\ \overline{Tc}.\ C)^{\mathcal{I}}_{\mathcal{V},i,\mathcal{H}}$ shows.

In absence of free variables in the concept expression – with the exception of $\sharp$ – it is introduced as a notational simplification the *natural* interpretation function $C^{\mathcal{I}}_{i}$, being equivalent to the interpretation function $C^{\mathcal{I}}_{\mathcal{V},i,\mathcal{H}}$ with any $\mathcal{V}$ such that $\mathcal{V}(\sharp) = i$, and $\mathcal{H} = \emptyset$. The set of interpretations $\{C^{\mathcal{I}}_{\mathcal{V},i,\mathcal{H}}\}$ obtained by varying $\mathcal{I}, \mathcal{V}, i$ 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 interpretations with a fixed $\mathcal{H}$. In fact, since $\mathcal{H}$ represents a constraint in the assignment of variables, the unconstrained set is the largest one. Note that, the feature's interpretation uses just the natural

$$
\begin{aligned}
A^{\mathcal{I}}_{\mathcal{V},i,\mathcal{H}} &= \{a \in \Delta^{\mathcal{I}} \mid \langle i,a\rangle \in A^{\mathcal{I}}\} \quad = \quad A^{\mathcal{I}}_i\\
\top^{\mathcal{I}}_{\mathcal{V},i,\mathcal{H}} &= \Delta^{\mathcal{I}} \quad = \quad \top^{\mathcal{I}}\\
(\neg C)^{\mathcal{I}}_{\mathcal{V},i,\mathcal{H}} &= \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}_{\mathcal{V},i,\mathcal{H}}\\
(C \sqcap D)^{\mathcal{I}}_{\mathcal{V},i,\mathcal{H}} &= C^{\mathcal{I}}_{\mathcal{V},i,\mathcal{H}} \cap D^{\mathcal{I}}_{\mathcal{V},i,\mathcal{H}}\\
(\forall P.C)^{\mathcal{I}}_{\mathcal{V},i,\mathcal{H}} &= \{a \in \Delta^{\mathcal{I}} \mid \forall b.(a,b) \in P^{\mathcal{I}}_i \Rightarrow b \in C^{\mathcal{I}}_{\mathcal{V},i,\mathcal{H}}\}\\
(p \downarrow q)^{\mathcal{I}}_{\mathcal{V},i,\mathcal{H}} &= \{a \in dom\, p^{\mathcal{I}}_i \cap dom\, q^{\mathcal{I}}_i \mid p^{\mathcal{I}}_i(a) = q^{\mathcal{I}}_i(a)\} \quad = \quad (p \downarrow q)^{\mathcal{I}}_i\\
(p : C)^{\mathcal{I}}_{\mathcal{V},i,\mathcal{H}} &= \{a \in dom\, p^{\mathcal{I}}_i \mid p^{\mathcal{I}}_i(a) \in C^{\mathcal{I}}_{\mathcal{V},i,\mathcal{H}}\}\\
(C@X)^{\mathcal{I}}_{\mathcal{V},i,\mathcal{H}} &= C^{\mathcal{I}}_{\mathcal{V},\mathcal{V}(X),\mathcal{H}}\\
(C[Y]@X)^{\mathcal{I}}_{\mathcal{V},i,\mathcal{H}} &= C^{\mathcal{I}}_{\mathcal{V},i,\mathcal{H}\cup\{Y\mapsto\mathcal{V}(X)\}}\\
(\Diamond(\overline{X})\ \overline{Tc}.\ C)^{\mathcal{I}}_{\mathcal{V},i,\mathcal{H}} &= \{a \in \Delta^{\mathcal{I}} \mid \exists \mathcal{W}.\ \mathcal{W} \in \langle \overline{X}, \overline{Tc}\rangle^{\mathcal{E}}_{\mathcal{H}\cup\{\sharp\mapsto i\}} \wedge a \in C^{\mathcal{I}}_{\mathcal{W},i,\emptyset}\}\\
P^{\mathcal{I}}_i &= \hat{P}_i \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid\ \forall a,b.\ \langle a,b\rangle \in \hat{P}_i \leftrightarrow \langle i,a,b\rangle \in P^{\mathcal{I}}\\
f^{\mathcal{I}}_i &= \hat{f}_i : \Delta^{\mathcal{I}} \overset{partial}{\longmapsto} \Delta^{\mathcal{I}} \mid\ \forall a.\ (a \in dom\, \hat{f}_i \leftrightarrow \langle i,a\rangle \in dom\, f^{\mathcal{I}}) \wedge\\
&\qquad\qquad\qquad\qquad\qquad \hat{f}_i(a) = f^{\mathcal{I}}(i,a)\\
(p \circ q)^{\mathcal{I}}_i &= p^{\mathcal{I}}_i \circ q^{\mathcal{I}}_i\\
\star g^{\mathcal{I}}_i &= \star g^{\mathcal{I}}
\end{aligned}
$$

Figure 10. The interpretation function.

one since it is not admitted to temporally qualify them.

An interpretation $\mathcal{I}$ is a *model* for a concept C if $C^{\mathcal{I}}_i \neq \emptyset$ for some $i \in \mathcal{T}^{\star}_{<}$. 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 $C^{\mathcal{I}}_i \subseteq D^{\mathcal{I}}_i$ for every interpretation $\mathcal{I}$ and every interval $i \in \mathcal{T}^{\star}_{<}$.

Similar to the case for the logic $\mathcal{HS}$, only the relations starts, finishes, met-by are really necessary (note that, the temporal structure does not allow durationless intervals), because it is possible to express any temporal relationship between two distinct intervals using only these three relations and their transposes started-by, finished-by, meets. In fact, the following equivalences hold:

$$
\begin{aligned}
\Diamond x\ (x \ \mathsf{after}\ \sharp).\ C@x &\equiv \Diamond xy\ (y\ \mathsf{met\text{-}by}\ \sharp)(x\ \mathsf{met\text{-}by}\ y).\ C@x\\
\Diamond x\ (x\ \mathsf{during}\ \sharp).\ C@x &\equiv \Diamond xy\ (y\ \mathsf{starts}\ \sharp)(x\ \mathsf{finishes}\ y).\ C@x\\
\Diamond x\ (x\ \mathsf{overlaps}\ \sharp).\ C@x &\equiv \Diamond xy\ (y\ \mathsf{starts}\ \sharp)(x\ \mathsf{finished\text{-}by}\ y).\ C@x
\end{aligned}
$$

We report here how the authors propose to represent complex actions that involve parameters. A stacking action involves two blocks, which should be both clear at the beginning; the central part of the action consists of holding one block; at the end, the blocks are one on top of the other, and the bottom one is no longer clear (Figure 11):

$$
\begin{aligned}
\texttt{Stack} \doteq\ & \diamond(x\ y\ z\ v\ w)\ \ (x\ \textsf{finished-by}\ \sharp)(y\ \textsf{meets}\ \sharp)(z\ \textsf{met-by}\ \sharp)\\
& (v\ \textsf{overlaps}\ \sharp)(w\ \textsf{finishes}\ \sharp)(w\ \textsf{met-by}\ v).\\
& ((\star\texttt{OBJECT2} : \texttt{Clear-Block})@x \sqcap (\star\texttt{OBJECT1}\circ\texttt{ON} \downarrow \star\texttt{OBJECT2})@y \sqcap\\
& \ (\star\texttt{OBJECT1} : \texttt{Clear-Block})@v \sqcap (\star\texttt{OBJECT1} : \texttt{Holding-Block})@w \sqcap\\
& \ (\star\texttt{OBJECT1} : \texttt{Clear-Block})@z)
\end{aligned}
$$

The definition makes use of temporal qualified concept expressions: 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; while $(\star\texttt{OBJECT1}\circ\texttt{ON} \downarrow \star\texttt{OBJECT2})@y$ indicates that at the interval y the object on which $\star$`OBJECT1` is placed is $\star$`OBJECT2`. The above defined 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.

Artale and Franconi contribute to explore the decidable realm of interval-based temporal Description Logics by presenting sound, complete and terminating procedures for subsumption reasoning. In order to obtain decidable languages a serious restriction has been posed on the temporal expressivity: the universal quantification on temporal variables has been eliminated. The main results are proved starting with the simplest language, $\mathcal{TL}$-$\mathcal{F}$, where $\mathcal{F}$ is a feature language (i.e., only functional roles are permitted) with neither negation nor disjunction. Then, the authors show how to reason with more expressive languages such as $\mathcal{TLU}$-$\mathcal{FU}$, which adds disjunction both at the temporal and non-temporal sides of the language, and $\mathcal{TL}$-$\mathcal{ALCF}$. The subsumption procedures are based on a *normalization procedure*, i.e., an interpretation preserving transformation which operates a separation between the temporal and the non-temporal part of the formalism. A concept in normal form can be seen as a *conceptual temporal constraint network* , i.e., a labeled directed graph $\langle \overline{X}, \overline{\mathcal{T}c}, \overline{Q@X}\rangle$ (in $\mathcal{TL}$-$\mathcal{ALCF}$ syntax: $\diamond(\overline{X})\ \overline{\mathcal{T}c}.\ (Q^0 \sqcap Q^1@X^1 \sqcap \ldots \sqcap Q^n@X^n))$ where arcs are labeled with a set of arbitrary temporal relationships – representing their disjunction, and temporal nodes are labeled with non-temporal concepts (i.e., each Q^j is an $\mathcal{ALCF}$ concept expression). The subsumption procedure checks whether there is a mapping function between the conceptual temporal constraint networks (i.e., a form of subgraph isomorphism, called by the authors *s-mapping*) such that a subsumption relation holds both among the non-temporal concepts labeling the corresponding nodes in the mapping function, and among the temporal relations of the corresponding arcs. Then the calculus can adopt standard procedures developed both in the Description Logics community and in the temporal constraints community. The following theorem summarizes the main results proved by Artale and Franconi:

Theorem 4.4. Let C_1 and C_2 be either $\mathcal{TL}$-$\mathcal{F}$ or $\mathcal{TL}$-$\mathcal{ALCF}$ concepts in normal form, then C_1 subsumes C_2 ($C_2 \sqsubseteq C_1$) if and only if there exists an *s-mapping*

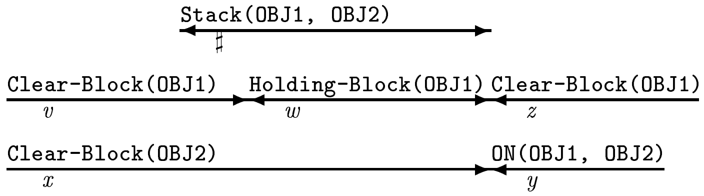

Figure 11. Temporal dependencies in the definition of the `Stack` action.

from C_1 to C_2.

Let $C = C_1 \sqcup \cdots \sqcup C_m$ and $D = D_1 \sqcup \cdots \sqcup D_n$ be $\mathcal{TLU}$-$\mathcal{FU}$ concepts in normal form; then D subsumes C if and only if $\forall i \exists j.\ C_i \sqsubseteq D_j$.

Concept subsumption between $\mathcal{TL}$-$\mathcal{F}$ or $\mathcal{TLU}$-$\mathcal{FU}$ concept expressions in normal form is an NP-complete problem.

5. Point-based Temporal Description Logics

In this Section we present various proposals that extend Description Logics with an explicit point-based notion of time following the *external* approach. The resulting logics are the combination of a static DL with a tense logic [43,13,24].

5.1. Combining Description and Tense Logics

Schild [40] combines the Description Logic $\mathcal{ALC}$ with *point*-based modal temporal connectives. The new language is called $\mathcal{ALCT}$, and the temporal connectives are those of tense logic [13]: Existential Future ($\Diamond$), Universal Future ($\Box$), Next Instant ($\bigcirc$), Until ($\mathcal{U}$), Reflexive Until ($\mathbf{U}$).

As in the case of Schmiedel, Bettini, Artale and Franconi's formalisms, the time is part of the semantic structure. A concept denotes a set of pairs of temporal points and individuals $\langle t, a\rangle$, while a role is interpreted as a set of triples $\langle t, a, b\rangle$. The operator *existential future* denotes those individuals that belong to C at *some* time coincident or successive to the actual time. As an example, the concept of `Mortal` can be defined in $\mathcal{ALCT}$ as:

$$\texttt{Mortal} \doteq \texttt{LivingBeing} \sqcap \Diamond\neg\texttt{LivingBeing}$$

which denotes the set of pairs $\langle t, a\rangle$ where a is a kind of `LivingBeing` at the time t, and there exists an instant $t' \geq t$ where a is no more a `LivingBeing`. We observe that, in this example, the time point t' should not be coincident with t, since an individual cannot belong to disjoint concepts, as is the case for `LivingBeing` and $\neg$`LivingBeing`. Thus, a better definition for `Mortal` makes use of the *next instant* operator, $\bigcirc$. Intuitively, given a time t, the concept $\bigcirc C$

denotes the set of individuals that belong to C at the *immediate* successor of t[6]. The new definition of `Mortal` is the following:

$\texttt{Mortal} \doteq \texttt{LivingBeing} \sqcap \bigcirc\Diamond\neg\texttt{LivingBeing}$

The operator *universal future*, $\Box$, is the dual of $\Diamond$. Given a time point t, the concept $\Box C$ denotes the set of individuals which are of kind C at every time $t' \geq t$. With this operator, the definition of a mortal can be refined by saying that from a certain future time, $t' > t$, he will never be alive again:

$\texttt{Mortal} \doteq \texttt{LivingBeing} \sqcap \bigcirc\Diamond\Box\neg\texttt{LivingBeing}$

This definition is still incomplete since does not tell anything about the time between t – when the mortal is alive – and t' – when a mortal dies. At each time t'' with $t < t'' < t'$, a mortal can be dead or alive. For this purpose the binary operator *until*, $\mathcal{U}$, can be used. At time t, the concept $C\mathcal{U}D$ denotes all those individuals which are of kind D at some time $t' > t$ and which are of kind C for all times t'' with $t < t'' < t'$. Thus, a mortal can redefined as a living being who is alive *until* he dies:

$\texttt{Mortal} \doteq \texttt{LivingBeing} \sqcap (\texttt{LivingBeing}\, \mathcal{U}\, \Box\neg\texttt{LivingBeing})$

A slight variant of the until operator is the constructor *reflexive until*, $\mathbf{U}$. At time t the concept $C\mathbf{U}D$ denotes all those individuals which are of kind D at some time $t' \geq t$, and which are of kind C for every time t'' with $t \leq t'' < t'$. In the last definition of mortal, $\mathcal{U}$ can be substituted by $\mathbf{U}$ without changing its meaning.

More formally, complex temporal concepts can be expressed using the following syntax.

Definition 5.1. The *tense-logical extension* of a concept language $\mathcal{L}$, called $\mathcal{LT}$, is the least set containing all concepts of $\mathcal{L}$, and such that $\bigcirc C, \Diamond C, C\mathcal{U}D, C\mathbf{U}D$ are concepts of $\mathcal{LT}$ if C and D are concepts of $\mathcal{LT}$.

An interpretation for $\mathcal{LT}$ is a triple $\mathcal{I} \doteq \langle \mathcal{T}, \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$, with the time domain $\mathcal{T} = (\mathcal{P}, <)$, a generic individual domain $\Delta^{\mathcal{I}}$ and an interpretation function $\cdot^{\mathcal{I}}$ which fixes the extension of atomic concepts and roles – denoted with the letters A and P respectively – in such a way that:

$$A^{\mathcal{I}} \subseteq \mathcal{T} \times \Delta^{\mathcal{I}} \qquad P^{\mathcal{I}} \subseteq \mathcal{T} \times \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$$

To interpret generic concept expressions the interpretation function has to satisfy the equations showed in Figure 6 (only the semantics for the temporal constructs in $\mathcal{LT}$ is illustrated)[7]. An $\mathcal{LT}$ concept C is satisfiable if there exists an interpretation $\mathcal{I}$ such that $C_t^{\mathcal{I}} \neq \emptyset$, for some t. A concept C is subsumed by D, $C \sqsubseteq D$,

[6] For the *immediate successor* to be definable, a discrete temporal structure is needed.

[7] Analogously to the case of interval-based extensions, the notation $C_t^{\mathcal{I}}$ stands for the set of individuals belonging to C at the time point t.

$$
\begin{aligned}
(\bigcirc C)_t^{\mathcal{I}} &= \{a \in \Delta^{\mathcal{I}} \mid \exists t'.t < t' \ \wedge \ a \in C_{t'}^{\mathcal{I}} \ \wedge \ \neg\exists t''.t < t'' < t'\} \\
(\diamond C)_t^{\mathcal{I}} &= \{a \in \Delta^{\mathcal{I}} \mid \exists t'.t \leq t' \ \wedge \ a \in C_{t'}^{\mathcal{I}}\} \\
(\Box C)_t^{\mathcal{I}} &= \{a \in \Delta^{\mathcal{I}} \mid \forall t'.t \leq t' \ \wedge \ a \in C_{t'}^{\mathcal{I}}\} \\
(C\mathcal{U}D)_t^{\mathcal{I}} &= \{a \in \Delta^{\mathcal{I}} \mid \exists t'.t < t' \ \wedge \ a \in D_{t'}^{\mathcal{I}} \ \wedge \ \forall t''.t < t'' < t' \ \rightarrow \ a \in C_{t''}^{\mathcal{I}}\} \\
(C\mathbf{U}D)_t^{\mathcal{I}} &= \{a \in \Delta^{\mathcal{I}} \mid \exists t'.t \leq t' \ \wedge \ a \in D_{t'}^{\mathcal{I}} \ \wedge \ \forall t''.t \leq t'' < t' \ \rightarrow \ a \in C_{t''}^{\mathcal{I}}\}
\end{aligned}
$$

Figure 12. The $\mathcal{LT}$ semantics.

if $C_t^{\mathcal{I}} \subseteq D_t^{\mathcal{I}}$, for all interpretations $\mathcal{I}$ and all time points t.

