Computing Generalizations of Temporal $\mathcal{EL}$ Concepts with Next and Global

Satyadharma Tirtarasa
Technische Universität Dresden
Dresden, Germany
satyadharma.tirtarasa@tu-dresden.de

Anni-Yasmin Turhan
Technische Universität Dresden
Dresden, Germany
anni-yasmin.turhan@tu-dresden.de

ABSTRACT

In ontology-based applications, the authoring of complex concepts or queries written in a description logic (DL) is a difficult task. An established approach to generate complex expressions from examples provided a user, is the bottom-up approach. This approach employs two inferences: the *most specific concept* (MSC), which generalizes an ABox individual into a concept and the *least common subsumer* (LCS), which generalizes a collection of concepts into a single concept. In ontology-based situation recognition the situation to be recognized is formalized by a DL query using temporal operators and that is to answered over a sequence of ABoxes. Now, while the bottom-up approach is well-investigated for the DL $\mathcal{EL}$, there are so far no methods for temporalized DLs.

We consider here the temporalized DL that extends the DL $\mathcal{EL}$ with the LTL operators next (**X**) and global (**G**) and we present an approach that extends the LCS and the MSC to the temporalized setting. We provide computation algorithms for both inferences–even in the presence of rigid symbols–and show their correctness.

CCS CONCEPTS

• **Computing methodologies** → **Description logics**; **Temporal reasoning**; *Ontology engineering*;

KEYWORDS

Description Logics, generalization, temporal reasoning

ACM Reference Format:
Satyadharma Tirtarasa and Anni-Yasmin Turhan. 2022. Computing Generalizations of Temporal $\mathcal{EL}$ Concepts with Next and Global. In *The 37th ACM/SIGAPP Symposium on Applied Computing (SAC '22), April 25–29, 2022, Virtual Event,* . ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3477314.3507136

1 INTRODUCTION

In description logic knowledge bases (KBs) each concept corresponds to a notion from the application domain written in a particular DL. Generally, concepts correspond to unary predicates and roles to binary relationships. A (complex) concept is an expression that combines concepts and roles by concept constructor available in the DL. The TBox part of a DL KB, states concept inclusions, which are sub-concept relationships. The ABox part of a DL KB, describes concrete objects. Two prominent DL reasoning problems are *subsumption*, i.e. to decide for two given concepts, if one is a specialization of the other, and *instance checking*, i.e. to decide whether a given individual is an instance of a given concept. For the DL $\mathcal{EL}$ that admits conjunction and existential restrictions (a form of existential quantification) as concept constructors, subsumption and instance checking can be decided in polynomial time [3].

When KBs or complex query concepts are built, it is often difficult for users to come up with a formulation of the concept they intend to use. Often it is easier to select example instances of the intended concept. An approach to generate complex concepts from a set of examples is the *bottom-up approach* [4] where the resulting concept is computed by applying two inferences. The first is the *most specific concept* (MSC), which computes from an ABox individual a concept (in a given DL) that is the least w.r.t. subsumption. The second inference is the *least common subsumer* (LCS), which computes a concept from a set of input concepts that subsumes all input concepts and is the least concept w.r.t. subsumption. Applying the MSC to all examples selected by the user and then applying the LCS results in a concept description of the selected examples.

LCS and MSC are typically studied for DLs that do not admit disjunction as a concept constructor, since in the presence of disjunction the LCS is simply the disjunction of the input concepts and thus not informative. Both inferences have been investigated for *unfoldable TBoxes*, which are TBoxes that use assign complex concepts to concept names by acyclic definitions. In this case the TBox can simply be treated in a pre-processing step that essentially "unfolds" the concept definitions from the TBox. For $\mathcal{EL}$ and some of its extensions, the LCS and MSC were investigated in [4, 5, 7, 10]. If a cyclic TBox is used, the LCS need not exist, since cyclic concepts cannot be expressed by finite concepts. For cyclic ABoxes the MSC need also not exist, due to the same reason [7]. Conditions for the existence of the LCS (and the MSC) in $\mathcal{EL}$ w.r.t. general TBoxes have given in [11]. However, often an approximation of the LCS or the MSC can be sufficient. Methods for computing the LCS or the MSC up to a given nesting depth of quantification, i.e. up to a role depth bound have been studied in [6–8].

In this paper we provide a first study on the computation of the LCS and the MSC for temporalized DLs. Temporalized DLs have been intensively investigated in the last decade mainly motivated by the application of ontology-based situation recognition. In such applications, the situation to be recognized is formalized as a query with temporal operators in the query language. The task of situation recognition is then to answer a temporalized query over a DL knowledge base. Since reasoning in temporalized DLs can easily

SAC '22, April 25–29, 2022, Virtual Event,

ACM ISBN 978-1-4503-8713-2/22/04…$15.00
https://doi.org/10.1145/3477314.3507136

become undecidable, the query is answered over a classical TBox and a sequence of classical ABoxes. The sequence captures the temporal data and models observations made over time. See [1, 2] for recent surveys on reasoning in this kind of setting. Many of the results for reasoning have been achieved for combinations of linear temporal logic (LTL) [9] and a range of DLs.

Example-based learning in temporalized DLs has so far not been studied in the literature, although automated support for authoring temporal query concepts (or temporal conjunctive queries) would be very helpful for situation recognition applications. In this paper, we lift the LCS and MSC to the temporal setting. We consider the logic $\text{LTL}^{\mathbf{X},\mathbf{G}}_{\mathcal{EL}}$, which uses the temporal operators next (**X**) and global (**G**) from LTL together with $\mathcal{EL}$ constructors. Our choice of temporal operators is motivated by the fact, that operators that imply a disjunction such as until (**U**) or finally (**F**), render the LCS again uninformative, since the resulting temporal concept would simply enumerate the variants found in the temporal data. Thus we concentrate on the "deterministic" temporal operators next (**X**) and global (**G**). Intuitively, the semantics of a $\text{LTL}^{\mathbf{X},\mathbf{G}}_{\mathcal{EL}}$ concept is interpreted in two dimensions. The $\mathcal{EL}$ constructors express relations to other elements in object domain, while the LTL operators express the evolution of objects in the temporal dimension.

We consider here only empty or unfoldable TBoxes as the LCS or MSC need not exist in the presence of general TBoxes. However, our techniques admit the computation of approximations of the LCS and the MSC by a role depth bound, if general TBoxes (or cyclic ABoxes) are used. Reasoning in temporal logics usually distinguishes whether the signature contains rigid symbols, whose extension cannot change over time, or not. We consider here the cases without and with rigid concepts and rigid roles.

The bottom-up approach in the temporal setting means that the user selects a set of example individuals from the sequence of ABoxes and each individual is generalized into a $\text{LTL}^{\mathbf{X},\mathbf{G}}_{\mathcal{EL}}$ concept by the MSC and subsequently these are generalized into a single $\text{LTL}^{\mathbf{X},\mathbf{G}}_{\mathcal{EL}}$ concept by the computing their LCS. The general approach for computing the LCS and the MSC in $\text{LTL}^{\mathbf{X},\mathbf{G}}_{\mathcal{EL}}$ follows the approach in [4]. To compute the LCS, the input concepts are normalized such that the consequences of the interaction of the concept constructors are made explicit. The normalized concepts are then represented by description trees and their cross-product yields (a representation of) the LCS. We devise a new normal form that treats the interaction of next and global. The central part here is our characterization of subsumption for $\text{LTL}^{\mathbf{X},\mathbf{G}}_{\mathcal{EL}}$ concepts, which we use to show the correctness of our LCS computation algorithm. To compute the MSC of an ABox individual from a sequence of ABoxes, is essentially to extract the connected component starting from that individual. The re-occurrence of an individual in the next ABox in the temporal sequence is modeled by connecting the individual and its re-occurrence by a new role reflecting the temporal information. This construction can be approximated for cyclic ABoxes, if a role depth bound is provided. In order to show the correctness of our MSC algorithm, we develop a characterization of the instance relationship in $\text{LTL}^{\mathbf{X},\mathbf{G}}_{\mathcal{EL}}$.

2 PRELIMINARIES

We briefly recall $\mathcal{EL}$ and propositional LTL to define the temporal DL $\text{LTL}^{\mathbf{X},\mathbf{G}}_{\mathcal{EL}}$ that extends $\mathcal{EL}$ concepts with **X** (next) and **G** (global).

The Description Logic $\mathcal{EL}$. Let $\mathsf{N_C}$, $\mathsf{N_R}$, $\mathsf{N_I}$ be sets of *concept names*, *role names* and *individual names*, respectively. Let $A \in \mathsf{N_C}$, $r \in \mathsf{N_R}$ and let C and D be $\mathcal{EL}$ concepts. Then $\mathcal{EL}$ *concepts* are defined by the following grammar:

$$C, D ::= A \mid C \sqcap D \mid \exists r.C \mid \top$$

An *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consists of a nonempty domain $\Delta^{\mathcal{I}}$ and a function $\cdot^{\mathcal{I}}$ that maps every $A \in \mathsf{N_C}$ to a subset $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, every $r \in \mathsf{N_R}$ to a relation $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, and every individual name $a \in \mathsf{N_R}$ to an element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$. Complex $\mathcal{EL}$ concepts are interpreted as follows: $(\top)^{\mathcal{I}} = \Delta^{\mathcal{I}}$, $(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$, and $(\exists r.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \exists y.y \in \Delta^{\mathcal{I}} \text{ s.t. } (x,y) \in r^{\mathcal{I}} \text{ and } y \in C^{\mathcal{I}}\}$.

