KI - Künstliche Intelligenz (2020) 34:291–301
https://doi.org/10.1007/s13218-020-00651-0

TECHNICAL CONTRIBUTION

Role-Value Maps and General Concept Inclusions in the Minimal Description Logic with Value Restrictions or Revisiting Old Skeletons in the DL Cupboard

Franz Baader[1] · Clément Théron[2]

Received: 31 October 2019 / Accepted: 6 March 2020 / Published online: 14 March 2020

Abstract
We investigate the impact that general concept inclusions and role-value maps have on the complexity and decidability of reasoning in the description logic $\mathcal{FL}_0$. On the one hand, we give a more direct proof for ExpTime-hardness of subsumption w.r.t. general concept inclusions in $\mathcal{FL}_0$. On the other hand, we determine restrictions on role-value maps that ensure decidability of subsumption, but we also show undecidability for the cases where these restrictions are not satisfied.

Keywords Description logic · Value restrictions · Role-value maps · $\mathcal{FL}_0$ · Decidability and complexity

1 Introduction

Description logics (DLs) [6] are a well-investigated family of logic-based knowledge representation formalisms, which are used to define ontologies in applications domains such as the semantic web [9, 19] and in biology and medicine [20]. DLs are descended from semantic networks [28, 29] and frames [26] via the knowledge representation system KL-ONE [17]. The design goal of KL-ONE was, on the one hand, to provide its users with a knowledge representation (KR) language that is equipped with a well-defined syntax and a formal, unambiguous semantics, which was not always true for early KR approaches such as semantic networks and frames. On the other hand, reasoning over knowledge bases written in this language was supposed to be tractable (i.e., realizable by polynomial-time inference procedures) [15]. Thus, it came as a considerable shock to the community when it was shown that the second requirement is not satisfied by the language employed by KL-ONE for two independent reasons.

On the one hand, KL-ONE provided its users with the *concept constructor role-value maps (RVMs)*, which can be employed to link role successor sets. For example, the concept described by the RVM

$$(\mathsf{child} \circ \mathsf{friend} \subseteq \mathsf{knows})$$

collects all individuals that know all the friends of their children. The general form of such an RVM is $(r_1 \circ \cdots \circ r_m \subseteq s_1 \circ \cdots \circ s_n)$, where $r_1, \ldots, s_n$ are roles (i.e., binary predicates). It was shown in [30] that the presence of RVMs actually makes reasoning in KL-ONE undecidable. As a consequence, general RVMs were removed from KL-ONE-based KR languages, and are not available in any of the DLs employed by today's DL systems. One possibility for avoiding the undecidability caused by RVMs is to restrict the roles occurring in them to being functional. This approach was employed by the CLASSIC system [14], where the corresponding constructor is called the same-as constructor. However, using same-as in place of RVMs only overcomes the undecidability problem if no general concept inclusions (GCIs) are available in the terminological formalism [5].

An alternative approach for restricting RVMs with the goal of achieving decidability is to consider global role-value maps in which role composition on the right-hand side is disallowed. In contrast to the local RVMs used in the KL-ONE system, a global RVM is not a concept constructor, but a terminological axiom, which states that the inclusion must

✉ Franz Baader
franz.baader@tu-dresden.de

Clément Théron
clement.theron@ens-paris-saclay.fr

[1] TU Dresden, Dresden, Germany

[2] ENS Paris-Saclay, Cachan, France

hold for all individuals in the interpretation domain. For example, while $(\mathsf{child}{\circ}\mathsf{friend} \subseteq \mathsf{knows})$ is a concept that is interpreted as the set of individuals that know all the friends of their children, the global RVM

$$(\mathsf{child}{\circ}\mathsf{friend} \sqsubseteq \mathsf{knows})$$

states that all individuals in the interpretation domain know all the friends of their children.[1] Global RVMs may still cause undecidability [3, 8] unless one imposes additional restrictions. For the inexpressive DL $\mathcal{EL}$ [4, 10], adding global RVMs of the form $(r_1 \circ \cdots \circ r_m \sqsubseteq s)$ leaves reasoning not only decidable, but also tractable even in the presence of GCIs [3, 4]. For more expressive DLs, additional restrictions on such RVMs need to be imposed to keep reasoning decidable [22, 24].

On the other hand, KL-ONE provided its users with the concept constructors conjunction ($\sqcap$) and value restriction ($\forall r.C$). For example, using these constructors one can build the concept

$$\mathsf{Person} \sqcap \forall \mathsf{child} \cdot \forall \mathsf{friend} \cdot \mathsf{Nice},$$

which describes the persons all of whose children have only nice friends. It was shown in [27] that the subsumption problem in the DL $\mathcal{FL}_0$, which has only these two constructors, is coNP-hard in the presence of the simplest terminological formalism, which are so-called acyclic TBoxes. For cyclic TBoxes, the complexity increases to PSpace [1, 25], and for general TBoxes consisting of GCIs even to ExpTime [4, 21]. Thus, w.r.t. general TBoxes, subsumption reasoning in $\mathcal{FL}_0$ is as hard as in $\mathcal{ALC}$, its closure under negation.

In the present paper, we first reconsider this ExpTime-hardness result. The proof in [4] leads to a rather long chain of reductions, which makes it hard to understand the reasons for ExpTime-hardness and to reuse the proof ideas for other DLs. The proof of ExpTime-hardness given in [21] is by a reduction from deciding the winner in pushdown games [31], but the argument used for showing the correctness of this reduction is based on a non-trivial proof-theoretic characterization of subsumption in $\mathcal{FL}_0$ w.r.t. general TBoxes. In Sect. 3, we provide a new reduction from the problem of deciding the winner in countdown games [23], for which we must show, among other things, that we can express (binary representations) of natural numbers and subtraction of such numbers in $\mathcal{FL}_0$. ExpTime-hardness for pushdown games and for countdown games was respectively shown in [23, 31] by direct reductions from the problem of the acceptance of a word by a linearly space-bounded alternating Turing machine.

