A Correspondence between Temporal Description Logics

Alessandro Artale
ITC-IRST
Povo (TN), I
artale@irst.itc.it

Carsten Lutz
LuFG Theoretical Computer Science
RWTH Aachen, DE
clu@cantor.informatik.rwth-aachen.de

1 Introduction

Description Logics (DLs) are formalisms for representing and reasoning about conceptual knowledge. There exist several extensions of DLs for an appropriate integration of temporal knowledge [4]. This paper investigates the relation between the two DLs $\mathcal{TL}$-$\mathcal{ALCF}$ [2, 3] and $\mathcal{ALCF}(\mathcal{D})$ [10, 8]. $\mathcal{TL}$-$\mathcal{ALCF}$ is an interval-based, temporal DL for reasoning about objects whose properties vary over time. $\mathcal{ALCF}(\mathcal{D})$ is a logic for integrated reasoning about conceptual and so-called concrete knowledge. If instantiated with a "temporal" concrete domain, $\mathcal{ALCF}(\mathcal{D})$ is well-suited for reasoning about temporal objects, i.e., objects which have a unique temporal extension.

This paper is a first attempt to clarify the relationship between these two formalisms. It is showed that satisfiability of $\mathcal{TL}$-$\mathcal{ALCF}$ concepts can be reduced to satisfiability of $\mathcal{ALCF}(\mathcal{D})$ concepts. This allows to use the available $\mathcal{ALCF}(\mathcal{D})$ tableau calculus for reasoning with $\mathcal{TL}$-$\mathcal{ALCF}$. Furthermore, it allows to settle the complexity of satisfiability of $\mathcal{TL}$-$\mathcal{ALCF}$ concepts, which was previously unknown.

The paper is organized as follows. Sections 2 and 3 introduce the syntax and semantics of the two temporal DLs. In Section 4, a normal form for $\mathcal{TL}$-$\mathcal{ALCF}$ concepts is introduced. Based on this normal form, the reduction of satisfiability of $\mathcal{TL}$-$\mathcal{ALCF}$ concepts to satisfiability of $\mathcal{ALCF}(\mathcal{D})$ concepts is given.

2 The Logic $\mathcal{TL}$-$\mathcal{ALCF}$

The language $\mathcal{TL}$-$\mathcal{ALCF}$ [3] is composed of the interval-based temporal logic $\mathcal{TL}$ and the non-temporal description logic $\mathcal{ALCF}$. The logic $\mathcal{TL}$ is able to represent temporal constraint networks based on Allen's relations, and to relate $\mathcal{ALCF}$ concept expressions with time intervals in these networks. *$\mathcal{TL}$-$\mathcal{ALCF}$ concepts* (denoted by C, D) are built following the syntax rules in Figure 1. Throughout this paper, we use C and D to denote temporal concepts, E and F for non-temporal concepts, f for non-parametric features, $\star g$ for parametric features, p for paths, and R for roles (all possibly with index). The $\star$ symbol is not intended as an operator, but only used to distinguish parametric from non-parametric features. For the basic temporal interval relations, Allen's notation [1] is used: 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), and finished-by (fi). Temporal variables are introduced by the temporal existential quantifier "$\diamond$". The special temporal variable $\sharp$, usually called NOW, is intended as the reference interval.

$$
\begin{array}{llcl}
\mathcal{TL} & C, D & \to & E \mid C \sqcap D \mid C@X \mid C[Y]@X \mid \diamond(\overline{X})\overline{Tc}.C \\
 & Tc & \to & (X\ (r)\ Y) \mid (X\ (r)\ \sharp) \mid (\sharp\ (r)\ Y) \\
 & \overline{Tc} & \to & Tc \mid Tc\ \overline{Tc} \\
 & r, s & \to & r\ ,\ s \mid \mathsf{s} \mid \mathsf{mi} \mid \mathsf{f} \mid \ldots \\
 & X, Y & \to & \mathrm{x} \mid \mathrm{y} \mid \mathrm{z} \mid \ldots \\
 & \overline{X} & \to & X \mid X\ \overline{X} \\
\mathcal{ALCF} & E, F & \to & A \mid \neg E \mid E \sqcap F \mid E \sqcup F \mid \forall R.E \mid \exists R.E \mid \\
 & & & p : E \mid p{\downarrow}q \mid p{\uparrow}q \mid p\uparrow \\
 & p, q & \to & f \mid \star g \mid p \circ q
\end{array}
$$