Let C and D be $\mathcal{EL}$ concepts, $r \in \mathsf{N_R}$, and $a, b \in \mathsf{N_I}$. An $\mathcal{EL}$ *concept inclusion* (CI) is of the form $C \sqsubseteq D$. An $\mathcal{EL}$ *concept assertion* is of the form $C(a)$ and a *role assertion* of the form $r(a,b)$. An interpretation $\mathcal{I}$ satisfies: a concept assertion $C(a)$ if $a^{\mathcal{I}} \in C^{\mathcal{I}}$; a role assertion $r(a,b)$ if $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$; and a CI $C \sqsubseteq D$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$.

The LTL Fragment $\text{LTL}^{\mathbf{X},\mathbf{G}}$. Let P be a set of propositional variables. $\text{LTL}^{\mathbf{X},\mathbf{G}}$ formulae are defined inductively: every propositional variable is an $\text{LTL}^{\mathbf{X},\mathbf{G}}$ formula; and if ϕ and ψ are $\text{LTL}^{\mathbf{X},\mathbf{G}}$ formulae, then $\phi \wedge \psi$ (conjunction), $\mathbf{X}\,\phi$ (next), and $\mathbf{G}\,\phi$ (global) are $\text{LTL}^{\mathbf{X},\mathbf{G}}$ formulae. The semantics of $\text{LTL}^{\mathbf{X},\mathbf{G}}$ formulae is based on the notion of an LTL structure. An LTL *structure* is a sequence $\mathfrak{I} = (w_i)_{i \geq 0}$ of worlds $w_i \subseteq P$. Intuitively, w_i is a set of propositional variables that are true at time point i. The validity of an $\text{LTL}^{\mathbf{X},\mathbf{G}}$ formula ϕ in an LTL structure $\mathfrak{I}$ at time point $i \geq 0$ (denoted $\mathfrak{I}, i \models \phi$) is defined inductively:

- $\mathfrak{I}, i \models p$ for $p \in P$ iff $p \in w_i$;
- $\mathfrak{I}, i \models \phi \wedge \psi$ iff $\mathfrak{I}, i \models \phi$ and $\mathfrak{I}, i \models \psi$;
- $\mathfrak{I}, i \models \mathbf{X}\,\phi$ iff $\mathfrak{I}, i+1 \models \phi$;
- $\mathfrak{I}, i \models \mathbf{G}\,\phi$ iff $\mathfrak{I}, j \models \phi$ for all $j \geq i$.

An $\text{LTL}^{\mathbf{X},\mathbf{G}}$ formula ϕ is satisfiable if there exists an LTL structure $\mathfrak{I}$ s.t. $\mathfrak{I}, 0 \models \phi$. Deciding satisfiability for LTL is PSpace-complete, but it is trivial for $\text{LTL}^{\mathbf{X},\mathbf{G}}$, due to the absence of negation.

The Temporal DL $\text{LTL}^{\mathbf{X},\mathbf{G}}_{\mathcal{EL}}$. Let $A \in \mathsf{N_C}$ and $r \in \mathsf{N_R}$. $\text{LTL}^{\mathbf{X},\mathbf{G}}_{\mathcal{EL}}$ *concepts* are defined by the following grammar:

$$C, D ::= A \mid C \sqcap D \mid \exists r.C \mid \mathbf{X}\,C \mid \mathbf{G}\,C \mid \top.$$

A *TBox* $\mathcal{T}$ a finite set of *concept inclusions* (CIs) $C \sqsubseteq D$. An *ABox* $\mathcal{A}$ is a finite set of *concept assertions* $C(a)$ and *role assertions* $r(a,b)$ where $a, b \in \mathsf{N_I}$. An *axiom* is either a CI or an assertion.

The semantics of $\text{LTL}^{\mathbf{X},\mathbf{G}}_{\mathcal{EL}}$ is based on temporal interpretations, which extends LTL structures. A *temporal interpretation* is a sequence $\mathfrak{I} = (\mathcal{I}_i)_{0 \leq i}$ of interpretations $\mathcal{I}_i = (\Delta, \cdot^{\mathcal{I}_i})$ over a common domain Δ and that respects rigid individual names, i.e., $a^{\mathcal{I}_i} = a^{\mathcal{I}_j}$ for all $a \in \mathsf{N_I}$ and $i, j \geq 0$. The top concept $(\top)^{\mathcal{I}_i} = \Delta$. Complex $\text{LTL}^{\mathbf{X},\mathbf{G}}_{\mathcal{EL}}$ concepts are interpreted as follows:

- $(C \sqcap D)^{\mathcal{I}_i} = C^{\mathcal{I}_i} \cap D^{\mathcal{I}_i}$
- $(\exists r.C)^{\mathcal{I}_i} = \{x \in \Delta \mid \exists y : (x,y) \in r^{\mathcal{I}_i} \text{ and } y \in C^{\mathcal{I}_i}\}$
- $(\mathbf{X}\,C)^{\mathcal{I}_i} = \{x \in \Delta \mid x \in C^{\mathcal{I}_{i+1}}\}$

- $(\mathbf{G}\, C)^{\mathcal{I}_i} \quad = \{x \in \Delta \mid x \in C^{\mathcal{I}_j} \text{ for all } j \geq i\}$

A temporal interpretation $\mathfrak{I}$ at time point i *satisfies* axiom α (denoted $\mathfrak{I}, i \models \alpha$) of the form: CI $C \sqsubseteq D$ iff $C^{\mathcal{I}_i} \subseteq D^{\mathcal{I}_i}$; concept assertion $C(a)$ iff $a^{\mathcal{I}_i} \in C^{\mathcal{I}_i}$; and role assertion $r(a, b)$ iff $(a^{\mathcal{I}_i}, b^{\mathcal{I}_i}) \in r^{\mathcal{I}_i}$.

$\mathfrak{I} = (\mathcal{I}_i)_{0 \leq i}$ is a *model of a concept* C if C is satisfied at time point 0, i.e., $C^{\mathcal{I}_0} \neq \emptyset$. $\mathfrak{I}$ is a *model of a CI* $C \sqsubseteq D$ iff $\mathfrak{I}, i \models C \sqsubseteq D$ for all $i \geq 0$. $\mathfrak{I}$ is a *model of an ABox* $\mathcal{A}_i$ at time point i iff it is a model of all $\alpha \in \mathcal{A}_i$ at time point i. $\mathfrak{I} = (\mathcal{I}_i)_{0 \leq i}$ is a *model of a sequence of ABoxes* $(\mathcal{A}_i)_{0 \leq i \leq n}$ iff it is a model of all ABox $\mathcal{A}_i$, $0 \leq i \leq n$. We sometimes abbreviate $(\mathcal{A}_i)_{0 \leq i \leq n}$ with $\vec{\mathcal{A}}_n$.

In temporal settings, some concepts or roles do not change over time. We use $\mathsf{N}_\mathsf{C}^{\mathsf{rig}} \subseteq \mathsf{N}_\mathsf{C}$ for the set of *rigid concepts* and $\mathsf{N}_\mathsf{R}^{\mathsf{rig}} \subseteq \mathsf{N}_\mathsf{R}$ for the set of *rigid roles.* The elements in $\mathsf{N}_\mathsf{C} \setminus \mathsf{N}_\mathsf{C}^{\mathsf{rig}}$ or $\mathsf{N}_\mathsf{R} \setminus \mathsf{N}_\mathsf{R}^{\mathsf{rig}}$ are called *flexible.* A temporal interpretation $\mathfrak{I} = (\mathcal{I}_i)_{0 \leq i}$ respects rigid concepts (roles) iff $A^{\mathcal{I}_p} = A^{\mathcal{I}_q}$ ($r^{\mathcal{I}_p} = r^{\mathcal{I}_q}$) for all $A \in \mathsf{N}_\mathsf{C}^{\mathsf{rig}}$ ($r \in \mathsf{N}_\mathsf{R}^{\mathsf{rig}}$) and all $0 \leq p \leq q$.

Every $\mathrm{LTL}_{\mathcal{EL}}^{\mathbf{X},\mathbf{G}}$ concept is satisfiable. Two DL reasoning problems are the basis for the LCS and the MSC. C *subsumes* D ($C \sqsubseteq D$) w.r.t. $\mathcal{T}$ iff for all models $\mathfrak{I}$ of $\mathcal{T}$, $C^{\mathfrak{I}} \subseteq D^{\mathfrak{I}}$ holds. Given concept C, individual a, and time point i, *instance checking* tests whether $a^{\mathcal{I}_i} \in C^{\mathcal{I}_i}$ holds for all models of $\mathcal{T} \cup \vec{\mathcal{A}}_n$, denoted by $\vec{\mathcal{A}}_n, i \models_{\mathcal{T}} C(a)$. We assume TBox $\mathcal{T}$ is unfolded into concept descriptions.

Definition 2.1 (LCS & MSC). An $\mathrm{LTL}_{\mathcal{EL}}^{\mathbf{X},\mathbf{G}}$ concept D is the *least common subsumer* (LCS) of the concepts $C_1, \ldots, C_n$ ($\mathbf{lcs}(C_1, \ldots, C_n)$ for short) iff it satisfies

- $C_i \sqsubseteq D$ for all $i = 1, \ldots, n$, and
- D is the least concept with this property, i.e., if an $\mathrm{LTL}_{\mathcal{EL}}^{\mathbf{X},\mathbf{G}}$ concept E satisfies $C_i \sqsubseteq E$ for all $i = 1, \ldots, n$, then $D \sqsubseteq E$.