Schild analyzes the computational property of the concept satisfiability problem[8] w.r.t. an empty KB. He proves that concept satisfiability checking in $\mathcal{ALCT}$ is of the same complexity class as concept satisfiability in $\mathcal{ALC}$, when interpreted over linear, unbounded and discrete time structures like the natural numbers. Thus, reasoning in $\mathcal{ALCT}$ $(\mathcal{N})$ – which denotes the language $\mathcal{ALCT}$ interpreted over the natural numbers – w.r.t. an empty KB, is a PSPACE-complete problem. These important results show that adding a point-based time dimension to $\mathcal{ALC}$ does not alter its computational behavior. However, since for branching, discrete and unbounded time reasoning in classical tense logic is an EXPTIME-hard problem [22] then the same lower complexity bound carries over $\mathcal{ALCT}$. When $\mathcal{ALC}$ is extended with an interval-based time dimension (let us call it $\mathcal{ALC}$-$\mathcal{INT}$) the undecidability results showed by Halpern and Shoham for the logic $\mathcal{HS}$, and the one presented by Bettini, apply also to $\mathcal{ALC}$-$\mathcal{INT}$. Interesting open problems remain. One concerns the complexity of $\mathcal{ALCT}$ $(\mathcal{N})$ when extended with *past tense* operators. It is also unknown whether $\mathcal{ALCT}$ is still decidable when interpreted on the structure of *real* numbers. Another analyzed language is the temporal extension of deterministic PDL (D-PDL). It is proved that satisfiability in temporal D-PDL interpreted over branching (linear), discrete and unbounded temporal structures is log-space reducible to the non-temporal version of D-PDL. Thus, also in this case adding tense operators does not deteriorate the computational complexity of D-PDL – which is EXPTIME-complete [39].

5.2. Description Logics with Modal Operators

Starting from the correspondence between description and modal logics many recent works investigate various way of combining modal operators within a description language. *Multi-Dimensional* Description Logics [6,7,48,50,49] have been studied where the usual object dimension is combined with other dimensions like time, knowledge, belief, actions, etc.

As observed by Wolter and Zakharyaschev [48], different design choices have

[8] We remind here that in $\mathcal{ALCT}$ concept subsumption reduces to concept satisfiability (see Section 2).

to be taken concerning the integration of modal operators. First of all, modal operators can be applied in different places. They can be used not only to form new concept terms but also in front of terminological and assertional axioms – as proposed by Baader and Laux [6] – and to build new roles – as proposed by Baader and Ohlbach [7]. The following examples show the use of a modalized terminological axiom, and of a modalized role:

$$[\texttt{BEL-JOHN}](\texttt{Happy-father} \doteq \exists\texttt{MARRIED-TO.}(\texttt{Woman} \sqcap [\texttt{BEL-JOHN}]\texttt{Pretty}) \sqcap \langle\texttt{future}\rangle\forall\texttt{CHILD.Graduate})$$

$$John : \exists[\texttt{always}]\texttt{LOVES.Woman}$$

expressing that, it is in the *John*'s belief that a `Happy-father` is someone married to a woman believed to be pretty by *John*, and whose children will be graduate sometime in the future; and that *John* will always love the same woman. Note that, the modalized role [`always`]`LOVES` restricts the role `LOVES` to the pairs $\langle x, y\rangle$ which are in the `LOVES` relation in every future time.

Another parameter concerns the different *dimensions* we want to model. For example, if we are interested in modeling both the time and the belief dimensions, then in the temporal dimension we could have *future* and *past* modalities as well as one for the *next* instant, while in the dimension of belief we could be interested in both *belief-John* and *belief-Mary*. In the following, we will introduce a general framework for combining a DL with additional modal dimensions, without necessarily reduce ourselves to just the temporal dimension. At the end, results about the specific temporal case will be reported.

With respect to the object domain dimension three different assumptions can be made. Either different modal worlds give rise to arbitrary object domains (*varying domain assumption*), or the object domain relative to a world w is contained in all the object domains relative to worlds reachable from w (*expanding domain assumption*), or the object domain is the same for all worlds (*constant domain assumption*). Reasoning with respect to different domain assumptions can give different results. For example, the following set of formulæ:

$$\Diamond(C \neq \bot), \quad (\Box\neg C) \doteq \top$$

is satisfiable adopting either the varying or the expanding domain assumption – since the $\top$ in the second formula denotes only the actual world – but not assuming constant domains – since the $\top$ in the second formula will always denote the whole domain in all possible worlds.

The final decision regards the distinction between *local* and *global* concepts, roles and constants. While global elements have the same extension in all possible worlds, local ones may have arbitrary extensions. In some sense, for a global element the interpretation is decoupled from the modal dimension. The *rigid designator* hypothesis assumes that each constant denotes the same domain object in all possible worlds – i.e., constants are globals.

Wolter and Zakharyaschev [48,49] introduce the language called $\mathcal{ALC}_{\mathcal{M}}$.

Definition 5.2. $\mathcal{ALC}_{\mathcal{M}}$ concepts are defined inductively as follows: All concept names as well as $\bot$ and $\top$ are concepts. If C and D are concepts, R is a role name and $\Diamond_i, \Box_i$, with $i = 1, \ldots, n$, are n modalities then $C \sqcap D, \neg C, \exists R.C, \Diamond_i C$ are concepts, while $\Diamond_i R, \Box_i R$ are roles.
Let C and D be concepts, R a role, a, b object names. Then expressions of the form $C \doteq D, aRb, a : C$ are (*atomic*) formulæ. If ϕ and ψ are formulæ then so are $\Diamond_i \phi, \neg\phi, \phi \sqcap \psi$. The language without modalized roles will be called $\mathcal{ALC}^{-}_{\mathcal{M}}$.

Each modal operator is interpreted as an accessibility relation on a set of possible worlds. Thus we need a set of possible worlds D_i for each dimension i (in the case of time an additional temporal structure on the possible worlds is needed). To each world a classical $\mathcal{ALC}$ interpretation structure is defined, where concepts and roles are given a meaning.

Definition 5.3. An interpretation for $\mathcal{ALC}_{\mathcal{M}}$ consists of a Kripke structure $\mathcal{K} = \langle \mathcal{W}, \Gamma, \mathcal{I} \rangle$ such that: $\mathcal{W}$, the set of possible worlds, is the Cartesian product of non-empty domains $D_1, \ldots, D_n$, one for each modal dimension; Γ contains for each modality of dimension i an accessibility relation γ_i, which is a function $\gamma_i : \mathcal{W} \mapsto 2^{D_i}$ (whenever $d'_i \in \gamma_i(d_1, \ldots, d_i, \ldots, d_n)$ we will write $\langle (d_1, \ldots, d_i, \ldots, d_n), (d_1, \ldots, d'_i, \ldots, d_n) \rangle \in \gamma_i$); $\mathcal{I}$ is a function associating to each world $w \in \mathcal{W}$ an interpretation structure $\langle \Delta^{\mathcal{I}(w)}, \cdot^{\mathcal{I}(w)} \rangle$ which consists of a non-empty set of objects $\Delta^{\mathcal{I}(w)}$, and of an interpretation function $\cdot^{\mathcal{I}(w)}$ that associates: to each object name a an element $a^{\mathcal{I}(w)} \in \Delta^{\mathcal{I}(w)}$ such that $a^{\mathcal{I}(w)} = a^{\mathcal{I}(v)}$ for any $w, v \in \mathcal{W}$ (i.e., the interpretation of individuals does not depend on the actual world, this is the *rigid designator hypothesis*); to each concept name A and world $w \in \mathcal{W}$ a set $A^{\mathcal{I}(w)} \subseteq \Delta^{\mathcal{I}(w)}$; to the $\bot, \top$ concepts the sets $\top^{\mathcal{I}(w)} = \Delta^{\mathcal{I}(w)}$ and $\bot^{\mathcal{I}(w)} = \emptyset$; to each role name R and world $w \in \mathcal{W}$ a binary relation $R^{\mathcal{I}(w)} \subseteq \Delta^{\mathcal{I}(w)} \times \Delta^{\mathcal{I}(w)}$. Furthermore, the interpretation is extended to generic concepts and roles as follows:

$$
\begin{aligned}
(C \sqcap D)^{\mathcal{I}(w)} &= C^{\mathcal{I}(w)} \cap D^{\mathcal{I}(w)} \\
(\neg C)^{\mathcal{I}(w)} &= \Delta^{\mathcal{I}(w)} \setminus C^{\mathcal{I}(w)} \\
(\exists R.C)^{\mathcal{I}(w)} &= \{x \in \Delta^{\mathcal{I}(w)} \mid \exists y. \langle x, y \rangle \in R^{\mathcal{I}(w)} \wedge y \in C^{\mathcal{I}(w)}\} \\
(\Diamond_i C)^{\mathcal{I}(w)} &= \{x \in \Delta^{\mathcal{I}(w)} \mid \exists v. \langle w, v \rangle \in \gamma_i \wedge x \in C^{\mathcal{I}(v)}\} \\
(\Diamond_i R)^{\mathcal{I}(w)} &= \{\langle x, y \rangle \in \Delta^{\mathcal{I}(w)} \times \Delta^{\mathcal{I}(w)} \mid \exists v. \langle w, v \rangle \in \gamma_i \wedge \langle x, y \rangle \in R^{\mathcal{I}(v)}\} \\
(\Box_i R)^{\mathcal{I}(w)} &= \{\langle x, y \rangle \in \Delta^{\mathcal{I}(w)} \times \Delta^{\mathcal{I}(w)} \mid \forall v. \langle w, v \rangle \in \gamma_i \rightarrow \langle x, y \rangle \in R^{\mathcal{I}(v)}\}.
\end{aligned}
$$

The choice of constant domain is realized by constraining $\Delta^{\mathcal{I}(w)} = \Delta^{\mathcal{I}(v)}$ for all $w, v \in \mathcal{W}$. In the case of expanding domain we have that $\Delta^{\mathcal{I}(w)} \subseteq \Delta^{\mathcal{I}(v)}$ whenever $\langle w, v \rangle \in \gamma_i$ for some modality of the dimension i. At this point it is possible to define the notion of *satisfiability* of a formula.