Figure 1: Syntax rules for $\mathcal{TL}$-$\mathcal{ALCF}$

$\mathcal{TL}$-$\mathcal{ALCF}$ 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 partial 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 $[u, v] \doteq \{x \in \mathcal{P} \mid u \leq x \leq v, u \neq v\}$ 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:

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

Parametric features differ from features for being independent from time.

The *temporal interpretation function* $\cdot^\mathcal{E}$ defined in the

$$
\begin{aligned}
(\mathsf{s})^{\mathcal{E}} &= \{\langle [u,v],[u_1,v_1]\rangle \in \mathcal{T}^{\star}_{<} \times \mathcal{T}^{\star}_{<} \mid u = u_1 \wedge v < v_1\} \\
&\ldots \textit{(similarly for the other Allen relations)} \\
(r,\ s)^{\mathcal{E}} &= r^{\mathcal{E}} \cup s^{\mathcal{E}} \\
\langle \overline{X}, \overline{Tc}\rangle^{\mathcal{E}} &= \{\mathcal{V} : \overline{X} \mapsto \mathcal{T}^{\star}_{<} \mid \ \forall (X\ (R)\ Y) \in \overline{Tc}.\ \langle \mathcal{V}(X), \mathcal{V}(Y)\rangle \in (R)^{\mathcal{E}}\}
\end{aligned}
$$

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

Figure 2: The $\mathcal{TL}$-$\mathcal{ALCF}$ semantics.

upper half of Figure 2 depends only on the temporal structure $\mathcal{T}$. A labeled directed graph $\langle \overline{X}, \overline{Tc}\rangle$, where $\overline{X}$ is a of variables representing the nodes and $\overline{Tc}$ is a set of temporal constraints representing the arcs, is called *temporal constraint network*. An interpretation of a temporal constraint network is a set of variable assignments that satisfy the temporal constraints. A *variable assignment* is a function $\mathcal{V} : \overline{X} \mapsto \mathcal{T}^{\star}_{<}$ associating an interval to a temporal variable. A temporal constraint network is *consistent* if it admits a non empty interpretation. The notation $\langle \overline{X}, \overline{Tc}\rangle^{\mathcal{E}}_{\{x_1 \mapsto t_1, x_2 \mapsto t_2, \ldots\}}$, used to interpret concept expressions, denotes the subset of $\langle \overline{X}, \overline{Tc}\rangle^{\mathcal{E}}$ where the variable x_i is mapped to the interval value t_i.

An *interpretation function* $\cdot^{\mathcal{I}}_{\mathcal{V},t,\mathcal{H}}$ for generic concepts, based on a variable assignment $\mathcal{V}$, an interval t, and a set of constraints $\mathcal{H} = \{x_1 \mapsto t_1, \ldots\}$ over the assignments of free variables, extends the primitive interpretation function in such a way that the equations of Figure 2 are satisfied – operators that can be obtained by negation are omitted. Intuitively, the interpretation of a concept $C^{\mathcal{I}}_{\mathcal{V},t,\mathcal{H}}$ is the set of elements of the domain which are of type C at the time interval t, with the assignment for the free temporal variables in C given by $\mathcal{V}$ (c.f. the definition of $(C@X)^{\mathcal{I}}_{\mathcal{V},t,\mathcal{H}}$) and with the constraints for the assignment of variables in the scope of the outermost temporal quantifiers given by $\mathcal{H}$. The *natural interpretation function* $C^{\mathcal{I}}_t$, being equivalent to the interpretation function $C^{\mathcal{I}}_{\mathcal{V},t,\mathcal{H}}$ with any $\mathcal{V}$ such that $\mathcal{V}(\sharp) = t$, and $\mathcal{H} = \emptyset$, is introduced as an abbreviation. An interpretation $\mathcal{I}$ is a *model* for a concept C if, for some $t \in \mathcal{T}^{\star}_{<}$, $C^{\mathcal{I}}_t \neq \emptyset$. If a concept has a model, then it is *satisfiable*, otherwise it is *unsatisfiable*.