An $\mathrm{LTL}_{\mathcal{EL}}^{\mathbf{X},\mathbf{G}}$ concept D is the *most specific concept* (MSC) of the individual a w.r.t. the sequence of ABoxes $\vec{\mathcal{A}}_n$ at time point i ($MSC_i(a)$ for short) iff

- $\vec{\mathcal{A}}_n, i \models D(a)$, and
- D is the least concept with this property, i.e., if E is an $\mathrm{LTL}_{\mathcal{EL}}^{\mathbf{X},\mathbf{G}}$ concept satisfying $\vec{\mathcal{A}}_n, i \models E(a)$, then $D \sqsubseteq E$.

In combination, the MSC and the LCS facilitate the learning of an $\mathrm{LTL}_{\mathcal{EL}}^{\mathbf{X},\mathbf{G}}$ concept from a set of individuals described in a sequence of ABoxes. To ease presentation, we focus on the computation of the LCS from two input concepts w.l.o.g., since the n-ary LCS can be obtained from applying the binary one repeatedly.

3 REPRESENTING $\mathrm{LTL}_{\mathcal{EL}}^{\mathbf{X},\mathbf{G}}$ CONCEPTS

In order to characterize the LCS and the MSC in $\mathrm{LTL}_{\mathcal{EL}}^{\mathbf{X},\mathbf{G}}$, we need a characterization of subsumption and of instance. We extend the approach for $\mathcal{EL}$ from [4], where concepts are normalized and represented by $\mathcal{EL}$-description trees. The subsumption test is then simply deciding the existence of a homomorphism between such trees. We first assume $\mathsf{N}_\mathsf{C}^{\mathsf{rig}} = \mathsf{N}_\mathsf{R}^{\mathsf{rig}} = \emptyset$ and then describe how to handle rigid symbols by a preprocessing step (Section 3.3).

3.1 Normal Form and $\mathrm{LTL}_{\mathcal{EL}}^{\mathbf{X},\mathbf{G}}$ Description Trees

We extend description trees for $\mathcal{EL}$ to $\mathrm{LTL}_{\mathcal{EL}}^{\mathbf{X},\mathbf{G}}$ by accommodating temporal operators $\mathbf{X}$ and $\mathbf{G}$. We treat both temporal operators as special roles and exemplify their effects by a normal form for $\mathrm{LTL}_{\mathcal{EL}}^{\mathbf{X},\mathbf{G}}$ concepts and a propagation phase. W.l.o.g. we assume $\mathbf{X}, \mathbf{G} \notin \mathsf{N}_\mathsf{R}$.

First, observe that each domain element of a temporal interpretation is only directly connected "in the next time point" to itself. This justifies to treat $\mathbf{X}$ like a (partial) functional role which in turn admits restriction to only one $\mathbf{X}$ subconcept per concept. Second, the information from the $\mathbf{G}$ subconcepts for one element need to be propagated to each time point of that element. Since $\mathbf{G}$ has "non-local" temporal effects, we use a role that represents the information that holds at every time point of an element. The $\mathbf{G}$-successor node then represents the concept that needs to be satisfied at all future time points. The following normal form realizes the functional behavior of $\mathbf{X}$ and $\mathbf{G}$ in $\mathrm{LTL}_{\mathcal{EL}}^{\mathbf{X},\mathbf{G}}$ concepts. Note that any domain element is an instance of $\mathbf{X}\,\top$ and $\mathbf{G}\,\top$.

Definition 3.1 (Normal form). Let $A_1, \ldots, A_n \in \mathsf{N}_\mathsf{C}$. An $\mathrm{LTL}_{\mathcal{EL}}^{\mathbf{X},\mathbf{G}}$ concept C is in *normal form* (NF) if it is of the form

$$C = A_1 \sqcap \ldots \sqcap A_n \sqcap \exists r_1.D_1 \sqcap \ldots \sqcap \exists r_m.D_m \sqcap \mathbf{X}\, E \sqcap \mathbf{G}\, F\,,$$

where $D_1, \ldots, D_m, E, F$ are $\mathrm{LTL}_{\mathcal{EL}}^{\mathbf{X},\mathbf{G}}$ concepts in normal form, and F does have neither $\mathbf{X}$ nor $\mathbf{G}$ on the top-level conjunction.

Note, that equivalent concepts can result in different normalized concepts. To see this, consider $C = A \sqcap \mathbf{X}(A \sqcap \mathbf{G}\, A) \sqcap \mathbf{G}\, A$ and $D = A \sqcap \mathbf{G}\, A$. Both are equivalent and each of them is in NF. The following rules transform any $\mathrm{LTL}_{\mathcal{EL}}^{\mathbf{X},\mathbf{G}}$ concept C into NF and also propagate information from $\mathbf{G}$ concepts "along" the time line:

$\mathbf{X}\, C_1 \sqcap \mathbf{X}\, C_2$	$\rightsquigarrow$	$\mathbf{X}(C_1 \sqcap C_2)$	[MergeX]
$\mathbf{G}\, C_1 \sqcap \mathbf{G}\, C_2$	$\rightsquigarrow$	$\mathbf{G}(C_1 \sqcap C_2)$	[MergeG]
$\mathbf{G}(\mathbf{G}\, C_1)$	$\rightsquigarrow$	$\mathbf{G}\, C_1$	[FlattenG]
$\mathbf{G}(\mathbf{X}\, C_1)$	$\rightsquigarrow$	$\mathbf{X}(\mathbf{G}\, C_1)$	[MoveG]
$\mathbf{G}\, C_1$	$\rightsquigarrow$	$C_1 \sqcap \mathbf{G}\, C_1$	[DistributeG]
$\mathbf{X}\, C_1 \sqcap \mathbf{G}\, C_2$	$\rightsquigarrow$	$\mathbf{X}(C_1 \sqcap \mathbf{G}\, C_2) \sqcap \mathbf{G}\, C_2$	[PropagateG]

All rules are applied exhaustively, but MergeX, MergeG, FlattenG and MoveG have higher priority. Applying these four rules exhaustively already results in a concept in NF. DistributeG and PropagateG are applied to the nested subconcepts first and then to the root concept. MergeX and MergeG make $\mathbf{X}$ and $\mathbf{G}$ functional roles by collecting all $\mathbf{X}$ and $\mathbf{G}$ concepts into one subconcept. FlattenG and MoveG ensure that $\mathbf{G}$ concepts do not contain $\mathbf{X}$ and $\mathbf{G}$ directly. DistributeG and PropagateG make the global information explicit at the current and all future time points referred to in the concept.

Normalization might cause an exponential blow-up of a concept due to PropagateG which copies concept C from $\mathbf{G}\, C$ along the chain of $\mathbf{X}$-successors. For a concept C, C^* denotes C in NF.

Proposition 3.2. *Let C be an $\mathrm{LTL}_{\mathcal{EL}}^{\mathbf{X},\mathbf{G}}$ concept. Then, it holds that $C \equiv C^*$. The size of C^* can be exponential in the size of C.*

Proof Sketch. Using well-known equivalences for LTL formulae, it is clear that the rules are equivalence preserving. The exponential blow-up in size comes from the interaction of the rules DistributeG and PropagateG, where concepts are copied. However, since there is no rule that extends a $\mathbf{X}$-chain and PropagateG is only applicable as many times as the length of the longest $\mathbf{X}$-chain in C. This ensures termination of the NF transformation. $\square$

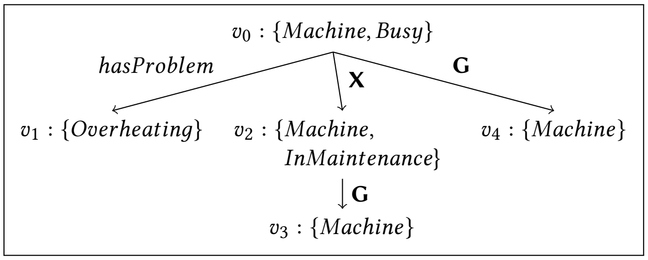

Figure 1: $\mathrm{LTL}_{\mathcal{EL}}^{\mathbf{X},\mathbf{G}}$ description tree.

Concepts in NF can be represented by description trees which have originally been introduced for $\mathcal{EL}$ in [4] and are essentially syntax trees of concepts. We extend these to $\mathrm{LTL}_{\mathcal{EL}}^{\mathbf{X},\mathbf{G}}$ concepts.

Definition 3.3 ($\mathrm{LTL}_{\mathcal{EL}}^{\mathbf{X},\mathbf{G}}$ description tree). An $\mathrm{LTL}_{\mathcal{EL}}^{\mathbf{X},\mathbf{G}}$ *description tree* is of the form $\mathcal{G} = (V, \mathcal{E}, v_0, \ell)$, where $\mathcal{G}$ is a tree with root v_0 and where

- the edges $vrw \in \mathcal{E}$ are labeled with a role name $r \in \mathsf{N_R} \cup \{\mathbf{X}, \mathbf{G}\}$; and
- the nodes $v \in V$ are labeled with sets $\ell(v)$ of concept names from $\mathsf{N_C}$. The empty label corresponds to $\top$.

Any $\mathrm{LTL}_{\mathcal{EL}}^{\mathbf{X},\mathbf{G}}$ concept C can be translated into an $\mathrm{LTL}_{\mathcal{EL}}^{\mathbf{X},\mathbf{G}}$ description tree $\mathcal{G}_C = (V, \mathcal{E}, v_0, \ell)$. Intuitively, concepts of the form $\exists r.C$, $\mathbf{X}\, C$, and $\mathbf{G}\, C$ give rise to successor nodes, while (conjunctions of) concept names build the node labels. The reverse construction, i.e. to read off an $\mathrm{LTL}_{\mathcal{EL}}^{\mathbf{X},\mathbf{G}}$ concept $C_{\mathcal{G}}$ from an $\mathrm{LTL}_{\mathcal{EL}}^{\mathbf{X},\mathbf{G}}$ description tree $\mathcal{G}$, is done in the obvious way. For a description tree $\mathcal{G}_C$ and a node w, we denote the *subtree of* $\mathcal{G}_C$ *with root* w by $\mathcal{G}_C(w)$. An $\mathbf{X}$-path in a description tree is a path, where each edge is labelled with $\mathbf{X}$.