Definition 5.4. Given a formula ϕ, a Kripke structure $\mathcal{K} = \langle \mathcal{W}, \Gamma, \mathcal{I} \rangle$ and a world $w \in \mathcal{W}$ the *truth relation* $\mathcal{K}, w \models \phi$ is defined inductively by:

$$
\begin{array}{ll}
\mathcal{K}, w \models C \doteq D & \text{iff } C^{\mathcal{I}(w)} = D^{\mathcal{I}(w)} \\
\mathcal{K}, w \models a : C & \text{iff } a^{\mathcal{I}(w)} \in C^{\mathcal{I}(w)} \\
\mathcal{K}, w \models aRb & \text{iff } \langle a^{\mathcal{I}(w)}, b^{\mathcal{I}(w)} \rangle \in R^{\mathcal{I}(w)} \\
\mathcal{K}, w \models \phi \wedge \psi & \text{iff } \mathcal{K}, w \models \phi \ \wedge \ \mathcal{K}, w \models \psi \\
\mathcal{K}, w \models \neg\phi & \text{iff } \mathcal{K}, w \not\models \phi \\
\mathcal{K}, w \models \Diamond_i \phi & \text{iff } \exists v. \langle w, v \rangle \in \gamma_i \ \wedge \ \mathcal{K}, v \models \phi.
\end{array}
$$

A formula ϕ is *satisfiable* if there is a Kripke structure $\mathcal{K}$ and a world w such that $\mathcal{K}, w \models \phi$. A formula is *valid* if for each world $w \in \mathcal{W}$ then $\mathcal{K}, w \models \phi$.

It should be clear that formulæ in $\mathcal{ALC}_{\mathcal{M}}$ are in a strict correspondence with axioms in DL with the exception that they are assumed to hold just in a single world, while a DL axiom should hold in all possible worlds – like in the case of the temporal description logics defined in the previous Sections. Thus, DL axioms can be captured by *valid* formulæ. The classical reasoning problems of concept satisfiability and concept subsumption can be reduced to formula satisfiability. A concept C is *satisfiable* iff there exists $\mathcal{K}$ and w such that $\mathcal{K}, w \models \neg(C = \bot)$ – indeed, this means that there exists an interpretation $\mathcal{I}$ such that $C^{\mathcal{I}(w)} \neq \emptyset$ for some w. A concept C is *subsumed* by D, $C \sqsubseteq D$, iff the formula $(C \rightarrow D = \top)$ is valid, i.e., $\neg(C \rightarrow D = \top)$ is unsatisfiable. As far as the entailment problem is concerned, two different problems can be defined [48,50]: *Local* and *Global consequence*. The *local consequence* problem – $\Sigma \models \phi$ – is defined as follows: for a finite set of formulæ Σ, $\Sigma \models \phi$ if for every $\mathcal{K}$ and every $w \in \mathcal{K}$, if $\mathcal{K}, w \models \Sigma$ then also $\mathcal{K}, w \models \phi$ – i.e. $\Sigma \models \phi$ iff $\bigwedge \Sigma \wedge \neg\phi$ is not satisfiable. The *global consequence* problem – $\Sigma \models^* \phi$ – is defined as follows: for a finite set of formulæ Σ, $\Sigma \models^* \phi$ if for every interpretation $\mathcal{K}$ $\mathcal{K}, w \models \Sigma$ for every w in $\mathcal{K}$, then also $\mathcal{K}, w \models \phi$ for every w in $\mathcal{K}$. Global consequence is reducible to local consequence (and then to a satisfiability problem) for example when temporal structures are considered and both future ($\Box^+$) and past ($\Box^-$) modalities are present: $\Sigma \models^* \phi$ iff $\Sigma \cup \{\Box^+\psi \mid \psi \in \Sigma\} \cup \{\Box^-\psi \mid \psi \in \Sigma\} \models \phi$.
Note that, the classical logical implication problem in Description Logics, $\Sigma \models C \sqsubseteq D$ where Σ is the knowledge base, is reformulated in a temporal DL as a global consequence: $\Sigma \models^* (C \rightarrow D = \top)$.

Baader and Laux [6] propose a complete and terminating algorithm, based on tableaux calculus, for testing satisfiability of $\mathcal{ALC}_{\mathcal{M}}^-$ formulæ under the expanding domain assumption. The main limitation is that all the modal operators do not satisfy any specific axiom for belief or time (i.e., the modalities are interpreted in the basic modal logic K).

The work of Wolter and Zakharyaschev [48] proves the decidability of satisfiability of $\mathcal{ALC}^-_{\mathcal{M}}$ formulæ when the accessibility relations satisfy the most common conditions for the belief and temporal operators (i.e., when the modalities give rise to the modal systems $K, S4, S4.3, S5, KD45, GL, GL.3$ and the tense logic over linear, discrete and unbounded temporal structures like $\langle \mathcal{N}, \leq \rangle$). They start by considering mono-dimensional description languages and then prove a general transfer theorem for deciding satisfiabily in the multi-dimensional case. Furthermore, they prove decidability of these logics under the constant domain assumption showing that both the varying and the expanding domain assumptions are reducible to it. These decidability results hold true only for the case where rigid designators are assumed. It is still an open problem the case where non-rigid designators are assumed. Finally, they investigate the model properties of the different logics. They show how for logics based on linear temporal models (i.e., $\langle \mathcal{N}, \leq \rangle$) or models whose accessibility relations are transitive and reflexive ($S4, S4.3$) the *Finite Model Property* does not hold anymore. As far as global roles are concerned, the satisfiability problem for $\mathcal{ALC}_{\mathcal{M}}$ is decidable in the modal systems $K, S5, KD45$ [49].

The Temporal Case. Wolter and Zakharyaschev [48,49] study the language $\mathcal{ALC}_{\mathcal{M}}$ interpreted over temporal stuctures: The formula satisfiability problem in $\mathcal{ALC}^-_{\mathcal{M}}$ is decidable in the class of linear, discrete and unbounded structures [48]; the same problem is undecidable for $\mathcal{ALC}_{\mathcal{M}}$ (i.e., considering modalized roles), or for $\mathcal{ALC}^-_{\mathcal{M}}$ with global roles [49]. Thus, modalized or global roles interpreted over temporal structures are a source of undecidability.

Wolter and Zakharyaschev [50] consider the combination of tense logics and three expressive DLs, $\mathcal{CIQ}, \mathcal{CIO}, \mathcal{CQO}$ [15,16] (see section 2.1) where $\mathcal{C}$ stands for propositional dynamic modal logic (PDL) while $\mathcal{I}$ adds the converse operator, $\mathcal{Q}$ adds qualified cardinality restrictions on roles, and $\mathcal{O}$ adds individuals as concepts constructors (called *nominals* in the modal logic literature [11]). The temporal operators introduced are: Existential Future ($\Diamond^+$), Existential Past ($\Diamond^-$), Universal Future ($\Box^+$), Universal Past ($\Box^-$), Next Instant ($\bigcirc$), Until ($\mathcal{U}$), Since ($\mathcal{S}$)[9]. As for $\mathcal{ALC}^-_{\mathcal{M}}$, the temporal operators can be applied both to concepts and formulæ. The following theorem summarises the obtained results.

Theorem 5.5. The problem of formula satisfiability is decidable for the following languages:

1. $\mathcal{CIQ}_{\mathcal{U},\mathcal{S}}$ assuming $\langle \mathcal{Z}, < \rangle$ as temporal structure.

2. $\mathcal{CIQ}_{\Diamond}$ in strictly linear ordered structures as well as in $\langle \mathcal{Q}, < \rangle$.

[9] The following equivalences hold: $\Diamond^+ C \doteq \top \mathcal{U} C$, $\Diamond^- C \doteq \top \mathcal{S} C$, $\bigcirc C \doteq \bot \mathcal{U} C$, $\Box^+ \doteq \neg \Diamond^+ \neg$, $\Box^- \doteq \neg \Diamond^- \neg$, if the temporal structure is linear and discrete, and $\mathcal{U}$ is the non reflexive until.

3. $\mathcal{CIO}_{\mathcal{U},\mathcal{S}}$,$\mathcal{CQO}_{\mathcal{U},\mathcal{S}}$ in $\langle \mathcal{Z}, < \rangle$.
4. $\mathcal{CIO}_{\diamond}$,$\mathcal{CQO}_{\diamond}$ in strictly linear ordered structures as well as in $\langle \mathcal{Q}, < \rangle$.

The framework presented in this Section in its generality gives us a very useful tool to compare the expressivity of the DL extensions presented until now. Let us consider, as an example, the language proposed by Schild (see Section 5.1). We have one dimension (i.e., the temporal one) with two modalities ($\mathcal{U}, \mathbf{U}$) applicable only in front of concepts. Each interpretation is based on a branching (linear), discrete, unbounded temporal structure under the constant domain and rigid designators assumptions. Analogous choices are the basis of the works presented in Sections 4 whenever interval-based modal operators are considered (dropping the variables in the case of Schmiedel, and Artale and Franconi).