In the second part of the paper (Sects. 4, 5), we investigate the effect that adding RVMs has on the decidability of subsumption in $\mathcal{FL}_0$. On the one hand, we introduce two classes of global RVMs that leave subsumption without GCIs decidable. A simple example of a class of RVMs that satisfy both restrictions are length-preserving RVMs. On the other hand, we show that the restrictions made to achieve decidability are really needed: (1) in the presence of GCIs, even adding a single length-preserving RVM can cause undecidability; (2) for unrestricted RVMs, we give a simpler proof of the result first shown in [30] that undecidability even holds without GCIs. The results (1) and (2) are independent of whether global or local RVMs are used.

Some of the more technical proofs of these results are not given in detail here. They can be found in [12].

2 The Description Logic $\mathcal{FL}_0$

In Description Logic, *concept constructors* are used to build complex *concepts* out of *concept names* (unary predicates) and *role names* (binary predicates). A particular DL is determined by the available constructors. Starting with mutually disjoint countably infinite sets N_C and N_R of concept and role names, respectively, the set of $\mathcal{FL}_0$ concepts is inductively defined as follows:

- $\top$ (top concept) and every concept name $A \in N_C$ is an $\mathcal{FL}_0$ concept,
- if C, D are $\mathcal{FL}_0$ concepts and $r \in N_R$ is a role name, then $C \sqcap D$ (conjunction) and $\forall r.C$ (value restriction) are $\mathcal{FL}_0$ concepts.

The *semantics* of $\mathcal{FL}_0$ concepts is defined using first-order interpretations $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ consisting of a non-empty domain $\Delta^\mathcal{I}$ and an interpretation function $\cdot^\mathcal{I}$ that assigns a set $A^\mathcal{I} \subseteq \Delta^\mathcal{I}$ to each concept name A, and a binary relation $r^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$ to each role name r. This function is extended to $\mathcal{FL}_0$ concepts as follows:

$$\top^\mathcal{I} = \Delta^\mathcal{I} \quad \text{and} \quad (C \sqcap D)^\mathcal{I} = C^\mathcal{I} \cap D^\mathcal{I},$$
$$(\forall r.C)^\mathcal{I} = \{x \in \Delta^\mathcal{I} \mid \forall y \in \Delta^\mathcal{I} \colon (x, y) \in r^\mathcal{I} \Rightarrow y \in C^\mathcal{I}\}.$$

A (general) $\mathcal{FL}_0$ *TBox* $\mathcal{T}$ is a finite set of *general concept inclusions (GCIs)*, which are expressions of the form $C \sqsubseteq D$ for $\mathcal{FL}_0$ concepts C, D. The interpretation $\mathcal{I}$ is a *model* of $\mathcal{T}$ if it satisfies all the GCIs in $\mathcal{T}$, i.e. $C^\mathcal{I} \subseteq D^\mathcal{I}$ holds for all GCIs $C \sqsubseteq D$ in $\mathcal{T}$. Given an $\mathcal{FL}_0$ TBox $\mathcal{T}$ and two $\mathcal{FL}_0$ concepts C, D, we say that C is *subsumed* by D (denoted as $C \sqsubseteq_\mathcal{T} D$) if $C^\mathcal{I} \subseteq D^\mathcal{I}$ for all models $\mathcal{I}$ of $\mathcal{T}$. These two concepts are *equivalent* (denoted as $C \equiv_\mathcal{T} D$) if $C \sqsubseteq_\mathcal{T} D$ and $D \sqsubseteq_\mathcal{T} C$. If the TBox is empty, we write $C \sqsubseteq D$ and $C \equiv D$ instead of $C \sqsubseteq_\emptyset D$ and $C \equiv_\emptyset D$.

[1] Following the syntax introduced in [2], we distinguish between local and global RVMs by using $\subseteq$ in the syntax of the former and $\sqsubseteq$ in the syntax of the latter.

For $\mathcal{FL}_0$, the subsumption problem w.r.t. general TBoxes is ExpTime-complete: the upper bound follows from the well-known ExpTime upper bound for $\mathcal{ALC}$ [8], which contains $\mathcal{FL}_0$ as a sublogic. Previous proofs of the lower bound can be found in [4, 21]. In the next section, we will provide a new proof of this hardness result.

Without TBox, subsumption and equivalence in $\mathcal{FL}_0$ can be characterized using inclusion of formal languages. This characterization relies on transforming $\mathcal{FL}_0$ concepts into an appropriate normal form as follows. First, the semantics given to the concept constructors in $\mathcal{FL}_0$ implies that value restrictions distribute over conjunction, i.e., for all $\mathcal{FL}_0$ concepts C, D and roles r it holds that $\forall r.(C \sqcap D) \equiv \forall r.C \sqcap \forall r.D$. Using this equivalence as a rewrite rule from left to right, every $\mathcal{FL}_0$ concept can be transformed into an equivalent one that is either $\top$ or a conjunction of concepts of the form $\forall r_1 \cdots \forall r_n.A$, where $r_1, \dots, r_n$ are role names and A is a concept name. Such a concept can be abbreviated as $\forall w.A$, where $w = r_1 \dots r_n$ is a word over the alphabet N_R. Note that $n = 0$ means that w is the empty word ε, and thus $\forall \varepsilon.A$ corresponds to A. Furthermore, a conjunction of the form $\forall w_1.A \sqcap \cdots \sqcap \forall w_m.A$ can be written as $\forall L.A$ where $L \subseteq N_R^*$ is the finite language $\{w_1, \dots, w_m\}$. We use the convention that $\forall \emptyset.A$ corresponds to the top concept $\top$. Thus, any two $\mathcal{FL}_0$ concepts C, D containing only the concept names $A_1, \dots, A_\ell$ can be represented as

$$\begin{aligned} C &\equiv \forall K_1.A_1 \sqcap \cdots \sqcap \forall K_\ell.A_\ell, \\ D &\equiv \forall L_1.A_1 \sqcap \cdots \sqcap \forall L_\ell.A_\ell, \end{aligned} \tag{1}$$

where $K_1, L_1, \dots, K_\ell, L_\ell$ are finite languages over the alphabet of role names N_R. We call this representation the *language normal form (LNF)* of C, D.