Example 3.4 ($\mathrm{LTL}_{\mathcal{EL}}^{\mathbf{X},\mathbf{G}}$ description tree). The $\mathrm{LTL}_{\mathcal{EL}}^{\mathbf{X},\mathbf{G}}$ concept

$$\begin{aligned} C := {} & Machine \sqcap Busy \sqcap \exists hasProblem.Overheating \sqcap \\ & \mathbf{X}(Machine \sqcap InMaintenance \sqcap \mathbf{G}(Machine)) \sqcap \\ & \mathbf{G}(Machine) \end{aligned}$$

corresponds to the description tree $\mathcal{G}_C$ depicted in Figure 1. Concept C describes that busy machines with the problem of overheating will get maintenance in the next time point.

Lemma 3.5. *Let C be an $\mathrm{LTL}_{\mathcal{EL}}^{\mathbf{X},\mathbf{G}}$ concept in NF and $\mathcal{G}_C = (V, \mathcal{E}, v_0, \ell)$ be the $\mathrm{LTL}_{\mathcal{EL}}^{\mathbf{X},\mathbf{G}}$ description tree of C. Then the following holds:*

1. *$C = C_{\mathcal{G}_C}$ up to commutativity and associativity of conjunction, and $\mathcal{G}_C = \mathcal{G}_{C_{\mathcal{G}_C}}$ up to renaming nodes.*
2. *For each node $v \in V$, v has at most one outgoing edge labeled $\mathbf{X}$ and at most one outgoing edge labeled $\mathbf{G}$.*
3. *Let $v\, \mathbf{G}\, w \in \mathcal{E}$. Then, there does not exists $x \in V$ such that either $w\, \mathbf{X}\, x \in \mathcal{E}$ or $w\, \mathbf{G}\, x \in \mathcal{E}$, i.e., $C_{\mathcal{G}}$ does not have a subconcept of the form $\mathbf{G}\,\mathbf{X}\, D$ or $\mathbf{G}\,\mathbf{G}\, D$.*
4. *Let $v\, \mathbf{G}\, w \in \mathcal{E}$ and D be the $\mathrm{LTL}_{\mathcal{EL}}^{\mathbf{X},\mathbf{G}}$ concept corresponding to the subtree of $\mathcal{G}_C$ with root w. Then for any v' s.t. there is an $\mathbf{X}$-path from v to v', and D' the $\mathrm{LTL}_{\mathcal{EL}}^{\mathbf{X},\mathbf{G}}$ concept corresponding to the subtree of $\mathcal{G}_C$ with root v', $D' \sqsubseteq D$ holds.*

To characterize subsumption, we need to use the connection between description trees and the semantics of $\mathrm{LTL}_{\mathcal{EL}}^{\mathbf{X},\mathbf{G}}$ concepts given by temporal interpretations. Specifically, next we describe how to generate a temporal interpretation from a description tree.

3.2 Canonical Interpretation of $\mathrm{LTL}_{\mathcal{EL}}^{\mathbf{X},\mathbf{G}}$ Concepts

Consider $C_{\mathrm{ex}} = Machine \sqcap \exists isConnected.(Machine \sqcap \mathbf{X}\, Clean) \sqcap (\mathbf{X}(Noisy \sqcap \mathbf{G}\, \exists hasProblem.Overheating))$ as an example. In Figure 2, that shows description tree $\mathcal{G}_{C_{\mathrm{ex}}}$, the nodes can be classified according to their distance from the root v_0 in terms of $\mathbf{X}$-edges, i.e. number of time steps. If a pair of nodes is connected by an $\mathbf{X}$-edge, then they represent the same element in the domain, but at subsequent points in time. In an $\mathrm{LTL}_{\mathcal{EL}}^{\mathbf{X},\mathbf{G}}$ description tree $\mathcal{G}_C = (V, \mathcal{E}, v_0, \ell)$ a node $v \in V$ is called a

- *next copy* if there exists $w\, \mathbf{X}\, v \in \mathcal{E}$,
- *global copy*, if $w\, \mathbf{G}\, v \in \mathcal{E}$,
- *temporal copy* iff v is either a next copy or a global copy, and
- *temporal root* iff v is not a temporal copy.

The *temporal depth* of v ($\mathrm{td}(v)$) is the number of $\mathbf{X}$-edges that occur in the path from v_0 to v. We use the following sets: V_X for next copies, V_G for global copies, and V_R for temporal roots, For a temporal copy $v \in V_G$, $w \in V_R$ is the *temporal root of* v ($w = \mathrm{tr}(v)$) if $w\, \mathbf{G}\, v \in \mathcal{E}$ or there is an $\mathbf{X}$-path from w to v in $\mathcal{G}_C$. For a temporal root, its temporal copies are the same object in the domain, but at later time points. Note, that a temporal root is its own temporal root. Next, we define the canonical temporal interpretation over the temporal roots as its domain gained from a description tree.

Definition 3.6 (Canonical Interpretation). Let C be an $\mathrm{LTL}_{\mathcal{EL}}^{\mathbf{X},\mathbf{G}}$ concept in NF and $\mathcal{G}_C = (V, \mathcal{E}, v_0, \ell)$ the $\mathrm{LTL}_{\mathcal{EL}}^{\mathbf{X},\mathbf{G}}$ description tree of C. The *canonical interpretation of* C is the temporal interpretation $\mathfrak{I}_C = ((\mathcal{I}_C)_i)_{0 \leq i}$, with $(\mathcal{I}_C)_i = (V_R, \cdot^{(\mathcal{I}_C)_i})$ for each $i \geq 0$, where

- for each $A \in \mathsf{N_C}$ and $i \geq 0$
  $A^{(\mathcal{I}_C)_i} := \{v \in V_R \mid A \in \ell(v) \text{ and } \mathrm{td}(v) = i\} \cup$
  $\{v \in V_R \mid \exists w \in V_X : v = \mathrm{tr}(w), A \in \ell(w) \text{ and } \mathrm{td}(w) = i\} \cup$
  $\{v \in V_R \mid \exists w \in V_G : v = \mathrm{tr}(w), A \in \ell(w) \text{ and } \mathrm{td}(w) \leq i\}$
- for $r \in \mathsf{N_R}$ and $i \geq 0$
  $r^{(\mathcal{I}_C)_i} := \{(v, w) \mid v \in V_R, vrw \in \mathcal{E} \text{ and } \mathrm{td}(v) = i\} \cup$
  $\{(v, w) \mid v = \mathrm{tr}(x) \text{ and } w = \mathrm{tr}(y) \text{ where}$
  $v, w \in V_X, xry \in \mathcal{E} \text{ and } \mathrm{td}(x) = \mathrm{td}(y) = i\} \cup$
  $\{(v, w) \mid v = \mathrm{tr}(x), x \in V_G, xrw \in \mathcal{E} \text{ and } \mathrm{td}(x) \leq i\}$.

Lemma 3.7. *Let C be a $\mathrm{LTL}_{\mathcal{EL}}^{\mathbf{X},\mathbf{G}}$ concept, v_0 the root of $\mathcal{G}_C$, and $\mathfrak{I}_C = ((\mathcal{I}_C)_i)_{0 \leq i}$ its canonical interpretation. Then $v_0 \in C^{\mathfrak{I}_C}$ holds.*

Proof Sketch. We proceed by induction on the depth of $\mathcal{G}_C$. The proof for the base case and the case of r-successors (for $r \in \mathsf{N_R}$) are similar, since they concern temporal roots. Intuitively, we show that each concept name in top conjunction is satisfied since they are contained in $\ell(v_0)$. For each $\exists r.D$, we show that there exists w such that $(v, w) \in r^{(\mathcal{I}_C)_i}$ and $w \in D^{(\mathcal{I}_C)_i}$.

For the case of $\mathbf{X}\, E$, we show there exists a domain element $w \in V_R$ s.t. $\mathrm{tr}(w) = v_0$ and $\mathrm{td}(w) = \mathrm{td}(v_0) + 1$. Furthermore, $w \in (E_{\mathcal{G}_C(w)})^{\mathfrak{I}_C}$ implies $v_0 \in (\mathbf{X}\, E_{\mathcal{G}_C(w)})^{\mathfrak{I}_C}$. For the case of $\mathbf{G}\, F$, we show there exists $w \in V_R$ s.t. $\mathrm{tr}(w) = v_0$ and $\mathrm{td}(w) = \mathrm{td}(v_0)$ and thus w is a $\mathbf{G}$-successor of v_0. Then, we use the existence of

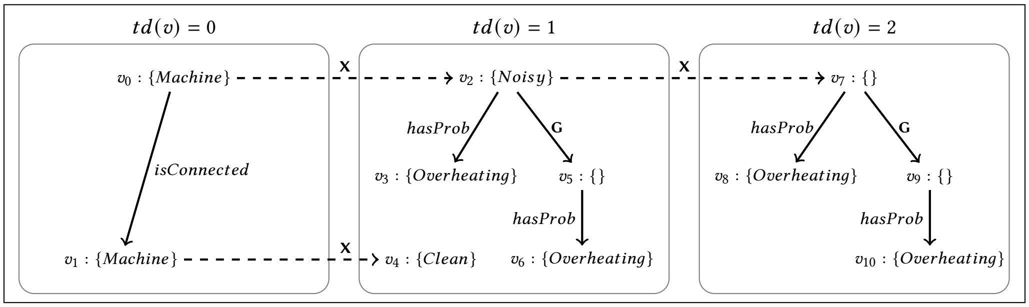

Figure 2: $\text{LTL}^{\mathbf{X},\mathbf{G}}_{\mathcal{EL}}$ description tree for $C_{\mathbf{ex}}$ in NF and with annotations for temporal depths.