6. Description Logics with Temporal Parts

This Section overviews two representative approaches of the *internal* perspective in adding a temporal dimension to a Description Logic.

6.1. The T-REX *System*

Weida and Litman [45,46] propose T-REX, a loose hybrid integration between Description Logics and temporal constraint networks with the aim of representing and reasoning about plans. Plans are defined as collections of steps (i.e., actions) together with temporal constraints between their duration. Each step is associated with an action type, represented by a generic concept in K-REP – a non-temporal Description Logic [35]. Thus, a plan is seen as a *plan network*: a temporal constraint network in the style of Allen [1], whose nodes, labeled with action types and corresponding to the steps of the plan itself, are associated with time intervals. As an example of plan in T-REX we show the plan of preparing spaghetti marinara:

```
(defplan Assemble-Spaghetti-Marinara
    ((step1  Boil-Spaghetti)
     (step2  Make-Marinara)
     (step3  Put-Together-SM))
    ((step1  (before  meets)  step3)
     (step2  (before  meets)  step3)))
```

This is a plan composed by three actions, i.e., boiling spaghetti, preparing marinara sauce and assembling all things at the end. Temporal constraints between the steps establish the temporal order in doing the corresponding actions. In this sense, T-REX con be classified as a system with an *internal* representation of time: a plan is a collection of temporal parts possibly holding at different times. The notion of `Mortal` can be expressed in this framework as:

(defplan `Mortal`
  ((*alive-state* `ALIVE`)
   (*dead-state* `DEAD`))
  ((*alive-state* (meets) *dead-state*)))

It is worth noting that no formal semantics was provided for T-Rex. For a better understanding of the consequences that an internal representation framework may have, the next Section should be enlightening.

A structural plan subsumption is defined, characterized in terms of graph matching, and based on two separate notions of subsumption: terminological subsumption between action types labeling the nodes, and temporal subsumption between interval relationships labeling the arcs. The main application of T-Rex is plan recognition. An individual plan is a network where nodes are individual actions while arcs are labeled with temporal relations. The following is an example of an individual plan of type `Assemble-Spaghetti-Marinara`:

Boil17 before *MakeMarinara1* before *PutTogetherSM27*

The plan library is used to guide plan recognition in a way similar to that proposed by Kautz [32]. According to these ideas, a Closed World Assumption (CWA) is made, assuming that the plan library is complete and an observed plan will be fully accounted for by a single plan. The plan recognition process partitions the plan library into the categories *possible*, *necessary* and *impossible* that describe the status of each plan in the plan library with respect to a given observation. A plan is said *possible* if the observation might eventually be an instance of it, also in case of further refinements of the observation itself. A possible plan which actually represents an observation is also *necessary*; possible but not necessary plans are called *optional* plans. When an observation cannot be an instance of the plan then it is an *impossible* plan. Before any observation is made, all plans are optional except for the plan root, which is obviously necessary.

6.2. Time as Concrete Domain

In the concrete domain extension of Description Logics, abstract individuals (i.e., elements of an abstract domain $\Delta^{\mathcal{I}}$) can now be related to values in a *concrete domain* (e.g., the integers, strings, etc.) via *features* (i.e., functional roles). Furthermore, tuples of concrete values identified by such features can be constrained to satisfy an n-ary predicate over the concrete domain. The first work in this direction is the one of Baader and Hanschke [5]. For example, by choosing the natural numbers with the usual total ordering relation "$\leq$" as a concrete domain, it is possible to express the concept describing managers that, every month, spend more money than they earn: $\big(\mathtt{Manager} \sqcap \forall\mathtt{MONTHLY\text{-}BALANCE}.\exists(\mathtt{INCOME},\mathtt{EXPENSES}).\leq\big)$. Here, `INCOME` and `EXPENSES` are features that map objects to elements of the actual concrete domain – integers in this example.

More formally, Baader and Hanschke propose an extension to $\mathcal{ALC}$, i.e., the so called $\mathcal{ALC}(\mathcal{D})$, where $\mathcal{D}$ stands for the concrete domain. A concrete domain is a pair $\mathcal{D} = (\mathsf{dom}(\mathcal{D}), \mathsf{pred}(\mathcal{D}))$ that consists of a set $\mathsf{dom}(\mathcal{D})$ (the domain), and a set of predicate symbols $\mathsf{pred}(\mathcal{D})$. Each predicate symbol $P \in \mathsf{pred}(\mathcal{D})$ is associated with an arity n and an n-ary relation $P^{\mathcal{D}} \subseteq \mathsf{dom}(\mathcal{D})^n$. The syntax of $\mathcal{ALC}$ is augmented with the following rule:

$$C, D \rightarrow \exists(u_1, \ldots, u_n).P \qquad \text{(concrete predicate)}$$

where $P \in \mathsf{pred}(\mathcal{D})$ is an n-ary predicate name, and $u_1, \ldots, u_n$ are n feature chains. The semantics of the new operator is the following:

$$(\exists(u_1, \ldots, u_n).P)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \mid \langle u_1^{\mathcal{I}}(a), \ldots, u_n^{\mathcal{I}}(a) \rangle \in P^{\mathcal{D}}\}$$

where the interpretation function $\cdot^{\mathcal{I}}$ is extended to map every feature name p to a partial function $p^{\mathcal{I}} : \Delta^{\mathcal{I}} \rightarrow \Delta^{\mathcal{I}} \cup \mathsf{dom}(\mathcal{D})$. Thus, we can review the previous example of the concept defining the managers that spend each month more money than they earn:

$$\big(\mathtt{Manager} \sqcap \forall \mathtt{MONTHLY\text{-}BALANCE}.\exists(\mathtt{INCOME}, \mathtt{EXPENSES}).\leq\big)$$

where "$\leq$" is a binary predicate symbol, and $\mathtt{INCOME}, \mathtt{EXPENSES}$ are features mapping individuals of the abstract domain $\Delta^{\mathcal{I}}$ – i.e., the monthly balance of a manager – into elements of the numeric domain $\mathsf{dom}(\mathcal{D})$ – i.e., the amount of their income and expenses.

Concrete domains are restricted to so-called *admissible* concrete domains in order to keep the inference problems of this extension decidable. We recall that, roughly speaking, a concrete domain $\mathcal{D}$ is called *admissible* iff (1) $\mathsf{pred}(\mathcal{D})$ is closed under negation and contains a unary predicate name $\top$ for $\mathsf{dom}(\mathcal{D})$, and (2) satisfiability of finite conjunctions over $\mathsf{pred}(\mathcal{D})$ is decidable. Given an admissible concrete domain, a sound, complete and terminating resoning technique for checking concept subsumption and satisfiability is devised. In fact, condition (1) says that $\mathsf{pred}(\mathcal{D})$ is a complete language on $\mathcal{D}$ allowing for the reduction of the reasoning services to the problem of checking for knowledge base consistency. The second condition is crucial for the decidability of the resoning procedure since, as a subtask, it will have to decide satisfiability of conjunctions of the form $\bigwedge_{i=1}^{k} P_i(\underline{x}^{(i)})$ in the concrete domain.

In this framework, assuming a concrete domain composed by temporal intervals and the Allen's predicates, the concept of $\mathtt{Mortal}$ can be defined as follows:

$$\begin{aligned}\mathtt{Mortal} \doteq\ & \mathtt{ALIVE\text{-}STATE} : \mathtt{LivingBeing} \sqcap \mathtt{DEAD\text{-}STATE} : (\neg\mathtt{LivingBeing}) \sqcap \\ & \exists(\mathtt{ALIVE\text{-}STATE} \circ \mathtt{HAS\text{-}TIME}, \mathtt{DEAD\text{-}STATE} \circ \mathtt{HAS\text{-}TIME}).\mathsf{meets}\end{aligned}$$

i.e., a mortal is any individual having the property of being alive at some temporal interval that meets some other temporal interval at which the same individual has the property of being dead.

It is important to emphasize a major difference of this approach, that formalizes the *internal* point of view in temporally extending a DL, from the languages surveyed in the previous Sections that adopt an *external* perspective. Temporal Description Logics that follow the external approach provide a logical framework for describing objects whose properties may vary in time. For example, a mortal describes all those objects "*being alive*" at certain time and then "*being dead*" at some following time (we use below the Bettini's language):

$\texttt{Mortal} \doteq \texttt{LivingBeing} \sqcap \Diamond(\mathsf{met\text{-}by}).\ \neg\texttt{LivingBeing}$

In the external approach a *property* is represented by means of a concept expression, which is interpreted as the class of objects having that property at the time of evaluation. Since time is embedded in the semantics, all the language constructors have time-varying extensions, thus the *same* object may satisfy different properties at different times. In the concrete domain approach – just like in the T-Rex system – the representation of time is lifted up to the language level – i.e., there is an *internal* representation of time. Considering the definition of mortal, in order to assign a temporal *property* (e.g., being alive) a *copy* of the object should be created by an explictly functional relation (ALIVE-STATE), and the temporal property will hold for that copy. To specify a validity time for this property an ad-hoc function (called HAS-TIME in the example) associates to the object copy its valid temporal interval. This has the drawback that, as far as an object has different temporal properties, different copies of the object itself (one for each temporal property) will be generated in the internal approach.