We will now informally discuss the intended *meaning* of $\mathcal{TL}$-$\mathcal{ALCF}$ concepts. Concept expressions are interpreted over pairs of *temporal intervals* and *individuals* $\langle i, a\rangle$, meaning that the individual a is in the extension of the concept at the interval i. Within a concept expression, the special "$\sharp$" variable denotes the current interval of evaluation. The temporal existential quantifier "$\Diamond$" introduces interval variables, related to each other and possibly to the $\sharp$ variable in a way defined by the set of *temporal constraints*. 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 $\sharp$ in the concept expression C is interpreted as the X variable. Please consider the following example from the blocks world domain which defines a concept representing the action of stacking a block on top of another block.

$$
\begin{aligned}
&\texttt{Basic-Stack} \doteq \Diamond(x\ y)(x\ \mathsf{meets}\ \sharp)(\sharp\ \mathsf{meets}\ y). \\
&\qquad \big((\star\texttt{BLOCK} : \texttt{OnTable})@x \sqcap (\star\texttt{BLOCK} : \texttt{OnBlock})@y\big)
\end{aligned}
$$

`Basic-Stack` denotes 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 stacking action. The temporal constraints $(x\ \mathsf{m}\ \sharp)$ and $(\sharp\ \mathsf{m}\ y)$ state that the interval $\mathcal{V}(x)$ should meet the interval $\mathcal{V}(\sharp)$ – the occurrence interval of the action type `Basic-Stack` – and that $\mathcal{V}(\sharp)$ should meet $\mathcal{V}(y)$. 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. Whereas the existence and identity of the $\star$`BLOCK` of the action is time invariant, it can be in the extension of different concepts in different intervals of time, e.g., the $\star$`BLOCK` is necessarily `OnTable` only during the interval $\mathcal{V}(x)$.

3 The Logic $\mathcal{ALCF}(A)$

Description logics represent knowledge on an abstract, logical level. So-called concrete domains provide a means to additionally represent "concrete information" such as, e.g., numbers or time intervals, and allow for integrated reasoning about both kinds of knowledge. In [6], the basic description logic incorporating concrete domains, $\mathcal{ALC}(\mathcal{D})$, is introduced. The logic $\mathcal{ALCF}(\mathcal{D})$ [10] extends $\mathcal{ALC}(\mathcal{D})$ by agreement and disagreement on features. Similar to $\mathcal{ALC}(\mathcal{D})$, an "admissible" concrete domain $\mathcal{D}$ yields decidability of $\mathcal{ALCF}(\mathcal{D})$. Before $\mathcal{ALCF}(\mathcal{D})$ is introduced, the definition of concrete domains is recalled.

Definition 3.1. A *concrete domain* $\mathcal{D}$ is a pair $(\Delta_{\mathcal{D}}, \Phi_{\mathcal{D}})$, where $\Delta_{\mathcal{D}}$ is a set called the domain, and $\Phi_{\mathcal{D}}$ is a set of predicate names. Each predicate name P in $\Phi_{\mathcal{D}}$ is associated with an arity n and an n-ary predicate $P^{\mathcal{D}} \subseteq \Delta^n_{\mathcal{D}}$. A concrete domain $\mathcal{D}$ is called *admissible* iff (1) the set of its predicate names is closed under negation and contains a name $\top_{\mathcal{D}}$ for $\Delta_{\mathcal{D}}$ and (2) the satisfiability problem for finite conjunctions of predicates is decidable.

The syntax of $\mathcal{ALCF}(\mathcal{D})$ is obtained from the syntax of $\mathcal{ALCF}$ as given in Figure 1 by adding an additional syntax rule for the *predicate operator*:

$$E, F \rightarrow \exists p_1, \ldots, p_n.P$$

where $P \in \Phi_{\mathcal{D}}$ is an n-ary predicate name, and $p_1, \ldots, p_n$ are paths.