$w \in (F_{\mathcal{G}_C(w)})^{(\mathcal{I}_C)_i}$ which propagates that $v_0 \in (F_{\mathcal{G}_C(w)})^{(\mathcal{I}_C)_j}$ for all time points $j \geq i$ due to the construction of $\mathfrak{I}_C$ where each temporal copy of v_0 is connected to w for all time points and $w \in F^{\mathfrak{I}_C}_{\mathcal{G}_C(w)}$. Thus, we have that $v_0 \in (\mathbf{G}\, F_{\mathcal{G}_C(w)})^{\mathfrak{I}_C}$ □

3.3 Admitting Rigid Names

Note that $\mathbf{G}\, A$ differs from a rigid concept A semantically, because $\mathbf{G}\, A$ is forward-looking only. Since $\text{LTL}^{\mathbf{X},\mathbf{G}}_{\mathcal{EL}}$ has no past temporal operators, a rigid concept A, can be simulated by using the concept $\mathbf{G}\, A$ at the initial time point.

Definition 3.8 ($\mathsf{N}^{\text{rig}}_{\mathsf{C}}$-induced concept, $\mathsf{N}^{\text{rig}}_{\mathsf{R}}$-induced concept). Let $C = A_1 \sqcap \ldots \sqcap A_n \sqcap \exists r_1.D_1 \sqcap \ldots \sqcap \exists r_m.D_m \sqcap \mathbf{X}\, E \sqcap \mathbf{G}\, F$ be an arbitrary $\text{LTL}^{\mathbf{X},\mathbf{G}}_{\mathcal{EL}}$ concept and for all $1 \leq i \leq k$ with $k \leq n$ let $A_i \in \mathsf{N}^{\text{rig}}_{\mathsf{C}}$ and let there exist $\mathbf{X}^{n_i}\, A_i$ ($n_i \in \mathbb{N}$) in the top-level conjunction of C. We define the $\mathsf{N}^{\text{rig}}_{\mathsf{C}}$*-induced concept* of C (denoted $\mathsf{rigC}(C)$) recursively:

$$\mathsf{rigC}(C) := A_1 \sqcap \ldots \sqcap A_n \sqcap \exists r_1.\mathsf{rigC}(D_1) \sqcap \ldots \sqcap \exists r_m.\mathsf{rigC}(D_m) \sqcap \mathbf{X}\, E \sqcap \mathbf{G}\, F \sqcap \mathbf{G}(A_1 \sqcap \ldots \sqcap A_k).$$

For all $1 \leq i \leq k$ with $k \leq m$, let $r_i \in \mathsf{N}^{\text{rig}}_{\mathsf{R}}$ and let there exist $\mathbf{X}^{m_i}(\exists r_i.\mathsf{rigR}(D_i))$ at the top-level conjunction of C; and F_i is the global operand ($\mathbf{G}\, F_i$) at the top-level conjunction of $\mathsf{rigR}(D_i)$. We define the $\mathsf{N}^{\text{rig}}_{\mathsf{R}}$*-induced concept* of C (denoted $\mathsf{rigR}(C)$) recursively:

$$\mathsf{rigR}(C) := A_1 \sqcap \ldots \sqcap A_n \sqcap \exists r_1.\mathsf{rigR}(D_1) \sqcap \ldots \sqcap \exists r_m.\mathsf{rigR}(D_m) \sqcap \mathbf{X}\, E \sqcap \mathbf{G}\, F \sqcap \mathbf{G}(\exists r_1.F_1 \sqcap \ldots \sqcap \exists r_k.F_k)$$

Notice that we assume $\mathsf{rigC}(C)$ and $\mathsf{rigR}(C)$ are transformed into normal form afterwards.

Proposition 3.9. *Let C be an $\text{LTL}^{\mathbf{X},\mathbf{G}}_{\mathcal{EL}}$ concept and $\mathfrak{I}$ a temporal interpretation.*

(1) *$\mathfrak{I}$ is a model of C w.r.t. $\mathsf{N}^{\text{rig}}_{\mathsf{C}}$ iff $\mathfrak{I}$ is a model of* $\mathsf{rigC}(C)$
(2) *$\mathfrak{I}$ is a model of C w.r.t. $\mathsf{N}^{\text{rig}}_{\mathsf{R}}$ if $\mathfrak{I}$ is a model of* $\mathsf{rigR}(C)$

While $\mathsf{rigC}(C)$ incorporates the semantics of rigid concept names, this is not the case for $\mathsf{rigR}(C)$. Proposition 3.9 shows that the set of models of C w.r.t. rigid concept names and $\mathsf{rigC}(C)$ coincide. Thus, we do not need to distinguish between rigid and flexible concept names. However, rigid concept names can have a negative effect on the computation, since a single rigid concept name may generate a $\mathbf{G}$ conjunct which can cause an exponential blow-up during normalization.

The set of models of C w.r.t. rigid concept names is contained in those for $\mathsf{rigR}(C)$, but the sets do not coincide. However, we show later that this extension is sufficient to characterize subsumption and LCS in $\text{LTL}^{\mathbf{X},\mathbf{G}}_{\mathcal{EL}}$.

4 CHARACTERIZATION OF SUBSUMPTION AND THE LCS

In order to decide subsumption, it needs to be fixed, up to which depth to consider the concepts in the $\mathbf{X}$-chains. Clearly, such chains can get arbitrarily long, if repeatedly $\mathbf{G}$ concepts are propagated onto a $\mathbf{X}$ concept. For a subsumption $C_1 \sqsubseteq C_2$ to be decided, $\mathcal{G}_{C_1}$ needs to have a temporal depth greater or equal to the one of $\mathcal{G}_{C_2}$, in order to be able to employ homomorphisms for the comparison. If concept C_1 describes "less time points" than C_2, then some padding is needed to achieve the same temporal depth.

Definition 4.1 ($\text{LTL}^{\mathbf{X},\mathbf{G}}_{\mathcal{EL}}$ concept padding). Let C_1 and C_2 be $\text{LTL}^{\mathbf{X},\mathbf{G}}_{\mathcal{EL}}$ concepts in normal form. A function to pad C_1 w.r.t. C_2 (denoted by $pad_{C_2}(C_1)$) proceeds as follows:

- for each existential restriction $\exists r.D_1$ in the top-level of C_1, replace $\exists r.D_1$ with $\exists r.pad_{D_j}(D_1)$ recursively for all $\exists r.D_j$ occurring in the top-level of C_2;
- if there exists $\mathbf{X}\, E_2$ in the top-level conjunction of C_2, then
  - if there exists $\mathbf{X}\, E_1$ in the top-level of C_1, then replace it with $\mathbf{X}\, pad_{E_2}(E_1)$;
  - otherwise:
    - if there exists $\mathbf{G}\, F$ in the top-level of C_1, replace C_1 with $C_1 \sqcap \mathbf{X}(pad_{E_2}(F \sqcap \mathbf{G}\, F))$; and
    - otherwise, replace C_1 with $C_1 \sqcap \mathbf{X}(pad_{E_2}(\top))$.

Furthermore, we say that C_1 is *aligned* w.r.t. C_2 if $pad_{C_2}(C_1) = C_1$.

Observe that the use of global concepts ensures that rigid concepts are used for the padding. The padding function preserves equivalence of $\text{LTL}^{\mathbf{X},\mathbf{G}}_{\mathcal{EL}}$ concepts. Now, using the notion of aligned concepts, we can ensure that the description tree for C_1 is deep enough in the temporal dimension to be compared with the one for C_2. We can thus use homomorphisms between two description trees to characterize subsumption.

Definition 4.2 (Homomorphism between $\text{LTL}^{\mathbf{X},\mathbf{G}}_{\mathcal{EL}}$ description trees). Let $\mathcal{H} = (V_H, \mathcal{E}_H, w_0, \ell_H)$ and $\mathcal{G} = (V_G, \mathcal{E}_G, v_0, \ell_G)$ be $\text{LTL}^{\mathbf{X},\mathbf{G}}_{\mathcal{EL}}$ description trees. A homomorphism from $\mathcal{H}$ into $\mathcal{G}$ is a mapping $\varphi : V_H \mapsto V_G$ where

(1) $\varphi(w_0) = v_0$;
(2) $\ell_H(v) \subseteq \ell_G(\varphi(v))$ for all $v \in V_H$; and
(3) $\varphi(v)r\varphi(w) \in \mathcal{E}_G$ for all $vrw \in \mathcal{E}_H$.

Theorem 4.3 (Characterization of subsumption). *Let C, D be $\mathrm{LTL}^{\mathbf{X},\mathbf{G}}_{\mathcal{EL}}$ concepts in NF and C is aligned w.r.t. D. Then, we have that $C \sqsubseteq D$ iff there exists a homomorphism from $\mathcal{G}_D$ to $\mathcal{G}_C$.*

Proof Sketch. "$\Leftarrow$": we prove $v_0 \in D^{\mathfrak{I}}$ by induction on $|V_D|$, i.e. the number of nodes in $\mathcal{G}_D$. In the induction step, we prove that if $a \in C^{\mathfrak{I}}$, i.e., a is a witness of each top-level conjunct in C, then it has an appropriate successor for each top-level conjunct in D. We separate cases depending on the type of conjunct and utilize the existence of a homomorphism to show, necessary successors exists for each element in $\mathsf{N_R} \cup \{\mathbf{X}, \mathbf{G}\}$.