If C, D have the LNFs shown above, then $C \sqsubseteq D$ holds iff $L_1 \subseteq K_1, \dots, L_\ell \subseteq K_\ell$ [11]. A similar characterization of subsumption can actually also be given in the presence of a TBox, but then $K_1, \dots, L_\ell$ are regular languages represented by automata of size exponential in the size of $\mathcal{T}$ [7].

3 ExpTime-Hardness of $\mathcal{FL}_0$ with GCIs

We give a new proof of the fact that subsumption in $\mathcal{FL}_0$ w.r.t. a general TBox is ExpTime-hard. This proof is by reduction from the problem of deciding the winner in countdown games, which are two-player games for which deciding which player has a winning strategy is known to be ExpTime-complete [23].

As defined in [23], a *countdown game* is given by a weighted graph (S, T), where S is the finite set of *states* and $T \subseteq S \times (\mathbb{N} \setminus \{0\}) \times S$ is the finite *transition relation*. If $t = (s, d, s') \in T$, then we say that the *duration* of the transition t is d. A *configuration* of a countdown game is a pair (s, c), where $s \in S$ is a state and $c \in \mathbb{N}$. A *move* of a countdown game from a configuration (s, c) is performed in the following way: first Player 1 chooses a number d such that $0 < d \leq c$ and there is $s' \in S$ with $(s, d, s') \in T$; then Player 2 chooses a transition $(s, d, s') \in T$ of duration d; the new configuration resulting from this move is then $(s', c - d)$. There are two types of *terminal* configurations, i.e., configurations (s, c) in which no more moves are available. If $c = 0$ then the configuration (s, c) is terminal and is a *winning configuration for Player 1*. If for all transitions $(s, d, s') \in T$ from the state s we have that $d > c$, then the configuration (s, c) is terminal and it is a *winning configuration for Player 2*. The algorithmic problem of *deciding the winner* in countdown games is the following problem: given a weighted graph (S, T) and a configuration (s_0, c_0), where all the durations of transitions and the number c_0 are assumed to be represented in binary, to determine whether Player 1 has a winning strategy from the configuration (s_0, c_0). Formally, a *winning strategy* can be described by a tree whose nodes are labeled with configurations. The root is labeled with the initial configuration, and the leaves are labeled with terminal configurations that are winning configurations for Player 1. A node labeled with a non-terminal configuration (s, c) has successor nodes that correspond to one choice of Player 1 followed by all possible choices of Player 2, i.e., to obtain the successors of this node, one chooses *one* number d that corresponds to an eligible choice of Player 1, and then adds successors nodes for all possible choices that Player 2 has: for each transition $(s, d, s') \in T$ a node with label $(s', c - d)$ is added. Theorem 2 in Section 4.2 of [23] shows that the problem of deciding whether such a winning strategy exists for a given countdown game and initial configuration is ExpTime-complete by a reduction from the word problem for linearly space-bounded alternating Turing machine.

When describing such games in a logic, one must, on the one hand, be able to express the alternation of choices between the players. In the presence of both existential and universal quantifiers, this is usually easy to achieve, but it is less clear how to do this in $\mathcal{FL}_0$, which has only universal quantification at its disposal. We will discuss at the end of this section how this is achieved in our reduction. In [21], this point is somewhat hidden in the proof-theoretic argument. On the other hand, one must be able to describe the moves and the winning conditions, which for countdown games means that one must be able to represent subtraction and a zero test. Again, it is quite surprising that this is possible in a logic like $\mathcal{FL}_0$, which has conjunction as its only Boolean operation. In [21], this part appears to be easier since configurations in pushdown games are words over a finite alphabet, which can directly be represented using sequences of value restrictions.

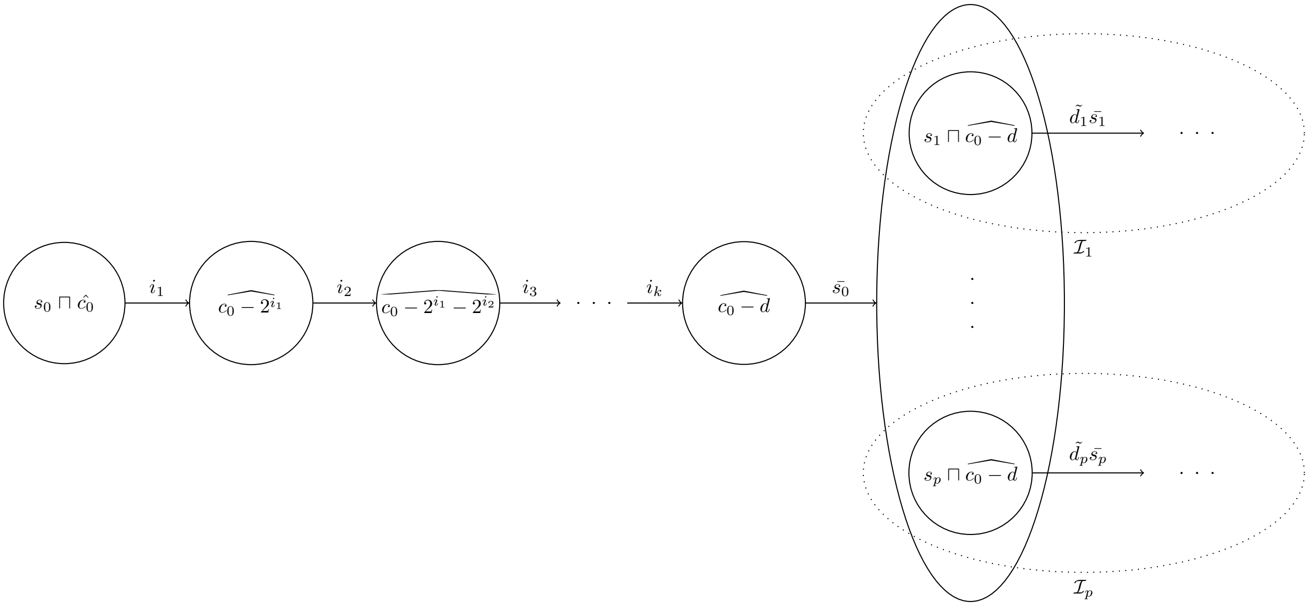

Fig. 1 The interpretation corresponding to a winning strategy of Player 1

Proposition 1 *Deciding the winner in a countdown game* (S, T) *with initial configuration* (s_0, c_0) *can be reduced in polynomial time to non-subsumption in* $\mathcal{FL}_0$ *w.r.t. a general TBox.*

Proof (sketch[2]) Let ℓ be the maximum between the number of bits needed to represent c_0 and the number of bits needed to represent any of the numbers occurring in T in binary.