An $\mathcal{ALCF}(\mathcal{D})$ *interpretation* $\mathcal{I} = (\Delta_{\mathcal{I}}, \cdot^{\mathcal{I}})$ consists of a set $\Delta_{\mathcal{I}}$ (the *abstract domain*) which is disjoint from $\Delta_{\mathcal{D}}$ and an interpretation function $\cdot^{\mathcal{I}}$. The interpretation function maps each concept name C to a subset $C^{\mathcal{I}}$ of $\Delta_{\mathcal{I}}$, each role name R to a subset $R^{\mathcal{I}}$ of $\Delta_{\mathcal{I}} \times \Delta_{\mathcal{I}}$, and each feature name f to a partial function $f^{\mathcal{I}}$ from $\Delta_{\mathcal{I}}$ to $\Delta_{\mathcal{D}} \cup \Delta_{\mathcal{I}}$. Parametric features are identical to non-parametric features w.r.t their $\mathcal{ALCF}(\mathcal{D})$ interpretation. If $p = f_1 \cdots f_k$ is a feature chain, then $p^{\mathcal{I}}$ is defined as the composition $f_1^{\mathcal{I}} \circ \cdots \circ f_k^{\mathcal{I}}$ of the partial functions $f_1^{\mathcal{I}}, \ldots, f_k^{\mathcal{I}}$. Each complex concept is interpreted as usual (i.e., as in Figure 2 with the temporal indices omitted) while the new predicate operator has the following meaning:

$$(\exists p_1, \ldots, p_n.P)^{\mathcal{I}} = \{a \in \Delta_{\mathcal{I}} \mid \exists x_1, \ldots, x_n \in \Delta_{\mathcal{D}} : \\ (a, x_1) \in p_1^{\mathcal{I}} \wedge \cdots \wedge (a, x_n) \in p_n^{\mathcal{I}} \wedge (x_1, \ldots, x_n) \in P^{\mathcal{D}}\}$$

In this paper, we consider the logic $\mathcal{ALCF}(A)$, i.e., $\mathcal{ALCF}(\mathcal{D})$ instantiated with the temporal concrete domain A. The concrete domain A is based on intervals and Allen's relations (hence the name "A"). Formally, A is defined as (Δ_A, Φ_A), where Δ_A is the interval set $\mathcal{T}^{\star}_{<}$ as defined in Section 2, and Φ_A contains:

- the unary predicates $\top_A, \bot_A$ denoting Δ_A and $\emptyset$.
- 13 binary predicates $\mathsf{b}, \mathsf{m}, \mathsf{d}, \ldots$ corresponding to Allen's 13 basic relations. The extensions $\mathsf{b}^A, \mathsf{m}^A, \mathsf{d}^A, \ldots$ are defined analogously to the interpretation of Allen's relations by $(\cdot)^{\mathcal{E}}$ in Figure 2.
- a binary predicate $\mathsf{r}_1\text{-}\cdots\text{-}\mathsf{r}_k$ for each disjunction $\mathsf{r}_1 \vee \cdots \vee \mathsf{r}_k$ of Allen relations $\mathsf{r}_1, \ldots, \mathsf{r}_k$ including the empty disjunction *empty-rel*. The extension of a disjunctive predicate $(\mathsf{r}_1\text{-}\cdots\text{-}\mathsf{r}_k)^A$ is $\mathsf{r}_1^A \cup \cdots \cup \mathsf{r}_k^A$; furthermore, $\textit{empty-rel}^A = \emptyset \times \emptyset$.

In [8], it is proved that the concrete domain A is admissible and that satisfiability of $\mathcal{ALCF}(A)$ concepts is PSPACE-complete.