"$\Rightarrow$": is shown by induction on depth(D) by constructing an appropriate homomorphism on-the-fly. For the base case this is straightforward, since $\ell_D \subseteq \ell_C$. In the induction step, we show for each $w_0rw \in \mathcal{E}_D$, that there exists an appropriate v_0rv for each $r \in \mathsf{N_R} \cup \{\mathbf{X}, \mathbf{G}\}$ and $C_{\mathcal{G}_{C(v)}} \sqsubseteq C_{\mathcal{G}_{D(w)}}$. Then, we map w_0 to v_0 in the construction of the homomorphism inductively. □

We use this characterization of subsumption to show correctness of our LCS construction. The LCS concept is the concept corresponding to the product of description trees of the rigidity-induced, normalized and aligned input concepts.

Definition 4.4 (Product of $\mathrm{LTL}^{\mathbf{X},\mathbf{G}}_{\mathcal{EL}}$ description trees). Let $\mathcal{G} = (V_G, \mathcal{E}_G, v_0, \ell_G)$ and $\mathcal{H} = (V_H, \mathcal{E}_H, w_0, \ell_H)$ be $\mathrm{LTL}^{\mathbf{X},\mathbf{G}}_{\mathcal{EL}}$ description trees. The *product of $\mathcal{G}$ and $\mathcal{H}$* is $\mathcal{G} \times \mathcal{H} := (V, \mathcal{E}, (v_0, w_0), \ell)$ with

- (v_0, w_0) is the root of $\mathcal{G} \times \mathcal{H}$, labeled with $\ell_G(v_0) \cap \ell_H(v_0)$,
- for each r-successor ($r \in \mathsf{N_R} \cup \{\mathbf{X}, \mathbf{G}\}$) v of v_0 in $\mathcal{G}$ and w of w_0 in $\mathcal{H}$, there is an r-successor (v, w) of (v_0, w_0) in $\mathcal{G} \times \mathcal{H}$ that is the root of $\mathcal{G}(v) \times \mathcal{H}(w)$.

Theorem 4.5. *Let C_1, C_2 be $\mathrm{LTL}^{\mathbf{X},\mathbf{G}}_{\mathcal{EL}}$ concepts in normal form and C_1 is aligned w.r.t. C_2 and C_2 is aligned w.r.t. C_1. Then, $C_{\mathcal{G}_{C_1} \times \mathcal{G}_{C_2}}$ is the LCS of C_1 and C_2.*

Proof Sketch. We show the following: (1) $C_1 \sqsubseteq C_{\mathcal{G}_{C_1} \times \mathcal{G}_{C_2}}$, (2) $C_2 \sqsubseteq C_{\mathcal{G}_{C_1} \times \mathcal{G}_{C_2}}$, and (3) for each $\mathrm{LTL}^{\mathbf{X},\mathbf{G}}_{\mathcal{EL}}$ concept E with $C_1 \sqsubseteq E$ and $C_2 \sqsubseteq E$, we have that $C_{\mathcal{G}_{C_1} \times \mathcal{G}_{C_2}} \sqsubseteq E$. The statements (1) and (2) can be proven by showing that the product of description trees captures, by construction, properties of both C_1 and C_2. Due to this, there exist the required homomorphisms which in turn shows that both subsumption relationships hold.

Since E is a common subsumer of C_1 and C_2, there exists homomorphisms φ_1 from $\mathcal{G}_E$ to $\mathcal{G}_{C_1}$ and φ_2 from $\mathcal{G}_E$ to $\mathcal{G}_{C_2}$. Then, (3) can be shown by defining a mapping $\varphi := \langle \varphi_1, \varphi_2 \rangle$ from $\mathcal{G}_E$ to $\mathcal{G}_{C_1} \times \mathcal{G}_{C_2}$ as the product of φ_1 and φ_2, i.e., $\varphi(v') := (\varphi_1(v'), \varphi_2(v'))$ for all $v' \in V_E$ of $\mathcal{G}_E$. Then, φ is well-defined, i.e., $\varphi(v') \in V_{\mathcal{G}_{C_1} \times \mathcal{G}_{C_2}}$ for all $v' \in V_E$ by induction. Finally, we show φ is a homomorphism from $\mathcal{G}_E$ to $\mathcal{G}_{C_1} \times \mathcal{G}_{C_2}$ due to the construction of the product of $\mathrm{LTL}^{\mathbf{X},\mathbf{G}}_{\mathcal{EL}}$ description trees. □

4.1 Subsumption and LCS with Rigid Names

Since rigid concepts can be expressed by $\mathbf{G}$ directly, we focus on rigid roles and extend the result from Theorem 4.3 to these roles.

Theorem 4.6. *Let C, D be $\mathrm{LTL}^{\mathbf{X},\mathbf{G}}_{\mathcal{EL}}$ concepts. Then we have :*

(1) *if* $\mathsf{rigR}(C)$ *is aligned w.r.t.* $\mathsf{rigR}(D)$*, then it holds that* $\mathsf{rigR}(C) \sqsubseteq \mathsf{rigR}(D)$ *w.r.t. rigid names iff there exists a homomorphism from* $\mathcal{G}_{\mathsf{rigR}(D)}$ *to* $\mathcal{G}_{\mathsf{rigR}(C)}$.
(2) *if* $\mathsf{rigR}(C)$ *is aligned w.r.t.* $\mathsf{rigR}(D)$*, then we have that* $C_{\mathcal{G}_{\mathsf{rigR}(C) \times \mathsf{rigR}(D)}}$ *is the LCS of* C_1 *and* C_2 *w.r.t. rigid role names.*

Proof Sketch. The proof of (1) is an extension of the proof of Theorem 4.3. Notice that the models of C w.r.t. rigid role names do not coincide with the models of $\mathsf{rigC}(C)$. Consider the case of $\exists r.A$ and a is an instance of $\exists r.A$. If r is rigid, $\exists r.A$ would be extended by rigR () with $\mathbf{G}\,\exists r.\top$ at the top-level conjunction. Observe $\mathbf{G}\,\exists r.\top$ does not restrict a to be connected with the same individual at each time point. However, this is not problematic, since the concept constructors in $\mathrm{LTL}^{\mathbf{X},\mathbf{G}}_{\mathcal{EL}}$ cannot distinguish those two cases and neither can the homomorphism.

Since (1) holds, we can prove (2) in a similar manner as we did in Theorem 4.5. Since the characterization of LCS make use of the characterization of subsumption, the proof is almost identical. □

While the computation is similar with the case without rigid names, the concepts are larger due to construction of $\mathsf{rigR}(C)$. Since a rigid role needs to be induced to $\mathbf{G}$ successor and propagated, it increases the complexity exponentially.

5 CHARACTERIZATION OF THE INSTANCE RELATIONSHIP AND MSC

Next, we develop a method for computing a $\mathrm{LTL}^{\mathbf{X},\mathbf{G}}_{\mathcal{EL}}$ concept that describes an individual from a sequence of ABoxes best, i.e. a computation method for the MSC in $\mathrm{LTL}^{\mathbf{X},\mathbf{G}}_{\mathcal{EL}}$. To show correctness of the method, we develop a characterization of the instance relationship in $\mathrm{LTL}^{\mathbf{X},\mathbf{G}}_{\mathcal{EL}}$. The general approach used here follows [7], but extends it to temporal operators and sequences of ABoxes as the input.

5.1 Characterization of the Instance Relationship

Since there are n ABoxes in $\vec{\mathcal{A}}_n$, we have to consider information from each time point on the input individual a. We can use the $\mathbf{X}$ operator to describe the temporal information in one concept. Intuitively, we construct a concept C that represents a from (a user-selected) time point i on. Let $Ind(\vec{\mathcal{A}}_n)$ denote the set of all individuals occurring in $\vec{\mathcal{A}}_n$. For each $a \in Ind(\vec{\mathcal{A}}_n)$, we define

$$C_a := \bigsqcap_{0 \le i \le n} \big(\bigsqcap_{D(a) \in \mathcal{A}_i} \mathbf{X}^i\, D \big),$$

if there exists an assertion $D(a) \in \mathcal{A}_i$ for any i; and $\top$ otherwise. We assume that each C_a for all a is rigidity induced and normalized.

As relational structures in the ABoxes can be arbitrary, we need to represent the information on a by graphs instead of trees [7] for $\mathcal{EL}$. An $\mathrm{LTL}^{\mathbf{X},\mathbf{G}}_{\mathcal{EL}}$ *description graph* is a labeled graph of the form $\mathcal{G} = (V, \mathcal{E}, \ell)$ whose edges $vrw \in \mathcal{E}$ are labeled with role names $r \in \mathsf{N_R} \cup \{\mathbf{X}, \mathbf{G}\}$ and whose nodes $v \in V$ are labeled with sets $\ell(v) \subseteq \mathsf{N_C}$. Let $\mathcal{G}_{C_a} = (V_a, \mathcal{E}_a, a, \ell_a)$ denote the $\mathrm{LTL}^{\mathbf{X},\mathbf{G}}_{\mathcal{EL}}$ description tree for C_a. W.l.o.g. assume that the sets V_a for all $a \in Ind(\vec{\mathcal{A}}_n)$ are pairwise disjoint. Given an individual $a \in Ind(\vec{\mathcal{A}}_n)$, we define the *temporal copy of a at time point i* as $\mathsf{tc}_i(a) = v$, where $v \in V_a$ such that there is an **X**-path of length i from a to v in $\mathcal{G}_{C_a}$. The *description graph of a sequence of ABoxes* is $\mathcal{G}(\vec{\mathcal{A}}_n) = (V, \mathcal{E}, \ell)$, with:

- $V = \bigcup_{a \in \mathsf{Ind}(\vec{\mathcal{A}}_n)} V_a$;
- $\mathcal{E} = \{xry \mid r(a,b) \in \mathcal{A}_i, x = \mathrm{tc}_i(a) \text{ and } y = \mathrm{tc}_i(b)\} \cup \bigcup_{a \in Ind(\vec{\mathcal{A}}_n)} \mathcal{E}_a$; and
- $\ell(v) = \ell_a(v)$ for all $v \in V_a$.