We assume without loss of generality that

$$S \uplus \{F\} \uplus \bigcup_{i=0}^{\ell} \{\mathsf{b}_i = 0, \mathsf{b}_i = 1\} \subseteq N_C, \quad \text{and}$$

$$\{\bar{s} \mid s \in S\} \uplus \{i \mid 0 \le i \le \ell\} \subseteq N_R.$$

Note that the number of concept and role names contained in these sets is polynomial in the size of the input since ℓ is bounded by the size of the binary representation of the largest number occurring in the input.

The idea is that each element of an interpretation $\mathcal{I}$ is labeled by one (or several[3]) number(s) written in binary: $\mathsf{b}_i = 0$ means that the ith bit of this number is equal to 0, and $\mathsf{b}_i = 1$ that it is equal to 1. In addition, if $(x, y) \in i^{\mathcal{I}}$, the number labeling y should be the same as the one labeling x minus 2^i.

The subsumption relationship we want to test is

$$s_0 \sqcap \hat{c}_0 \sqsubseteq_{\mathcal{T}} F,$$

where $\hat{c}_0$ stands for the conjunction of all $\mathsf{b}_i = k_i$, where k_i is the value of the ith bit in the binary representation of c_0. The concept F stands for "fail", i.e., a configuration where Player 1 has no winning strategy.

The goal is to define the TBox $\mathcal{T}$ such that any model of $\mathcal{T}$ that does not satisfy the subsumption corresponds to a winning strategy for Player 1. To do this, we use the fact that, if Player 1 has a winning strategy in configuration (s_0, c_0) whose first step chooses duration d, then for all $(s_0, d, s') \in T$, Player 1 must also have a winning strategy in configuration $(s', c_0 - d)$. Thus, if $s_1, \dots, s_p$ are the states such that $(s_0, d, s_i) \in T$, then we can construct inductively the structures corresponding to the winning strategies on $(s_i, c_0 - d)$, as shown in Fig. 1 (where $\mathcal{I}_i$ is the interpretation corresponding to a winning strategy in configuration $(s_i, c_0 - d)$).

Given a duration d occurring in T, we write $\forall \tilde{d}$ as an abbreviation for $\forall i_1 i_2 \dots i_k$, where $i_1, i_2, \dots, i_k$ are the bits equal to 1 in the binary representation of d, written in decreasing order.[4]

The TBox $\mathcal{T}$ consists of the following GCIs:

1. $s \sqsubseteq \forall \tilde{d} \bar{s}. s'$ for all $(s, d, s') \in T$,

[2] A more detailed proof of this result can be found in [12].

[3] The absence of negation prevents us from enforcing a single number per element.

[4] The only condition needed is that the same number must always be represented in the same order, and using decreasing order is an easy way to achieve this.

2. $s \sqcap \mathsf{b}_i = 1 \sqcap \sqcap_{d \in E_s} \forall \tilde{d}\bar{s}.F \sqsubseteq F$ for all $s \in S$ and all $i, 1 \leq i \leq \ell$, where $E_s = \{d \mid \exists s' \text{ s.t. } (s, d, s') \in T\}$,
3. $\mathsf{b}_i = x \sqsubseteq \forall k.\mathsf{b}_i = x$ for all $0 \leq i < k \leq \ell$ and $x \in \{0, 1\}$,
4. $\mathsf{b}_i = 1 \sqcap \mathsf{b}_j{=}x \sqsubseteq \forall k.\mathsf{b}_j{=}x$ for all $k \leq i < j$ and $x \in \{0, 1\}$,
5. $\sqcap_{j=k}^{i-1} \mathsf{b}_j{=}0 \sqcap \mathsf{b}_i = 1 \sqsubseteq \forall k.(\sqcap_{j=k}^{i-1} \mathsf{b}_j{=}1 \sqcap \mathsf{b}_i = 0)$ for all $k \leq i$,
6. $\mathsf{b}_i = x \sqsubseteq \forall \bar{s}.\mathsf{b}_i = x$ for all $s \in S, 1 \leq i \leq \ell$, and $x \in \{0, 1\}$,
7. $\sqcap_{j=i}^{\ell} \mathsf{b}_j{=}0 \sqsubseteq \forall i.F$ for all $i, 1 \leq i \leq \ell$,
8. $F \sqsubseteq \forall i.F$ for all $i, 1 \leq i \leq \ell$,
9. $F \sqsubseteq \forall \bar{s}.F$ for all $s \in S$.

The intuition underlying these GCIs is the following:

- The GCIs in 1. say that, if we choose the duration d, then we must consider every state accessible this way.

  The GCIs in 2. reflect the fact that, if at least one of the configurations in which we could end up this way does not have a winning strategy, then choosing d does not yield a winning strategy either, unless the number of the configuration is already 0.
- The GCIs in 3., 4., 5., and 6. are there to ensure that subtraction is performed properly.
- The GCIs in 7. say that, if we choose a duration greater than the number in the current configuration (which is checked by verifying that the subtraction would return a negative number), then this leads to a failure for Player 1. The GCIs in 8. and 9. propagate this information forward to the next element corresponding to a configuration (so that it can then be propagated backwards using the GCIs in 2.).