In the framework of $\mathcal{ALCF}(A)$, a basic stack action similar to the one in Section 2 can be defined as follows:

$$\begin{aligned} \mathtt{Basic\text{-}Stack} \doteq\ & \mathtt{step}_1 : (\mathtt{BLOCK} : \mathtt{OnTable}) \sqcap \\ & \mathtt{step}_2 : (\mathtt{BLOCK} : \mathtt{OnBlock}) \sqcap \\ & \exists (\mathtt{step}_1 \circ \mathtt{time}), (\mathtt{step}_\sharp \circ \mathtt{time}).\mathsf{m} \sqcap \\ & \exists (\mathtt{step}_\sharp \circ \mathtt{time}), (\mathtt{step}_2 \circ \mathtt{time}).\mathsf{m} \end{aligned}$$

The concept states that any `Basic-Stack` is related to three objects via the features $\mathtt{step}_1, \mathtt{step}_2$, and $\mathtt{step}_\sharp$. These objects describe the basic stack action at different time intervals – with $\mathtt{step}_\sharp$ representing the occurring time of the action, often called the "current" interval. For each step, a corresponding time interval is associated by the `time` feature. The relation between these time intervals is described using the predicate operator and resembles the temporal network in the $\mathcal{TL}\text{-}\mathcal{ALCF}$ definition of the basic stack.

Comparing the two definitions of `Basic-Stack`, their main difference can be characterized as follows: In the $\mathcal{TL}\text{-}\mathcal{ALCF}$ definition, the basic stack is represented by a single logical object which is "temporal", i.e., whose properties are defined separately for each temporal interval. To the contrary, in $\mathcal{ALCF}(A)$, the basic stack is represented by a logical "meta-object" (the `Basic-Stack` object itself in the above concept definition) and a set of additional logical objects each of which has unique properties and represents the basic stack at a unique time interval. A reduction from $\mathcal{TL}\text{-}\mathcal{ALCF}$ to $\mathcal{ALCF}(A)$, as defined in the next Section, has to bridge this discrepancy. Furthermore, it has to capture the temporal invariance of parametric features. The basic stack as defined above is not to be intended as a translation of the $\mathcal{TL}\text{-}\mathcal{ALCF}$ `Basic-Stack`.

4 A Tableau for SAT in $\mathcal{TL}\text{-}\mathcal{ALCF}$

The logic $\mathcal{ALCF}(A)$ is provided with a sound and complete tableau calculus which is optimal w.r.t. worst case complexity [10]. To obtain a tableau calculus and establish complexity results for $\mathcal{TL}\text{-}\mathcal{ALCF}$, we will build on the $\mathcal{ALCF}(A)$ calculus. This section shows how to reduce satisfiability of $\mathcal{TL}\text{-}\mathcal{ALCF}$ concepts to satisfiability of $\mathcal{ALCF}(A)$ concepts.

As a starting point for the reduction to be devised, we do not consider arbitrary $\mathcal{TL}\text{-}\mathcal{ALCF}$ concepts but only those in a certain normal form. In [3], it is shown that every $\mathcal{TL}\text{-}\mathcal{ALCF}$ concept can be reduced to an equivalent concept in *existential form*, i.e., of the form $\diamond(\overline{X})\overline{Tc}.Q_0 \sqcap Q_1@X_1 \sqcap \ldots \sqcap Q_n@X_n$. In the existential form, the only temporal operator that may occurr is a single "$\diamond$" operator, while each Q_i is an $\mathcal{ALCF}$ concept. The normal form for a $\mathcal{TL}\text{-}\mathcal{ALCF}$ concept is obtained by starting from its existential form, and then applying simple form, and path explicitation steps.

Definition 4.1 (Normal form). Given a concept in existential form, its *Normal Form (NF)* is obtained by sequentially applying the following tranformations.

(*Simple Form*) Transform each Q_i into the equivalent *simple form* following the rewrite rules reported in [7]. A concept in simple form contains only complements of the form $\neg A$, where A is a primitive concept, and no sub-concepts of the form $p\uparrow$, where p is a path with length greater than one. This corresponds to a first order formula in negation normal form.

(*Path Explicitation*) Apply the following normalization rules which make explicit all the possible chains of features.

$$p:(C\sqcap D)\to p:C\sqcap p:D \qquad p:(C\sqcup D)\to p:C\sqcup p:D$$
$$p:(q:C)\to(p\circ q):C \qquad p:(q_1{\downarrow}q_2)\to p\circ q_1{\downarrow}p\circ q_2$$
$$p:(q_1{\uparrow}q_2)\to p\circ q_1{\uparrow}p\circ q_2$$

For example, the normal form of the $\mathcal{ALCF}$ concept $p:(q:C\sqcap\neg f:D)$ is $p\circ q:C\sqcap(p:f\uparrow\sqcup\, p\circ f:\neg D)$. (note that the simple form of $\neg f:D$ is $f\uparrow\sqcup f:\neg D$).