Example 5.1 (Description graph). Let $\vec{\mathcal{A}}_2 = (\mathcal{A}_i)_{0 \leq i \leq 2}$ be the following sequence of ABoxes:

$$\mathcal{A}_0 = \{Machine(a)\}$$

$$\mathcal{A}_1 = \{(Noisy \sqcap \mathbf{G}(\exists hasProblem.Overheating))(a),\ Machine(b)\}$$

$$\mathcal{A}_2 = \{Noisy(a),\ Noisy(b),\ isConnected(a,b)\}$$

The description graph $\mathcal{G}(\vec{\mathcal{A}}_2)$ is depicted in Figure 3. Although b does not occur in $\mathcal{A}_0$, $\mathcal{G}(\vec{\mathcal{A}}_2)$ contains b at time point 0.

For a sequence of interpretations, one might be interested in concept membership from a certain time point on, not only in those that hold from the beginning. We characterize the instance relationship using $\mathcal{G}(\vec{\mathcal{A}}_n)$ and $\mathcal{G}_C$. To put starting point i into consideration, we place the temporal copy $\mathrm{tc}_i(a)$ as the temporal root instead of a.

Theorem 5.2 (Characterization of instance). *Let $\vec{\mathcal{A}}_n$ be a sequence of ABoxes, $a \in Ind(\vec{\mathcal{A}}_n)$, C be a $\mathrm{LTL}^{\mathbf{X},\mathbf{G}}_{\mathcal{EL}}$ concept, and $0 \leq i$. Then, $\vec{\mathcal{A}}_n, i \models C(a)$ if there exists a homomorphism φ from $\mathcal{G}_C$ into $\mathcal{G}(\vec{\mathcal{A}}_n)$ such that $\varphi(v_0) = tc_i(a)$, where v_0 is the root of $\mathcal{G}_C$.*

Proof Sketch. We show by induction that the concept represented by subtree with root $\mathrm{tc}_i(a)$ together with existence of homomorphism yield $\mathrm{tc}_i(a) \in C^{\mathcal{I}_i}$ of the canonical interpretation. Let $C_{\mathrm{tc}_i(a)} := \bigsqcap_{0 \leq i \leq n} (\bigsqcap_{D(\mathrm{tc}_i(a) \in \mathcal{A}_i} \mathbf{X}^i D)$, then $\mathfrak{I} \models \vec{\mathcal{A}}_n$ implies that $a \in C^{\mathfrak{I}}_{\mathrm{tc}_i(a)}$ due to the construction of $\mathcal{G}(\vec{\mathcal{A}}_n)$. Then, we show $a^{\mathfrak{I}} \in C^{\mathfrak{I}}$ by induction on the depth(C) by utilizing the existence of homomorphism. □

5.2 Computing the k-MSC in $\mathrm{LTL}^{\mathbf{X},\mathbf{G}}_{\mathcal{EL}}$

The *role depth* of C (denoted rdepth(C)) is the maximum number of nested quantifiers in C. The MSC in $\mathrm{LTL}^{\mathbf{X},\mathbf{G}}_{\mathcal{EL}}$ suffers the same problem as the one in $\mathcal{EL}$: cycles in the description graph would cause infinite role depth in the MSC [7] and thus the MSC need not exist since concepts are finite descriptions. A common approach is to approximate the MSC by the k-MSC, i.e. to limit the role depth of the MSC concept to $k \in \mathbb{N}$.

Definition 5.3 (k-MSC). Let $\vec{\mathcal{A}}_n$ be a sequence of ABoxes, $a \in Ind(\vec{\mathcal{A}}_n)$, C an $\mathrm{LTL}^{\mathbf{X},\mathbf{G}}_{\mathcal{EL}}$ concept, and $i, k \geq 0$. Then, C is the *k-MSC of a* w.r.t. $\vec{\mathcal{A}}_n$ and time point i (k-$\mathbf{msc}_i(a)$) iff

- $\vec{\mathcal{A}}_n, i \models C(a)$;
- rdepth$(C) \leq k$; and
- $C \sqsubseteq C'$ for all $\mathrm{LTL}^{\mathbf{X},\mathbf{G}}_{\mathcal{EL}}$ concepts C' with $\vec{\mathcal{A}}_n, i \models C'(a)$ and rdepth$(C') \leq k$.

The computation of the k-MSC performs the following steps. First, it performs a tree unraveling of $\mathcal{G}(\vec{\mathcal{A}}_n)$ with root a to obtain a $\mathrm{LTL}^{\mathbf{X},\mathbf{G}}_{\mathcal{EL}}$ description tree $\mathcal{T}(a, \mathcal{G}(\vec{\mathcal{A}}_n))$. The starting point is $\mathrm{tc}_i(a)$ as the root instead of a. Second, it prunes all paths to length k to obtain $\mathcal{T}_k(a, \mathcal{G}(\vec{\mathcal{A}}_n))$. We give conditions for the existence of the MSC in following theorem. Note that this condition is necessary, but in may not be sufficient— similar to the condition for the MSC in $\mathcal{ALE}$ given in [7].

Theorem 5.4. *Let $\vec{\mathcal{A}}_n$ be a sequence of ABoxes, $a \in Ind(\vec{\mathcal{A}}_n)$, let $i, k \geq 0$, and $T = \mathcal{T}_k(tc_i(a), \mathcal{G}(\vec{\mathcal{A}}_n))$. Then*

1. *C_T is the k-MSC of a w.r.t. $\vec{\mathcal{A}}_n$ at time point i.*
2. *If no cyclic path is reachable from $tc_i(a)$ in $\mathcal{G}(\vec{\mathcal{A}}_n)$, then C_T is the MSC of individual a w.r.t. $\vec{\mathcal{A}}_n$ at time i.*

Proof Sketch. Proving (1) is mainly an extension of Lemma 5.2. We can map $\mathrm{tc}_i(a)$ of $\mathcal{G}_{C_T}$ to $\mathrm{tc}_i(a)$ of $\mathcal{G}_{\vec{\mathcal{A}}_n}$ to obtain a homomorphism, since it $\mathcal{G}_{\vec{\mathcal{A}}_n}$ contains $\mathcal{G}_{C_T}$ with root $\mathrm{tc}_i(a)$. This yields that a is an instance of C_T. Then, if C is a $\mathrm{LTL}^{\mathbf{X},\mathbf{G}}_{\mathcal{EL}}$ concept such that a is an instance of C with depth$(C) \leq k$, we can construct a homomorphism from $\mathcal{G}_C$ to $\mathcal{G}_{C_T}$. Then, we have that $C_T \sqsubseteq C$ and finally C_T is the k-MSC.

To prove (2), consider the case where the depth of the unraveled tree is finite, e.g., k'. Then, the depth of the k-MSC is bounded by $k \geq k'$, i.e., all concept with a larger role depth (k'-MSC) are equivalent to the k-MSC. Now consider the case where the unraveled tree has infinite depth. Assume that there exists k-MSC that also serves as MSC of a and call it C_k. Then, it is easy to see that there always exists $(k+1)$-MSC such that $C_{k+1} \sqsubseteq C_k$ due to the cycle. Then, we have to construct an infinitely large k-MSC of C for a. Since a concept description only has a fixed and finite depth, a cannot have an MSC. □

5.3 Instance and k-MSC w.r.t. Rigid Names

The admission of rigid roles in the computation of k-MSC is not as straightforward in as the case of subsumption and LCS. In ABoxes with role assertions for rigid roles, these assertions affect the MSC. In contrast to the LCS with rigid symbols, where r always connects to anonymous individuals, the MSC needs to use information on other ABox individuals (possibly) at different time points.

Consider a sequence of ABoxes $\vec{\mathcal{A}}_n$. Let $\Sigma_{\mathsf{C}^{\mathrm{rig}}}(\vec{\mathcal{A}}_n) \subseteq \mathsf{N}^{\mathrm{rig}}_{\mathsf{C}}$ and $\Sigma_{\mathsf{R}^{\mathrm{rig}}}(\vec{\mathcal{A}}_n) \subseteq \mathsf{N}^{\mathrm{rig}}_{\mathsf{R}}$ denote the sets of rigid concept names and role names that occur in $\vec{\mathcal{A}}_n$, respectively. We collect the rigid assertions in the rigid ABox (written $\mathcal{A}_{\mathrm{rig}}(\vec{\mathcal{A}}_n)$), which is defined as follows:

$$\begin{aligned}\mathcal{A}_{\mathrm{rig}}(\vec{\mathcal{A}}_n) = \{&A(a) \mid \exists i, 0 \leq i \leq n \text{ s.t. } A(a) \in \mathcal{A}_i \text{ and } A \in \Sigma_{\mathsf{C}^{\mathrm{rig}}}(\vec{\mathcal{A}}_n)\} \cup \\ \{&r(a,b) \mid \exists i, 0 \leq i \leq n \text{ s.t. } r(a,b) \in \mathcal{A}_i \text{ and } r \in \Sigma_{\mathsf{R}^{\mathrm{rig}}}(\vec{\mathcal{A}}_n)\}\end{aligned}$$

All assertions in the rigid ABox hold at all time points and beyond the observed time. Clearly, a temporal interpretation $\mathfrak{I}$ respects rigid concepts and roles if $\mathfrak{I}, i \models \alpha$ for all $\alpha \in \mathcal{A}_{rig}(\vec{\mathcal{A}}_n)$ and $0 \leq i$. If $\vec{\mathcal{A}}_n$ is clear from the context, we simply write $\mathcal{A}_{rig}$.