We introduce now an example which shows that the $\mathcal{ALC}(\mathcal{D})$ Description Logic is more suitable to describe properties of temporal objects (e.g., intervals) rather than properties of objects varying in time (like in the Mortal example). In [5] the Allen's interval relations is internally defined using the set of real numbers $\mathcal{R}$ together with the predicates $\leq, \leq, \geq, \geq, =, \neq$ as the concrete admissible domain. The Interval concept can be defined as an ordered pair of real numbers by referring to the concrete predicate $\leq$ applied to the features LEFT-HAS-TIME and RIGHT-HAS-TIME:

$\texttt{Interval} \doteq \exists(\texttt{LEFT-HAS-TIME}, \texttt{RIGHT-HAS-TIME}).\ \leq$

Allen's relations are binary relations on two intervals and are represented by the Pair concept which uses the features FIRST and SECOND:

$\texttt{Pair} \doteq \exists\texttt{FIRST.Interval} \sqcap \exists\texttt{SECOND.Interval}$

Now Allen's relation can be easily defined as concepts:

$$
\begin{aligned}
\texttt{C-Equals} &\doteq \texttt{Pair} \sqcap \exists(\texttt{FIRST} \circ \texttt{LEFT-HAS-TIME}, \texttt{SECOND} \circ \texttt{LEFT-HAS-TIME}).= \\
&\phantom{\doteq \texttt{Pair}}\ \sqcap \exists(\texttt{FIRST} \circ \texttt{RIGHT-HAS-TIME}, \texttt{SECOND} \circ \texttt{RIGHT-HAS-TIME}).= \\
\texttt{C-Before} &\doteq \texttt{Pair} \sqcap \exists(\texttt{FIRST} \circ \texttt{RIGHT-HAS-TIME}, \texttt{SECOND} \circ \texttt{LEFT-HAS-TIME}).\leq \\
\texttt{C-Meets} &\doteq \texttt{Pair} \sqcap \exists(\texttt{FIRST} \circ \texttt{RIGHT-HAS-TIME}, \texttt{SECOND} \circ \texttt{LEFT-HAS-TIME}).= \\
\ldots
\end{aligned}
$$

An estension to the language $\mathcal{ALC}(\mathcal{D})$ was studied by Haarslev, Lutz and Möller [26]. The new language, called $\mathcal{ALCRP}(\mathcal{D})$, allows for the definition of roles based on properties between concrete objects ($\mathcal{RP}$ stands for $\mathcal{R}$ole definition based on $\mathcal{P}$redicates). The new role-forming operator has the following syntax:

$\exists(u_1, \ldots, u_n)(v_1, \ldots, v_m).P$ (*role forming predicate restriction*)

where $u_1, \ldots, u_n, v_1, \ldots, v_m$ are feature chains, and $P \in \mathit{pred}(\mathcal{D})$ with arity $n+m$. The interpretation function has to be extended in order to satisfy the following equation:

$$
\begin{aligned}
(\exists(u_1, \ldots, u_n)(v_1, \ldots, v_m).P)^{\mathcal{I}} = \{&(a,b) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid \\
&(u_1^{\mathcal{I}}(a), \ldots, u_n^{\mathcal{I}}(a), v_1^{\mathcal{I}}(b), \ldots, v_m^{\mathcal{I}}(b)) \in P^{\mathcal{D}}\}
\end{aligned}
$$

Given an abstract object, say a, it is possible to refer to all those objects whose concrete facet relates with the concrete facet of the starting object a by some specific concrete predicate. In this way we can quantify over these roles using both the existential and the universal quantifier on roles. As a simple example, we can define the `BEFORE` role as being the counterpart of the concrete predicate before in the abstract domain, and use it for defining a new concept `NoBefore`, as the class of objects which do not have any `BEFORE`-related object:

$\texttt{BEFORE} \doteq \exists(\texttt{HAS-TIME})(\texttt{HAS-TIME}).\mathsf{before}$
$\texttt{NoBefore} \doteq \forall\texttt{BEFORE}.\bot$

It is important to point out the difference with the similar definition which can be done using a Description Logic that follows an external approach. As an example, let's consider the Bettini's logic:

$\texttt{NoBefore} \doteq \Box(\mathsf{before}).\ \bot$

Assuming that in both cases the temporal structure is isomorphic to the real numbers $\Re$, while the concept `NoBefore` in the concrete domain approach is satisfiable, denoting all the objects of the *abstract* domain having no `BEFORE`-related objects, the concept `NoBefore` in the external approach is clearly unsatisfiable. The reason is that the concrete domain approach follows an internal temporal representation. This means that we can only quantify over the abstract domain and not over the concrete one, i.e., we can only quantify over the abstract objects which may *possibly* have a specific temporal facet lifted up from the concrete domain. On the other hand, in the logics that follow an external approach (like the

Bettini's logic) both the abstract objects and the temporal elements are first-class citizens, resulting in a language where it is possible to quantify on both abstract objects and temporal elements. A partial study on the relative expressive power between the languages $\mathcal{TL}$-$\mathcal{ALCF}$ (see Section 4.3) and $\mathcal{ALCF}(\mathcal{D})$ has been conducted [4]. In particular, it has been proved how the satisfiability of a $\mathcal{TL}$-$\mathcal{ALCF}$ concept can be reduced to the satisfiability of some corresponding concept in the language $\mathcal{ALCF}(\mathcal{D})$. The limit of this result is that the encoding preserves only satisfiability, and it does not clarify the real relationships between the two languages with respect to the problems of subsumption and logical implication.

As far as the computational properties are concerned, Lutz [34] proves that concept satisfiability, subsumption and ABox consistency for the logics $\mathcal{ALC}(\mathcal{D})$ and $\mathcal{ALCF}(\mathcal{D})$ are PSpace-complete – provided that satisfiability in the concrete domain is in PSpace. In the case of $\mathcal{ALCRP}(\mathcal{D})$, it has been proved the undecidability of reasoning in the full language [26]. However, the authors propose a restricted language for which a sound, complete and terminating reasoning procedure based on tableaux calculus is presented. The undecidability result is due to the interaction of complex roles with existential and universal restrictions.

Finally we should mention the flexibility of these appoaches in order to represent a variety of concrete domains: every concrete domain can be embedded in these logics as far as it is admissible. Indeed, $\mathcal{ALCRP}(\mathcal{D})$ has been considered both over the Allen's algebra [33], and over a spatial concrete domain [26] where the concrete predicates are the binary spatial relations of the RCC-8 theory [38].

7. State-change based Description Logics

In this section we illustrate the approaches where the temporal dimension is only implicit in the language. Both languages presented below describe essentially sets of linearly ordered objects. Time has no first-class citizenship in these representation languages.

7.1. The Clasp *System*

Devambu and Litman [17,18] describe the Clasp system (CLASsification of Scenarios and Plans), a DL system extending the notion of subsumption and classification to plans, in order to build an efficient information retrieval system. Clasp was used to represent plan-like knowledge in the domain of telephone switching software by extending the use of the software information system LASSIE [19]. Clasp is designed for representing and reasoning about large collections of plan descriptions, using a language able to express ordering, conditional and looping operators. Following the Strips tradition, plan descriptions are built starting from states and actions, both represented by using the Classic [12] Description Logic. The simplest action is represented by the atomic Classic concept `Action` (Figure 13), which constrains every kind of action to have at

```
(DEFINE-CONCEPT Action
    (PRIMITIVE (AND Classic-Thing
                    (AT-LEAST 1 ACTOR)
                    (ALL ACTOR Agent)
                    (EXACTLY 1 PRECONDITION)
                    (ALL PRECONDITION State)
                    (EXACTLY 1 ADD-LIST)
                    (ALL ADD-LIST State)
                    (EXACTLY 1 DELETE-LIST)
                    (ALL DELETE-LIST State)
                    (EXACTLY 1 GOAL)
                    (ALL GOAL State))))
```

Figure 13. The generic `Action` concept

least one `ACTOR`, all of whose `ACTOR`s are of type `Agent`, whose `PRECONDITION` is of type `State`, whose `ADD-LIST` is of type `State`, whose `DELETE-LIST` is of type `State`, and whose `GOAL` is of type `State`. Note that EXACTLY is an operator introduced by the authors which simulate the conjunction of AT-LEAST and AT-MOST constructs.

State descriptions are restricted to a simple conjunction of atomic CLASSIC concepts. Furthermore, the concept `State` is a predefined atomic CLASSIC concept specializing `Classic-Thing`:

```
(DEFINE-CONCEPT State
    (PRIMITIVE Classic-Thing))
```

Actions and states can be further restricted; for example, the following is the definition for a `System-Act`:

```
(DEFINE-CONCEPT System-Act
    (AND Action
         (ALL ACTOR System-Agent)))
```

which fully defines `System-Act` as the subconcept of `Action` where the fillers of the role `ACTOR` are restricted to belong to the concept `System-Agent`.