Proposition 4.2 (Equivalence of NF). *Every concept C can be reduced into an equivalent concept in normal form.*

In the following, a satisfiability preserving translation Ψ from $\mathcal{TL}$-$\mathcal{ALCF}$ concepts in NF to $\mathcal{ALCF}(A)$ concepts is given. Let γ denote features (parametric or non-parametric). Given a $\mathcal{TL}$-$\mathcal{ALCF}$ concept C in normal form, i.e., $\diamond(\overline{X})\overline{Tc}.Q_0\sqcap Q_1@X_1\sqcap\ldots\sqcap Q_n@X_n$, $\Psi(C)$ is obtained as follows:

1. Let $\overline{Tc}$ be $\{(X_1\ r_1\ Y_1),\ldots,(X_k\ r_k\ Y_k)\}$ and let $f_0,\ldots,f_n$ be features not used in C. The mapping α from $\mathcal{TL}$-$\mathcal{ALCF}$ temporal constraints to $\mathcal{ALCF}(A)$ concepts is defined as follows:
$$\alpha(X_i\ r\ X_j)=\exists(f_i\circ time),(f_j\circ time).r$$
$$\alpha(X_i\ r\ \sharp)=\exists(f_i\circ time),(f_0\circ time).r$$
$$\alpha(\sharp\ r\ X_i)=\exists(f_0\circ time),(f_i\circ time).r$$
Define C_T as $\alpha(X_1\ r_1\ Y_1)\sqcap\ldots\sqcap\alpha(X_k\ r_k\ Y_k)$.
2. Let *Path* be the set of paths used in the concept C. For each $0\le i\le n$, the mapping $\Phi_i: Path\to Path\cup\{f_i\}$, with f_i as introduced in Point 1, is defined in Figure 3.
3. For each $0\le i\le n$, the mapping Ψ_i which maps $\mathcal{ALCF}$ concepts in normal form to $\mathcal{ALCF}$ concepts in normal form is defined in Figure 3.
4. Let $\star g_0,\ldots,\star g_m$ be the parametric features used in C. Define C_F as
$$\prod_{i=0}^{m}\left((\sqcap_{j=0}^{n}f_j\circ\star g_i\uparrow)\sqcup(\sqcap_{j=1}^{n}(f_0\circ\star g_i){\downarrow}(f_j\circ\star g_i))\right).$$
5. Define two concepts Ω and Ω' as follows:
$$\Omega=f_0:\Psi_0(Q_0)\sqcap\ldots\sqcap f_n:\Psi_n(Q_n)$$
$$\Omega'=\prod_{0\le i<j\le n}(\exists(f_i\circ time),(f_j\circ time).=)\to f_i:\Psi_i(Q_j)$$
where $E\to F$ is an abbreviation for $\neg E\sqcup F$. We are now ready to assemble the concept $\Psi(C)$: $\Psi(C)=C_F\sqcap C_T\sqcap\Omega\sqcap\Omega'$

$$\begin{aligned}
\Phi_i(\gamma) &:= \gamma\\
\Phi_i(\gamma\circ\star g\circ p) &:= \gamma\circ\Phi_i(\star g\circ p)\\
\Phi_i(\gamma\circ f\circ p) &:= \gamma\circ f_i\circ\Phi_i(f\circ p)\\
\\
\Psi_i(A) &:= A\\
\Psi_i(\neg A) &:= \neg\,\Psi_i(A)\\
\Psi_i(\star g\uparrow) &:= \star g\uparrow\\
\Psi_i(f\uparrow) &:= f\uparrow\\
\Psi_i(D\sqcap E) &:= \Psi_i(D)\ \sqcap\ \Psi_i(E)\\
\Psi_i(D\sqcup E) &:= \Psi_i(D)\ \sqcup\ \Psi_i(E)\\
\Psi_i(p:D) &:= \begin{cases}\Phi_i(p):\Psi_i(D) & \text{if } D=\star g\uparrow\\ \Phi_i(p)\circ f_i:\Psi_i(D) & \text{otherwise}\end{cases}\\
\Psi_i(p{\downarrow}q) &:= \Phi_i(p){\downarrow}\Phi_i(q)\\
\Psi_i(p\uparrow q) &:= \Phi_i(p)\uparrow\Phi_i(q)\\
\Psi_i(\exists R.D) &:= \exists R.\Psi_i(D)\\
\Psi_i(\forall R.D) &:= \forall R.\Psi_i(D)
\end{aligned}$$