Given this intuition, it is easy to see that a winning strategy for Player 1 can be used to construct a model of $\mathcal{T}$ of the form shown in Fig. 1, in which F is interpreted as the empty set. This yields a counterexample to the subsumption. Conversely, we can show that a counterexample to the subsumption can be used to extract a winning strategy for Player 1, but proving this is a bit more involved (see [12] for details). □

Given the ExpTime-hardness result for deciding the winner in countdown games shown in [23], this proposition yields the following hardness result for $\mathcal{FL}_0$.

Theorem 1 *Subsumption in $\mathcal{FL}_0$ w.r.t. general TBoxes is ExpTime-hard.*

Before closing this section, let us analyze how our reduction has addressed the two main requirements for such a reduction mentioned above Proposition 1. First, consider the alternation of choices between the players. The fact that it is sufficient that one choice of Player 1 is successful is expressed by the GCI 2. In fact, if one considers the contraposition of this GCI, then it becomes clear that is says: if Player 1 does not fail, then it already has reached a winning configuration or there is a successor configuration in which it does not fail. The universal quantification over the moves of Player 2 is expressed using the GCI 1. Regarding the representation of numbers in binary, one should note that we cannot enforce that every individual is associated with only one number. In fact, since $\mathcal{FL}_0$ cannot express disjointness of concepts, we cannot prevent an individual from belonging to both $\mathsf{b}_i = 0$ and $\mathsf{b}_i = 1$. This is not a problem for the following reasons. Of course, when constructing an interpretation from a winning strategy, one can just build a counter-interpretation to the subsumption where every element has a unique associated number, which is given by the strategy tree. Conversely, having more numbers in a counter-interpretation can only make satisfying the conditions for a winning strategy formulated by the GCIs harder to achieve. By removing superfluous elements and numbers, one can modify the given interpretation to one that looks like the one depicted in Fig. 1 and still satisfies all GCIs. From this, the winning strategy can then be extracted.

4 Decidable Role-Value Maps in $\mathcal{FL}_0$

As already mentioned in the introduction, role-value maps come in two variants [8]: local RVMs as originally introduced in KL-ONE are concept constructors whereas global RVMs are axioms that constrain the interpretation of roles. To be more precise,

- a *local role-value map* is a concept constructor with syntax $(r_1 \circ \cdots \circ r_m \subseteq s_1 \circ \cdots \circ s_n)$ where $r_1, \ldots, s_n$ are role names. To define its semantics, let

  $(t_1 \circ \cdots \circ t_k)^{\mathcal{I}}(d) = \{e \mid (d, e) \in t_1{}^{\mathcal{I}} \circ \cdots \circ t_k{}^{\mathcal{I}}\},$

  for role names $t_1, \ldots, t_k$, where "$\circ$" on the right-hand side is composition of binary relations. Then, $(r_1 \circ \cdots \circ r_m \subseteq s_1 \circ \cdots \circ s_n)^{\mathcal{I}} = \{d \in \Delta^{\mathcal{I}} \mid (r_1 \circ \cdots \circ r_m)^{\mathcal{I}}(d) \subseteq (s_1 \circ \cdots \circ s_n)^{\mathcal{I}}(d)\}$.
- a *global role-value maps* has the syntax $(r_1 \circ \cdots \circ r_m \sqsubseteq s_1 \circ \cdots \circ s_n)$, and is viewed to be an axiom. An interpretation $\mathcal{I}$ is a model of this axiom if $(r_1 \circ \cdots \circ r_m)^{\mathcal{I}}(d) \subseteq (s_1 \circ \cdots \circ s_n)^{\mathcal{I}}(d)$ holds for all $d \in \Delta^{\mathcal{I}}$.

In the presence of GCIs, local RVMs can express global ones since the global RVM $(r_1 \circ \cdots \circ r_m \sqsubseteq s_1 \circ \cdots \circ s_n)$ has the same models as the GCI $\top \sqsubseteq (r_1 \circ \cdots \circ r_m \subseteq s_1 \circ \cdots \circ s_n)$.

In the present section we consider only global RVMs without GCIs.

To simplify notation, we write $t_1 \dots t_k$ in place of $t_1 \circ \cdots \circ t_k$, and again view this expression as a word over the alphabet of role names. Thus, a set $\mathcal{T}$ of global RVMs can be written as $\mathcal{T} = \{u_1 \sqsubseteq v_1, \dots, u_k \sqsubseteq v_k\}$ where $u_1, \dots, v_k \in N_R^*$. Such a set induces the following string-rewriting relation [13] between words over N_R:

$$v \rightarrow_{\mathcal{T}} u \text{ iff there are } x, y \in N_R^* \text{ and } 1 \le i \le n \\ \text{such that } v = x v_i y \text{ and } u = x u_i y.$$

As usual, we denote the reflexive, transitive closure of $\rightarrow_{\mathcal{T}}$ as $\overset{*}{\underset{\mathcal{T}}{\rightarrow}}$. Given a formal language L over N_R, i.e., a subset of N_R^*, we now define the languages

$$L^{\downarrow_{\mathcal{T}}} = \{x \in N_R^* \mid \exists y \in L \text{ with } y \overset{*}{\underset{\mathcal{T}}{\rightarrow}} x\},$$

$$L^{\uparrow_{\mathcal{T}}} = \{x \in N_R^* \mid \exists y \in L \text{ with } x \overset{*}{\underset{\mathcal{T}}{\rightarrow}} y\},$$

which can be used to characterize subsumption w.r.t. $\mathcal{T}$ as follows.