A plan is a conceptual description which uses the roles `PLAN-EXPRESSION`, `INITIAL` and `GOAL`. While `INITIAL` and `GOAL` roles can be restricted with CLASSIC concepts, the `PLAN-EXPRESSION` role is restricted to a *plan concept expression* which is compositionally built from CLASSIC actions and states concepts using the operators SEQUENCE, LOOP, REPEAT, TEST, OR and SUBPLAN, as showed by the following syntax rules:

$$
\begin{array}{lll}
< \textit{plan-expression} > & ::= & < \textit{action-concept} > \\
 & | & (\text{SEQUENCE } < \textit{plan-expression} > +) \\
 & | & (\text{LOOP } < \textit{plan-expression} >) \\
 & | & (\text{REPEAT } < \textit{integer} > \ < \textit{plan-expression} >) \\
 & | & (\text{TEST } (< \textit{state-concept} > \ < \textit{plan-expression} >)+) \\
 & | & (\text{OR } < \textit{plan-expression} > +) \\
 & | & (\text{SUBPLAN } < \textit{symbol} >)
\end{array}
$$

where $<$ *action-concept* $>$ and $<$ *state-concept* $>$ refer to CLASSIC concepts subsumed by the concepts `Action` and `State`. The intuitive meaning of the CLASP constructs is clarified by the following examples:

- (SEQUENCE `A B C`): An action of type `A` is followed by an action of type `B`, which is followed by an action of type `C`.
- (LOOP `A`): Zero or more repetitions of actions of type `A`.
- (REPEAT 7 `A`): Equivalent to (SEQUENCE `A A A A A A A`).
- (TEST (`S1 A`) (`S2 B`)): If the current state is of type `S1`, then action type `A`, else if state type `S2`, then action type `B`.
- (OR `A B`): Either action type `A` or type `B`.
- (SUBPLAN `Plan-Name`): Insert the `Plan-Name`'s definition in the current plan-expression.

The root of the plan taxonomy is the following CLASP concept `Plan`:

```
(DEFINE-PLAN Plan
    (PRIMITIVE (AND Clasp-Thing
                    (EXACTLY 1 INITIAL)
                    (ALL INITIAL State)
                    (EXACTLY 1 GOAL)
                    (ALL GOAL State)
                    (EXACTLY 1 PLAN-EXPRESSION)
                    (ALL PLAN-EXPRESSION (LOOP Action)))))
```

More specific plans are built by refining the roles `PLAN-EXPRESSION`, `INITIAL` and `GOAL`. The example domain in which CLASP is tested is that one of telephone switching software. Here we show the plan called `Pots-Plan`:

```
(DEFINE-PLAN Pots-Plan
    (AND Plan
        (ALL PLAN-EXPRESSION
            (SEQUENCE (SUBPLAN Originate-And-Dial-Plan)
                      (TEST (Callee-On-Hook-State
                                (SUBPLAN Terminate-Plan))
                            (Callee-Off-Hook-State
                                (SEQUENCE
                                    Non-Terminate-Act
                                    Caller-On-Hook-Act
                                    Disconnect-Act)))))))
```

which makes use of the previously defined plans `Originate-And-Dial-Plan` and `Terminate-Plan`. Intuitively, the plan `Pots-Plan` describes a situation in which the caller picks up a phone, gets a dial tone, and dials a callee. If the callee's phone is on-hook (TEST on `Callee-On-Hook-State`), the call goes through; if the callee's phone is off-hook (TEST on `Callee-Off-Hook-State`), the caller gets a busy signal, hangs up, and is disconnected.

Clasp gives also the possibility to describe individual plans, called *scenarios*. Every scenario corresponds to an initial state, a final state and a sequence of individual actions. A scenario is *well-formed* if the given sequence of individual actions will indeed transform the specified initial state into the goal state. Whenever a scenario is created, Clasp checks it for well-formedness. During this process any unspecified intermediate state is inferred, and any partially specified intermediate state is completed by using Strips-like rules.

The temporal expressivity of Clasp is implicit in the representation language provided for building plans. In this language you can essentially express a sequential temporal order of actions, where each action is instantaneous. Plan definitions can use disjunction to represent alternative ways of accomplishing a given plan, and iteration constructors to abstractly describe repetitions of the same action.

The authors provided Clasp both with a plan type subsumption algorithm, and with a *plan recognition* algorithm that verifies whether a scenario fulfills the conditions to belong to a given plan type. The key idea in developing both algorithms is the observation that *plan concept expressions* correspond to regular expressions. Clasp is able to trasform each plan expression into a finite state automaton. This correspondence allows the authors to develop algorithms for subsumption and recognition by integrating work in automata theory with work in concept subsumption and recognition. More formally, a plan description, P, subsumes another plan description, Q, if there is a subsumption relation between the expressions restricting the roles `INITIAL`, `GOAL` and `PLAN-EXPRESSION`. Since `INITIAL` and `GOAL` are restricted with Classic concepts standard algorithms for computing terminological subsumption are adopted. The problem of plan expressions subsumption is reduced to finite automata (or regular language) subsumption. The case of plan recognition algorithm is similar. A scenario s satisfies a plan P if the actual fillers of `INITIAL` and `GOAL` satisfy the Classic concept restrictions of the respective roles in P, and the plan expression of s is a string in the language defined by the abstract plan expression of P.

7.2. *The* Rat *System*

Heinsohn, Kudenko, Nebel and Profitlich [30] describe the Rat system, used in the Wip project at the German Research Center for AI (DFKI). They use Description Logics to represent both the world states and the atomic actions. A second formalism is added to compose actions in plans and to reason about simple temporal relationships. Rat actions are defined by the change of the world state they cause, and they are instantaneous as in the Strips-like systems, while plans are linear sequences of actions. Thus, as for Clasp, explicit temporal constraints are not expressible in the language.

Formally, an action is defined as a triple of *parameters*, *pre-conditions* and *post-conditions*, $\langle pars, pre, post\rangle$. As an example, consider:

$$
\begin{array}{l}
\texttt{PutCupUnderWaterOutlet} \doteq \\
\quad \langle(\texttt{agent} : \texttt{Person} \sqcap \texttt{object} : \texttt{Cup} \sqcap \texttt{machine} : \texttt{EspressoMachine}), \\
\quad\ (\texttt{object} \circ \texttt{position} \downarrow \texttt{agent} \circ \texttt{has-hand} \circ \texttt{inside-region}), \\
\quad\ (\texttt{object} \circ \texttt{position} \downarrow \texttt{machine} \circ \texttt{has-water-outlet} \circ \texttt{under-region})\rangle
\end{array}
$$

where `agent`, `object` and `machine` are the formal parameters of the action; the pre-conditions state that the `Cup` is held by the agent's hand; the post-conditions state that the `Cup` is located under the water-outlet.

Actions can be composed to build *plan schemata*. A plan schema is a triple of action parameters, sequence of actions, and equality constraints over the action parameters, $\langle pars, seq, cons\rangle$:

$$
\begin{array}{l}
\texttt{MakeEspresso} \doteq \\
\quad \langle(\texttt{agent} : \texttt{Person} \sqcap \texttt{object1} : \texttt{Cup} \sqcap \texttt{object2} : \texttt{EspressoMachine}), \\
\quad\ (\ldots, \\
\quad\ (A5 : \texttt{PutCupUnderWaterOutlet}), \\
\quad\ (A6 : \texttt{TurnSwitchToEspresso}), \\
\quad\ \ldots), \\
\quad\ (\texttt{object2} \downarrow A5 \circ \texttt{machine} \sqcap \texttt{object2} \downarrow A6 \circ \texttt{machine})\rangle
\end{array}
$$

The plan `MakeEspresso` has one `agent` and two `objects` as parameters. Each sub-action – or subplan – in the sequence is prefixed by a label that will be used as an index in the constraint part which, in this case, states that the `EspressoMachine`, the `machine` of the action `PutCupUnderWaterOutlet`, and the `machine` of the action `TurnSwitchToEspresso` are the same object.

The state representation in Rat uses the *feature* construct – i.e., a functional role – to describe the parameters of an action. Furthermore, the possibility to express equality constraints between paths of features is a powerful mechanism in order to bind action parameters. While the language for the state representation is more expressive than the one used in CLASP, the language for composing plans is much richer in CLASP than in RAT, which only allows for sequences.

The most important reasoning services offered by RAT are the *simulated execution* of parts of a plan, and the *feasibility* checking of a plan. The feasibility test is similar to the usual consistency check for a concept description: the pre- and post-conditions of individual actions composing the plan are *temporally projected*, respectively backwards and forward. This procedure differs from the same service offered in CLASP due to the richer expressiveness of state descriptions. If the feasibility test does not lead to an inconsistent initial, final or intermediate state, the plan is feasible and the global pre- and post-conditions are determined as a side effect.

8. Conclusions

In this paper we have presented an overview of the various approaches found in the literature to represent temporal knowledge using Description Logics. The most interesting and general approaches are based on the combination of a standard Description Logic with some temporal logic, either interval-based or point-based. We have first analyzed the results for interval-based temporal description logics. It turns out that a full fledged interval-based logic is undecidable [28,10]. However, there are still fragments which are interesting from the application point of view which have been found decidable [3]. On the other hand, point-based logics have nicer computational properties [40,50]. The most expressive decidable logic is $\mathcal{CIQ}_{\mathcal{U},\mathcal{S}}$, which allows for temporal operators on concepts and formulæ. In general, having temporal operators on the role side leads to undecidability when considering the usual temporal structures [49]. The combination of a description logic with a temporal concrete domain [26] has been pointed out as a promising framework.

As a suggestion for the development of the field, it might be then very interesting to devise a general framework subsuming most of the work presentd here, based on the combination of a Description Logic, seen as a propositional modal logic over unconstrained frames, with some propositional temporal logic, over various temporal structures and including some notion of time granularity.