Figure 3: Definition of $\Phi_i(p)$ and $\Psi_i(Q)$ mappings.

The main idea behind the reduction has already been discussed at the end of Section 3: A (temporal) object o, which is in the extension of a $\mathcal{TL}$-$\mathcal{ALCF}$ concept C, is reflected by a "meta-object" o' and a set of objects $O'=\{o'_0,\ldots,o'_n\}$ on the $\mathcal{ALCF}(A)$ side, where each o'_i represents o' at a different time interval. The features $f_0,\ldots,f_i$ are introduced during the translation in order to relate o' with the objects in O' (o'_i is an f_i-filler of o' for $0\le i\le n$). Each object in O' has a unique time interval associated via the *time* feature. The object o'_0 represents o at the current time interval. The concept C_T ensures that the temporal relations between the associated intervals are as defined by the temporal constraint network $\langle\overline{X},\overline{Tc}\rangle$. Additional care has to be taken in order to deal correctly with parametric features. Since they are time-independent, it has to be assured that all $\mathcal{ALCF}(A)$ objects in O' have identical fillers of parametric features. This is done by the C_F concept together with the mappings Ψ_i and Φ_i. Furthermore, it is possible that different objects in O' are associated with the same interval. In this case, they both describe the object o at the same time interval, and, hence, should be identical w.r.t. concept membership. The concept Ω' ensures this.

As an example, the translation of the `Basic-Stack` concept as introduced in Section 2 is given.

$\Psi(\texttt{Basic-Stack})\doteq$
$f_0\circ\star\texttt{BLOCK}{\downarrow}f_1\circ\star\texttt{BLOCK}\sqcap$
$f_0\circ\star\texttt{BLOCK}{\downarrow}f_2\circ\star\texttt{BLOCK}\sqcap$
$\exists(f_1\circ\texttt{time}),(f_0\circ\texttt{time}).\mathsf{m}\sqcap$
$\exists(f_0\circ\texttt{time}),(f_2\circ\texttt{time}).\mathsf{m}\sqcap$
$f_1:\star\texttt{BLOCK}\circ f_1:\texttt{OnTable}\sqcap$

$f_2 : \star\texttt{BLOCK} \circ f_2 : \texttt{OnBlock} \sqcap$
$(\exists(f_0 \circ \texttt{time}), (f_1 \circ \texttt{time}).=) \rightarrow f_0 : \star\texttt{BLOCK} \circ f_0 : \texttt{OnTable} \sqcap$
$(\exists(f_0 \circ \texttt{time}), (f_2 \circ \texttt{time}).=) \rightarrow f_0 : \star\texttt{BLOCK} \circ f_0 : \texttt{OnBlock} \sqcap$
$(\exists(f_1 \circ \texttt{time}), (f_2 \circ \texttt{time}).=) \rightarrow f_1 : \star\texttt{BLOCK} \circ f_1 : \texttt{OnBlock}$

The following proposition, together with the next theorem, shows the main result of this paper.

Proposition 4.3. *The $\mathcal{ALCF}(A)$ concept $\Psi(C)$ is satisfiable if and only if the $\mathcal{TL}$-$\mathcal{ALCF}$ concept C is satisfiable.*

A proof of this proposition can be found in [5]. As already noted in Section 3, satisfiability of $\mathcal{ALCF}(A)$ concepts is a PSpace-complete problem [8]. Together with the reduction given above, the following theorem is an obvious consequence.