Theorem 2 *Let $\mathcal{T}$ be a finite set of global RVMs, and C, D be $\mathcal{FL}_0$ concepts with LNFs as in (1). Then the following are equivalent:*

1. *$C \sqsubseteq_{\mathcal{T}} D$, i.e., $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for all models of $\mathcal{T}$;*
2. *$L_i \subseteq K_i^{\downarrow_{\mathcal{T}}}$ for all i, $1 \le i \le \ell$;*
3. *$\{w\}^{\uparrow_{\mathcal{T}}} \cap K_i \neq \emptyset$ for all i, $1 \le i \le \ell$ and $w \in L_i$.*

Regarding the proof of this theorem, first note that 2. and 3. are easily seen to be equivalent. In fact, given a word $w \in L_i$, we have $w \in K_i^{\downarrow_{\mathcal{T}}}$ iff $\exists y \in K_i$ with $y \overset{*}{\underset{\mathcal{T}}{\rightarrow}} w$ iff $\exists y \in K_i$ with $y \in \{w\}^{\uparrow_{\mathcal{T}}}$ iff $\{w\}^{\uparrow_{\mathcal{T}}} \cap K_i \neq \emptyset$.

Our proof of 2. ⇒ 1. uses the following proposition, which is an easy consequence of the semantics of global RVMs and value restrictions, and our definition of $\rightarrow_{\mathcal{T}}$.

Proposition 2 *If $x \overset{*}{\underset{\mathcal{T}}{\rightarrow}} y$, then $\forall x.A \sqsubseteq_{\mathcal{T}} \forall y.A$ holds for all $A \in N_C$.*

This proposition yields that $C \sqsubseteq_{\mathcal{T}} \forall w.A_i$ holds for all $w \in K_i^{\downarrow_{\mathcal{T}}}$ and all i, $1 \le i \le \ell$. If $L_i \subseteq K_i^{\downarrow_{\mathcal{T}}}$, then this implies that $C \sqsubseteq_{\mathcal{T}} \forall L_i.A_i$ for all i, $1 \le i \le \ell$, and thus we have $C \sqsubseteq_{\mathcal{T}} D$.

We show 1. ⇒ 2. by contraposition. Thus, assume that there is an i and a word $w = t_1 \dots t_p$ such that $w \in L_i \setminus K_i^{\downarrow_{\mathcal{T}}}$. We use w and i to build a counterexample to the subsumption $C \sqsubseteq_{\mathcal{T}} D$, i.e., a model $\mathcal{I}_{w,i}$ of $\mathcal{T}$ in which $C^{\mathcal{I}} \not\subseteq D^{\mathcal{I}}$. To construct $\mathcal{I}_{w,i}$, we start with a sequence of individuals $d_0, \dots, d_p$ and connect them with the roles in w, i.e., we set $(d_0, d_1) \in t_1^{\mathcal{I}_{w,i}}, \dots, (d_{p-1}, d_p) \in t_p^{\mathcal{I}_{w,i}}$. Then we extend $\mathcal{I}_{w,i}$ with additional individuals in order to add the role paths required by the RVMs in $\mathcal{T}$. Assume that $r_1 \dots r_m \sqsubseteq s_1 \dots s_n$ is a RVM in $\mathcal{T}$, and there are two individuals d, e in the domain of $\mathcal{I}_{w,i}$ such that $e \in (r_1 \circ \cdots \circ r_m)^{\mathcal{I}_{w,i}}(d)$, but $e \notin (s_1 \circ \cdots \circ s_n)^{\mathcal{I}_{w,i}}(d)$. Then we add new individuals $f_1, \dots, f_{n-1}$ to the domain of $\mathcal{I}_{w,i}$ and connect them via the roles $s_1, \dots, s_n$ as follows: $(d, f_1) \in s_1^{\mathcal{I}_{w,i}}, (f_1, f_2) \in s_2^{\mathcal{I}_{w,i}}, \dots, (f_{n-1}, e) \in s_n^{\mathcal{I}_{w,i}}$. In general, this process needs to be iterated infinitely, and the resulting interpretation $\mathcal{I}_{w,i}$ is the limit obtained by this infinite process. To ensure that this interpretation indeed satisfies all RVMs in $\mathcal{T}$, the extension process needs to be organized in a fair way, i.e., for every pair of individuals d, e and RVM $r_1 \dots r_m \sqsubseteq s_1 \dots s_n$ violated by this pair (as described above), we must eventually apply an extension step that removes this violation. This is, however, not hard to achieve (e.g., by labeling such violations with the iteration step in which they were introduced, and then always choosing one of the oldest violations). Finally, we interpret concept names in $\mathcal{I}_{w,i}$ as follows: all individuals except for d_p belong to all concept names, whereas d_p belongs to all concept names except for A_i. The following proposition is now easy to show.

Proposition 3 *The interpretation $\mathcal{I}_{w,i}$ satisfies all the RVMs in $\mathcal{T}$, and for all words u we have that $(d_0, d_p) \in u^{\mathcal{I}_{w,i}}$ implies $u \overset{*}{\underset{\mathcal{T}}{\rightarrow}} w$.*

Since $\mathcal{I}_{w,i}$ is a model of $\mathcal{T}$, it is sufficient to show that $d_0 \in C^{\mathcal{I}_{w,i}} \setminus D^{\mathcal{I}_{w,i}}$. First, suppose that $d_0 \in C^{\mathcal{I}_{w,i}}$ does not hold. By our definition of the interpretation of concept names in $\mathcal{I}_{w,i}$, this can only be the case if there is a word $u \in K_i$ such that $(d_0, d_p) \in u^{\mathcal{I}_{w,i}}$. The above proposition yields $u \overset{*}{\underset{\mathcal{T}}{\rightarrow}} w$, and thus $w \in K_i^{\downarrow_{\mathcal{T}}}$, contradicting our choice of w. Consequently, we must have $d_0 \in C^{\mathcal{I}_{w,i}}$. Finally, we have $d_0 \notin D^{\mathcal{I}_{w,i}}$ since $w \in L_i$, $(d_0, d_p) \in w^{\mathcal{I}_{w,i}}$, and $d_p \notin A_i^{\mathcal{I}_{w,i}}$. This completes the proof of Theorem 2.

In order to derive decidability results for subsumption w.r.t. RVMs in $\mathcal{FL}_0$ from this theorem, we need to find restrictions under which the condition 2. or 3. is decidable. We say that the finite set of RVMs $\mathcal{T}$ is *downward (upward) admissible* if for every finite language L we can effectively compute a representation of $L^{\downarrow_{\mathcal{T}}}$ ($L^{\uparrow_{\mathcal{T}}}$) for which the word problem is decidable. We say that $\mathcal{T}$ is *admissible* if it is downward admissible or upward admissible.