References

[1] J.F. Allen, Temporal reasoning and planning, in J.F. Allen, H.A. Kautz, R.N. Pelavin, and J.D.Tenenberg, editors, *Reasoning about Plans*, chapter 1, pages 2–68. Morgan Kaufmann, 1991.

[2] A. Artale and E. Franconi, A computational account for a description logic of time and action, in J.Doyle, E.Sandewall, and P.Torasso, editors, *Proc. of the 4 th International Conference on Principles of Knowledge Representation and Reasoning*, pages 3–14, Bonn, Germany, May 1994. Morgan Kaufmann.

[3] A. Artale and E. Franconi, A temporal description logic for reasoning about actions and plans, *J. of Artificial Intelligence Research (JAIR)*, 9, 1998.

[4] A. Artale and C. Lutz, A correspondence between temporal description logics, in *Workshop Notes of the Int. Workshop on Description Logics, DL-99*, pages 145–149, Linköping, Sweden, July 1999.

[5] F. Baader and P. Hanschke, A scheme for integrating concrete domains into concept languages, in *Proc. of the 12 th IJCAI*, pages 446–451, Sidney, Australia, 1991.

[6] F. Baader and A. Laux, Terminological logics with modal operator, in *Proc. of the 13 th IJCAI*, pages 808–814, Montreal, Canada, 1995.

[7] F. Baader and H.J. Ohlbach, A multidimensional terminological knowledge representation language, *J. of Applied Non-Classical Logic*, 5:153–197, 1995.

[8] F. Baader, H.J. Bürckert, J. Heinsohn, B. Hollunder, J. Müller, B. Nebel, W. Nutt, and H.J. Profitlich, Terminological knowledge representation: a proposal for a terminological logic, Technical Memo TM-90-04, DFKI, Saarbrücken, Germany, December 1990.

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

[10] C. Bettini, Time dependent concepts: Representation and reasoning using temporal description logics, *Data & Knowledge Engineering*, 22(1):1–38, 1997.

[11] P. Blackburn, Nominal tense logic, *Notre Dame Journal of Formal Logic*, 34:56–83, 1993.

[12] R.J. Brachman, D.L. McGuiness, P.F. Patel-Schneider, L. Alperin Resnick, and Alexander Borgida, Living with CLASSIC: When and how to use a KL-ONE-like language, in J. Sowa, editor, *Principles of Semantic Networks*. Morgan Kaufmann, 1991.

[13] J.P. Burgess, Basic tense logic, in D.Gabbay and F.Guenther, editors, *Handbook of Philosophical Logic*, volume 2, pages 89–133. Reidel, Dordrecht, Holland, 1984.

[14] D. Calvanese, G. De Giacomo, M. Lenzerini, and D. Nardi, Reasoning in expressive description logics, in Alan Robinson and Andrei Voronkov, editors, *Handbook of Automated Reasoning*. Elsevier Science Publishers (North-Holland), Amsterdam, 1999.

[15] G. De Giacomo and M. Lenzerini, Tbox and abox reasoning in expressive description logics, in *Proc. of the 5 th International Conference on Principles of Knowledge Representation and Reasoning*, pages 316–327, Boston, MA, 1996. Morgan Kaufmann.

[16] G. De Giacomo, *Decidability of Class-Based Representation Formalisms*, PhD thesis, Univ. "La Sapienza", Roma, 1995.

[17] P.T. Devanbu and D.J. Litman, Plan-based terminological reasoning, in *Proc. of the 2 nd International Conference on Principles of Knowledge Representation and Reasoning*, pages 128–138, Cambridge, MA, May 1991. Morgan Kaufmann.

[18] P.T. Devanbu and D.J. Litman, Taxonomic plan reasoning, *Artificial Intelligence*, 84:1–35, 1996.

[19] P.T. Devanbu, R.J. Brachman, P. Selfridge, and B. Ballard, Taxonomic plan reasoning. *Communication of the ACM*, 34(5), 1991.

[20] F.M. Donini, M. Lenzerini, D. Nardi, and W. Nutt, The complexity of concept languages, Technical Report RR-95-07, DFKI, Germany, June 1995. A preliminary version appears in Proc. of the 2^{nd} International Conference on Principles of Knowledge Representation and Reasoning (KR-91).

[21] F.M. Donini, M. Lenzerini, D. Nardi, and A. Schaerf, Reasoning in description logic, in Gerhard Brewka, editor, *Principles of Knowledge Representation and Reasoning*, Studies in Logic, Language and Information, pages 193–238. CSLI Publications, 1996.

[22] E.A. Emerson and J.Y. Halpern, Decision procedures and expressiveness in the temporal logic of branching time, *Journal of Computer and System Science*, 30:1-24, 1985.

[23] M. Finger and D. Gabbay, Adding a temporal dimension to a logic system. *Journal of Logic, Language and Information*, 2:203–233, 1992.

[24] D. Gabbay, I. Hodkinson, and M. Reynolds, *Temporal Logic.* Oxford University Press, 1994.

[25] R. Goldblatt, *Logics of Time and Computation*, CSLI Publications, 1983.

[26] V. Haarslev, C. Lutz, and R. Möller, Foundation of spatiotemporal reasoning with description logics, in A.Cohn, L.Schubert, and S.C.Shapiro, editors, *Proc. of the 6 th International Conference on Principles of Knowledge Representation and Reasoning*, pages 112–123, Trento, Italy, June 1998. Morgan Kaufmann.

[27] J. Y. Halpern and Y. Moses, A guide to the modal logic of knowledge and belief: Preliminary draft, in *Proc. of the 9 th IJCAI*, pages 480–490, Los Angeles, CA, 1985.

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

[29] D. Harel, Dynamic Logic, in D.Gabbay and F.Guenther, editors, *Handbook of Philosophical Logic*, volume 2, pages 497-604. Reidel, Dordrecht, Holland, 1984.

[30] J. Heinsohn, D. Kudenko, B. Nebel, and H.J. Profitlich, RAT: representation of actions using terminological logics, Technical report, DFKI, Saarbrücken, Germany, November 1992.

[31] B. Hollunder and W. Nutt, Subsumption algorithms for concept languages, Technical Report RR-90-04, DFKI, Germany, April 1990.

[32] H.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.

[33] C. Lutz, V. Haarslev, and R. Möller, A concept language with role-forming predicate restrictions, Technical Report FBI-HH-M-276/97, University of Hamburg – Computer Science Department, Germany, december 1997.

[34] C. Lutz, Reasoning with concrete domains, in *Proc. of the 16 th IJCAI*, pages 90–95, Stockholm, Sweden, 1999.

[35] E. Mays, R. Dionne, and R. Weida, K-Rep system overview, *SIGART Bulletin*, 2(3):93–97, 1991.

[36] B. Nebel, Terminological reasoning is inherently intractable, *Artificial Intelligence*, 43:235–249, 1990.

[37] B. Nebel, Terminological cycles: Semantics and computational properties, in John F. Sowa, editor, *Principles of Semantic Networks*, chapter 11, pages 331–362. Morgan Kaufmann, 1991.

[38] D. A. Randell, Z. Cui, and A. G. Cohn, A spatial logic based on regions and connections, in *Proc. of the 3 rd International Conference on Principles of Knowledge Representation and Reasoning*, pages 522–527, Cambridge, MA, 1992. Morgan Kaufmann.

[39] K.D. Schild, A correspondence theory for terminological logics: Preliminary report, in *Proc. of the 12 th IJCAI*, pages 466–471, Sidney, Australia, 1991.

[40] K.D. Schild, Combining terminological logics with tense logic, in *Proceedings of the 6th Portuguese Conference on Artificial Intelligence, EPIA'93*, October 1993.

[41] M. Schmidt-Schauß and G. Smolka, Attributive concept descriptions with complements, *Artificial Intelligence*, 48(1):1–26, 1991.

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

[43] J. van Benthem, *The Logic of Time*, D. Reidel Publishing Company, Dordrecht, Holland, 1983.

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

[45] R. Weida and D. Litman, Terminological reasoning with constraint networks and an ap-

plication to plan recognition, in *Proc. of the 3 rd International Conference on Principles of Knowledge Representation and Reasoning*, pages 282–293, Cambridge, MA, October 1992. Morgan Kaufmann.

[46] R. Weida and D. Litman, Subsumption and recognition of heterogeneous constraint networks, in *Proceedings of CAIA-94*, 1994.

[47] P. Wolper, The tableaux method for temporal logic: An overview, *Logique et Analyse*, 110-111:119–136, 1985.

[48] F. Wolter and M. Zakharyaschev. Satisfiability problem in description logics with modal operators, in A.Cohn, L.Schubert, and S.C.Shapiro, editors, *Proc. of the 6 th International Conference on Principles of Knowledge Representation and Reasoning*, pages 512–523, Trento, Italy, June 1998. Morgan Kaufmann.

[49] F. Wolter and M. Zakharyaschev. Modal description logics: Modalizing roles, *Fundamentæ Informaticæ*, 39:411–438, 1999.

[50] F. Wolter and M. Zakharyaschev, Temporalizing description logics, in D. Gabbay and M. de Rijke, editors, *Frontiers of Combining Systems*. Studies Press-Wiley, 1999. Also in the Proceedings of *FroCoS'98*, Amsterdam, 1998.

[51] W.A. Woods and J.G. Schmolze, The KL-ONE family. *Computer and Mathematics with Applications, special issue: Semantic Networks in Artificial Intelligence*, 23(2-5):133–177, March-May 1992.