Theorem 4.4. *Satisfiability of $\mathcal{TL}$-$\mathcal{ALCF}$ concepts is* PSpace*-complete.*

Proof: The reduction given above proves that satisfiability of $\mathcal{TL}$-$\mathcal{ALCF}$ concepts is in PSpace. Since $\mathcal{TL}$-$\mathcal{ALCF}$ contains $\mathcal{ALC}$ as a proper fragment, it is also PSpace-hard and hence PSpace-complete. $\square$

Please note that in [3], another PSpace-completeness result for the satisfiability of $\mathcal{TL}$-$\mathcal{ALCF}$ concepts is given (Proposition 7.5). It does, however, only apply to concepts in the so-called completed existential form. The problem with this result is that converting a $\mathcal{TL}$-$\mathcal{ALCF}$ concept to this normal form involves converting it to disjunctive normal form which results in the worst case in an exponential blowup in size.

5 Conclusion

We have presented a reduction of the satisfiability of $\mathcal{TL}$-$\mathcal{ALCF}$ concepts to the satisfiability of $\mathcal{ALCF}(A)$ concepts. This allows to use tableau based algorithms for reasoning with $\mathcal{TL}$-$\mathcal{ALCF}$. Furthermore, it allows to identify the satisfiability of $\mathcal{TL}$-$\mathcal{ALCF}$ concepts as a PSpace-complete problem.

For the reduction, we considered plain concepts without reference to TBoxes. From the results in [9], it follows that satisfiability of $\mathcal{TL}$-$\mathcal{ALCF}$ concepts w.r.t. simple TBoxes is NExpTime-hard.[1] It is, however, yet to be proven that it is also in NExpTime.

Actually computing the given reduction is a polynomial problem. In fact, we consider it very likely that efficient implementations of the reduction can be found. As future work, we plan to extend the reduction to the subsumption and ABox consistency problem. This is not a trivial task since full negation is not available in the temporal part of the logic $\mathcal{TL}$-$\mathcal{ALCF}$.

[1] This refers to acyclic TBoxes where the left-hand sides are unique and comprised of atomic concepts, only. It also follows from results in [9] that $\mathcal{TL}$-$\mathcal{ALCF}$ extended by general TBoxes (GCIs) yields a logic for which concept satisfiability is undecidable.

Acknowledgements The work presented in this paper was partially supported by the "Foundations of Data Warehouse Quality" (*DWQ*) European ESPRIT IV Long Term Research (LTR) Project 22469.

References

[1] J.F. Allen. Temporal reasoning and planning. In James F. Allen, Henry A. Kautz, Richard N. Pelavin, and Josh D. Tenenberg, editors, *Reasoning about Plans*, chapter 1, pages 2–68. Morgan Kaufmann, San Mateo, California, 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} KR*, 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. *Journal of Artificial Intelligence Research*, 9:463–506, 1998.

[4] A. Artale and E. Franconi. Temporal description logics. In L. Vila, P. van Beek, M. Boddy, M. Fisher, D. M. Gabbay, A. Galton, and R. Morris, editors, *Handbook of Time and Temporal Reasoning in Artificial Intelligence.* MIT Press, To appear.

[5] A. Artale and C. Lutz. A correspondence between temporal description logics. LTCS-Report 99-10, LuFG Theoretical Computer Science, RWTH Aachen, Germany, 1999.

[6] F. Baader and P. Hanschke. A scheme for integrating concrete domains into concept languages. In *Proc. of the 12^{th} IJCAI*, pages 452–457, Sidney, Australia, 1991.

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

[8] C. Lutz. The complexity of reasoning with concrete domains. LTCS-Report 99-01, LuFG Theoretical Computer Science, RWTH Aachen, Germany, 1999.

[9] C. Lutz. On the complexity of terminological reasoning. LTCS-Report 99-04, LuFG Theoretical Computer Science, RWTH Aachen, Germany, 1999.

[10] C. Lutz. Reasoning with concrete domains. In *Proc. of the 16^{th} IJCAI*, Stockholm, Sweden, July 31 – August 6, 1999.

[11] P. van Beek and D.W. Manchak. The design and experimental analysis of algorithms for temporal reasoning. *Journal of Artificial Intelligence Research*, 4:1–18, January 1996.