Corollary 1 *The following problem is decidable:*

Given: *A finite, admissible set of global RVMs $\mathcal{T}$ and $\mathcal{FL}_0$ concepts C, D.*
Question: *Does $C \sqsubseteq_{\mathcal{T}} D$ hold or not?*

Proof If $\mathcal{T}$ is downward admissible, then we can use condition 2 in Theorem 2 to decide subsumption: to test whether $L_i \subseteq K_i^{\downarrow_\mathcal{T}}$, we must decide for each of the finitely many words $u \in L_i$ whether $u \in K_i^{\downarrow_\mathcal{T}}$, which is possible since the word problem for $K_i^{\downarrow_\mathcal{T}}$ is decidable.

If $\mathcal{T}$ is upward admissible, then we can use condition 3 in Theorem 2: to check whether$\{w\}^{\uparrow_\mathcal{T}} \cap K_i \neq \emptyset$ it is sufficient to decide, for the finitely many words $u \in K_i$ whether $u \in \{w\}^{\uparrow_\mathcal{T}}$. □

To the best of our knowledge, this is the first decidability result for RVMs where role composition is allowed on the right-hand side. On the one hand, one can use our result to find syntactic restrictions on global RVMs that ensure decidability.

For example, if all RVMs $u_i \sqsubseteq v_i$ in $\mathcal{T}$ satisfy $|u_i| \leq |v_i|$, then $L^{\downarrow_\mathcal{T}}$ is also finite (and thus trivially has a decidable word problem) and can effectively be computed. Thus, such a set of RVMs is downward admissible. Symmetrically $|u_i| \geq |v_i|$ for all RVMs $u_i \sqsubseteq v_i$ in $\mathcal{T}$ implies that $\mathcal{T}$ is upward admissible. On the other hand, one can also have downward (upward) admissible sets of RVMs where the languages $L^{\downarrow_\mathcal{T}}$ ($L^{\uparrow_\mathcal{T}}$) are not necessarily finite, but one can compute a finite automaton or a pushdown automaton accepting them.

Example 1 Consider the set of global RVMs $\mathcal{T} = \{rr \sqsubseteq r, s \sqsubseteq ss\}$. Given a finite language L over the alphabet $\{r, s\}$, the languages $L^{\downarrow_\mathcal{T}}$ and $L^{\uparrow_\mathcal{T}}$ need not be finite. But it is easy to see that they are both regular, and thus have a decidable word problem, which shows that $\mathcal{T}$ is both upward and downward admissible. For example, given a word of the form $w = r^{m_1}s^{n_1} \dots r^{m_k}s^{n_k}$ with $m_i, n_i \geq 1$, we have$\{w\}^{\downarrow_\mathcal{T}} = \{r^{m'_1}s^{n'_1} \dots r^{m'_k}s^{n'_k} \mid m'_i \geq m_i, 1 \leq n'_i \leq n_i\}$,which is obviously regular.

In general, one needs to analyze the specific set of RVMs $\mathcal{T}$ to find out whether a finite or pushdown automaton exists for the upward or downward languages. This is akin to the approach used in [22, 24] to deal with RVMs without composition of roles on the right-hand side, but for expressive DLs. There, the set of RVMs needs to satisfy a regularity restriction in order to obtain decidability.

5 Undecidable Role-Value Maps in $\mathcal{FL}_0$

The decidability results proved in the previous section depend, on the one hand, on the absence of GCIs. On the other hand, they require the string-rewriting system induced by the role-value maps to be well-behaved (see the definition of *admissible* above).

First, we show that, even without GCIs, RVMs can cause undecidability in $\mathcal{FL}_0$. This result was first proved in [30] for an extension of $\mathcal{FL}_0$[5] with local role value maps of the form $(r_1 \circ \cdots \circ r_m = s_1 \circ \cdots \circ s_n)$, which in our notation can be written as $(r_1 \circ \cdots \circ r_m \subseteq s_1 \circ \cdots \circ s_n) \sqcap (s_1 \circ \cdots \circ s_n \subseteq r_1 \circ \cdots \circ r_m)$. Below, we will first show undecidability for global RVMS, and then extend this result to the case of local RVMs without TBox, using an approach first employed in [30]. We will say more about the connection between our proofs and the one in [30] after having presented our proofs.

Theorem 3 *There exists a fixed finite set of global role-value maps $\mathcal{T}$ such that subsumption of $\mathcal{FL}_0$ concepts w.r.t. $\mathcal{T}$ is undecidable.*

Proof We prove this theorem by reduction from the word problem for string-rewriting systems. As shown in [13] (Theorem 2.5.9), there is a fixed finite string-rewriting system R such that its word problem (i.e., given two words u, v, decide whether $u \overset{*}{\leftrightarrow}_R v$ holds or not) is undecidable. Here $\overset{*}{\leftrightarrow}_R$ denotes the reflexive, transitive, and *symmetric* closure of the rewrite relation

$$\rightarrow_R = \{(xu_iy, xv_iy) \mid \quad (u_i, v_i) \in R \text{ and } x, y \in \Sigma^*\},$$

where Σ is the finite alphabet over which the strings in R are built.

Let $R = \{(u_i, v_i) \mid 1 \leq i \leq n\}$ be such a string-rewriting system over the alphabet Σ. We set $N_R = \Sigma$ and define the set of global RVMs corresponding to R as $\mathcal{T}_R = \{u_1 \sqsubseteq v_1, \dots, u_n \sqsubseteq v_n, v_1 \sqsubseteq u_1, \dots, v_n \sqsubseteq u_n\}$.

It is easy to see that the relations $\overset{*}{\leftrightarrow}_R$ and $\overset{*}{\rightarrow}_{\mathcal{T}_R}$ coincide.