It is straightforward that we can simply augment each ABox $\mathcal{A}_i$ in the sequence with the rigid information collected in the ABox $\mathcal{A}_{\mathrm{rig}}$ from all of the ABoxes in the sequence, i.e., to use

$$\vec{\mathcal{A}}_n{}^{\mathrm{rig}} = (\mathcal{A}_i \cup \mathcal{A}_{\mathrm{rig}})_{0 \leq i \leq n}$$

to propagate the rigid information to each time point. However, this is not enough, since rigid information holds even beyond the

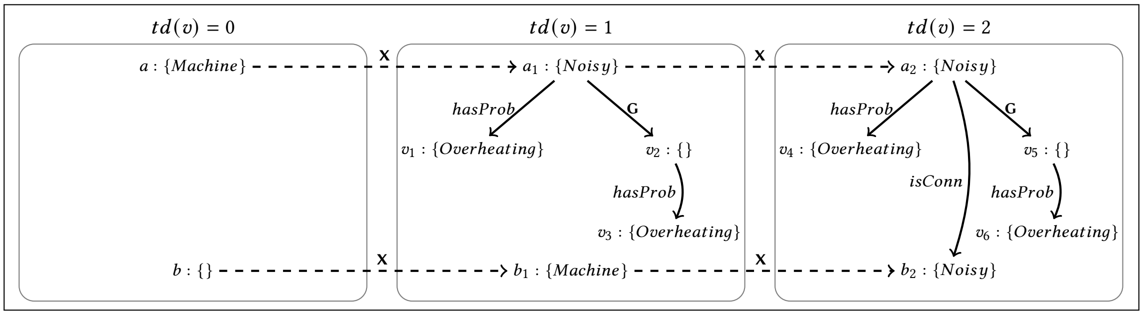

Figure 3: $\text{LTL}_{\mathcal{EL}}^{\mathbf{X},\mathbf{G}}$ description graph for $\vec{\mathcal{A}}_2$ from Example 5.1.

observed time points. This needs to be realized in the MSC in $\text{LTL}_{\mathcal{EL}}^{\mathbf{X},\mathbf{G}}$ by the use of **G** concepts.

Consider the example $\mathcal{A}_0 = \{r(a,b), r(b,a)\}$ and $\mathcal{A}_1 = \{C(b)\}$ with r and C being rigid, then 1-$MSC_0(a) = \mathbf{G}\,\exists r.C \sqcap \exists r.(\mathbf{G}\,C)$.

To compute the k-MSC of a at time point i, we employ a recursive algorithm that proceeds as follows. First, it constructs $\vec{\mathcal{A}}_n{}^{\text{rig}}$ as defined. Then, it performs a modified tree unraveling when traversing the tree. Suppose, the current node is a, at time point i and $r(a,b)$ is contained in the ABox. In $\mathcal{EL}$, one can compute this kind of conjunct recursively by computing $\exists r.((k-1)\text{-}MSC(b))$. For $\text{LTL}_{\mathcal{EL}}^{\mathbf{X},\mathbf{G}}$, however, we have to construct

$$\exists r.((k-1)\text{-}MSC_i(b)_{\Sigma_{\mathsf{R}^{\text{rig}}}(\vec{\mathcal{A}}_n)})$$

instead. Furthermore, we have to extend the MSC with a conjunct $\mathbf{G}\,\exists r.F$, where F is the operand of **G** at the top-level conjunct of $\exists r.(((k-1)\text{-}MSC_i(b)_{\Sigma_{\mathsf{R}^{\text{rig}}}(\vec{\mathcal{A}}_n)})$ and this has to be computed recursively with base $k = 0$.

Note, that the rigid information is propagated "beyond" the observed time points by the **G** branch for each individual at each time point. Although a rigid concept name A is already represented by $\mathbf{G}\,A$ at the initial time point, the unraveling process can be extended to accommodate this. When we traverse an **X** chain and find a rigid concept, we go back to the temporal root and attach $\mathbf{G}\,A$ there.

6 CONCLUSIONS

For ontology-based application that use temporal DLs, the learning of temporal concepts is an important task. So far learning concepts written in a temporal DL has not been addressed in the literature. In this paper, we have devised a method how to derive a temporal concept from example individuals occurring in the sequence of ABoxes. Specifically, we investigated $\text{LTL}_{\mathcal{EL}}^{\mathbf{X},\mathbf{G}}$ concepts, which extend $\mathcal{EL}$ with temporal operators next (**X**) and global (**G**) and have devised computation methods for the LCS and the (k-)MSC in this logic. We have investigated the settings with rigid concept and roles, and can accommodate them by a preprocessing step in our computation algorithm.

Intuitively, from a given sequence of $\text{LTL}_{\mathcal{EL}}^{\mathbf{X},\mathbf{G}}$-ABoxes, our methods can generalize a given set of example individuals into a $\text{LTL}_{\mathcal{EL}}^{\mathbf{X},\mathbf{G}}$-concept by applying the (k-)MSC and then the LCS. The result is an $\text{LTL}_{\mathcal{EL}}^{\mathbf{X},\mathbf{G}}$-concept that captures the shared properties of all individuals as precise as possible in $\text{LTL}_{\mathcal{EL}}^{\mathbf{X},\mathbf{G}}$. The resulting concept can serve as a query concept to be used in instance queries or it can be used in concept definitions (or GCIs) to augment the knowledge base.

Main extensions of the corresponding methods for $\mathcal{EL}$ are the normal form which can cause an exponential blow-up of the input concepts and the notion of concept padding. Clearly, an exhaustive normalization need not be necessary in every case. It would be desirable to devise a dynamic algorithm for normalization and even a rewriting procedure to obtain succinct $\text{LTL}_{\mathcal{EL}}^{\mathbf{X},\mathbf{G}}$ concepts.

Obvious extensions of our methods for the LCS and MSC are to learn w.r.t. a general TBox and to use more expressive DLs or temporal logics. In the longer run, we would like to use a bigger fragment of LTL, e.g., $\Diamond$ (eventually), and to investigate the case of learning temporalized conjunctive queries.

Acknowledgements: The authors are supported by the DFG through the Collaborative Research Center TRR 248[1] and the Research Training Group RoSI (GRK 1907).

REFERENCES

[1] A. Artale, R. Kontchakov, A. Kovtunova, V. Ryzhikov, F. Wolter, and M. Zakharyaschev. 2017. Ontology-Mediated Query Answering over Temporal Data: A Survey. In *24th Int. Sympos. on Temp. Representation and Reasoning, TIME.*

[2] F. Baader, S. Borgwardt, P. Koopmann, V. Thost, and A.-Y. Turhan. 2020. Semantic Technologies for Situation Awareness. *KI – Künstliche Intelligenz* (2020).

[3] F. Baader, S. Brandt, and C. Lutz. 2005. Pushing the $\mathcal{EL}$ Envelope. In *Proc. of the 19th Int. Joint Conf. on Artificial Intelligence IJCAI-05.*

[4] F. Baader, R. Küsters, and R. Molitor. 1999. Computing Least Common Subsumers in Description Logics with Existential Restrictions. In *Proc. of the 16th Int. Joint Conf. on Artificial Intelligence (IJCAI'99).*

[5] S. Brandt, A.-Y. Turhan, and R. Küsters. 2003. Extensions of Non-standard Inferences to Description Logics with transitive Roles. In *Proc. of the 10th Int. Conf. on Logic for Programming, AI, and Reasoning (LPAR'03) (LNCS).* Springer.

[6] A. Ecke, R. Peñaloza, and A.-Y. Turhan. 2013. Computing Role-depth Bounded Generalizations in the Description Logic $\mathcal{ELOR}$. In *Proceedings of the 36th German Conference on Artificial Intelligence (KI 2013) (LNAI).* Springer, 49–60.

[7] R. Küsters and R. Molitor. 2002. Approximating most specific concepts in description logics with existential restrictions. *AI Commun.* 15, 1 (2002).

[8] R. Peñaloza and A.-Y. Turhan. 2011. A Practical Approach for Computing Generalization Inferences in $\mathcal{EL}$. In *Proceedings of the 8th European Semantic Web Conference (ESWC'11) (LNCS).* Springer-Verlag.

[9] A. Pnueli. 1977. The Temporal Logic of Programs. In *Proc. of the 18th Annual Symposium on Foundations of Computer Science (SFCS '77).*

[10] A.-Y. Turhan. 2007. *On the Computation of Common Subsumers in Description Logics.* Doctoral Thesis. Technische Universität Dresden.

[11] B. Zarrieß and A.-Y. Turhan. 2013. Most Specific Generalizations w.r.t. General $\mathcal{EL}$-TBoxes. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI'13).* AAAI Press.

[1] https://perspicuous-computing.science, project ID 389792660