Now, assume that, given words u, v over Σ, we want to test whether $u \overset{*}{\leftrightarrow}_R v$ holds. We claim that this is the case iff $\forall u.A \sqsubseteq_{\mathcal{T}_R} \forall v.A$ holds.

In fact, by Theorem 2 we know that $\forall u.A \sqsubseteq_{\mathcal{T}_R} \forall v.A$ holds iff $\{v\} \subseteq \{u\}^{\downarrow_{\mathcal{T}_R}}$. The latter is obviously equivalent to $u \overset{*}{\rightarrow}_{\mathcal{T}_R} v$, which in turn is equivalent to $u \overset{*}{\leftrightarrow}_R v$. □

Using a trick originally introduced in [30], we can easily transfer this undecidability result from global RVMs to local ones.

[5] In [30], the logic that is extended with RVMs is actually called $\mathcal{AL}$, which later on was used in the DL literature [18] to denote the extension of $\mathcal{FL}_0$ by atomic negation ($\neg A$) and unqualified existential restriction ($\exists r.\top$). However, in addition to the RVMs only the concept constructors conjunction and value restriction are used in [30] to show undecidability.

Corollary 2 *Subsumption in $\mathcal{FL}_0$ extended with local role-value maps is undecidable even without a TBox.*

Proof Given a set of global RVMs $\mathcal{T}$ for which subsumption of $\mathcal{FL}_0$ concepts w.r.t. $\mathcal{T}$ is undecidable and two $\mathcal{FL}_0$ concepts C, D, we build an $\mathcal{FL}_0$ concept with local RVMs E such that $C \sqsubseteq_{\mathcal{T}} D$ holds iff $C \sqcap E \sqsubseteq D$.

Let Σ be the set of all role names occurring in C, D, and $\mathcal{T}$, and let s be a new role name not contained in Σ. We define the concept E as follows:

$$\begin{aligned} E = {} & \sqcap_{u \sqsubseteq v \in \mathcal{T}} (u \subseteq v) \sqcap \forall s.\big(\sqcap_{u \sqsubseteq v \in \mathcal{T}} (u \subseteq v)\big) \\ & \sqcap_{r \in \Sigma} (r \subseteq s) \sqcap \sqcap_{r \in \Sigma} (sr \subseteq s). \end{aligned}$$

The local RVMs in the second line ensure that, for any element d of E, the role s can be used to access all elements reachable via a role chain from d, and thus the first line enforces that d and all these reachable elements satisfy the RVMs in $\mathcal{T}$. Given this intuition, it is then easy to see that the statements $C \sqsubseteq_{\mathcal{T}} D$ and $C \sqcap E \sqsubseteq D$ are indeed equivalent. □

Regarding the comparison with the proof in [30], first note that global RVMs are not considered there, but one could easily adapt the approach in [30] to show undecidability also for global RVMs. Second, as mentioned above, the RVMs considered in [30] use equality rather than inclusion, which appears to be less expressive, and thus makes the undecidability result stronger. However, considering our reduction in the proof of Theorem 3, we see that we actually have the RVMs in both directions, and thus can replace them by ones using equality only. This is not the case for Corollary 2 since there the local RVMs $(r \subseteq s)$ and $(sr \subseteq s)$ are used only in one direction. In the general setting considered in that corollary, adding the RVMs in the other direction would not work. Basically, this is the reason why the proof in [30] uses a reduction from the word problem for finitely presented *groups*, rather than the word problem for finitely presented *semigroups* (which is the same as the word problem for string-rewriting systems employed in our proof). On a quite technical level, this is explained in [30] in a remark below the proof of Theorem 3.5. The main difference between the two proofs is, however, that we obtain our result as a simple consequence of the elegant characterization of subsumption in $\mathcal{FL}_0$ with RVMs given in Theorem 2, whereas the proof of correctness of the reduction in [30] is rather technical.

Next, we show that, in the presence of GCIs, undecidability can also be caused by global RVMs that satisfy the admissibility condition introduced in the previous section. In fact, we will see that a single global RVM of the form $tr \sqsubseteq rt$ is sufficient to obtain undecidability. Since this RVM is length-preserving, it is both downward and upward admissible.

Theorem 4 *Subsumption $C \sqsubseteq_{\mathcal{T}} D$ of $\mathcal{FL}_0$ concepts C, D w.r.t. TBoxes $\mathcal{T}$ consisting of $\mathcal{FL}_0$ GCIs and global role-value maps is undecidable. This is the case even if $\mathcal{T}$ contains only GCIs and a single global RVM of the form $tr \sqsubseteq rt$.*

Readers that are familiar with the undecidability proof for subsumption in $\mathcal{ALC}$ with global RVMs given in [8], which is by reduction from the tiling problem, may think that the proof of the above theorem should be an easy adaptation of the proof in [8]. A closer look at that proof reveals, however, that it makes extensive use of concept constructors not available in $\mathcal{FL}_0$ (such as negation, disjunction, and existential restrictions). In addition, it requires not only the RVM $tr \sqsubseteq rt$, but also its backward direction $rt \sqsubseteq tr$. The undecidability proof in [30] uses the concept constructors of $\mathcal{FL}_0$ and *local* role-value maps. In addition to being local instead of global, the RVMs used there are not admissible since they contain the defining relations of a finitely presented group with undecidable word problem.

The main new contribution of the proof sketched below is thus to show that one can obtain the undecidability results for a single global RVM of the form $tr \sqsubseteq rt$ also with the seriously restricted expressive power of $\mathcal{FL}_0$.

We prove Theorem 4 by a reduction from the halting problem for deterministic Turing machines (DTMs). Without loss of generality, we consider DTMs that have a one-side infinite tape, where the left-most tape cell is marked using the special symbol \$. Whenever the machine moves to the left onto this cell, in the next step it immediately goes to the right again and leaves the symbol \$ and the state unchanged. We also assume that the machine can only go left or right (i.e., it cannot stay in place).

The machine starts with an "empty" tape, i.e., a tape where the left-most cell contains \$ and all other cells contain the blank symbol B. The blank symbol and \$ cannot be written by the machine. It halts when a special halting state *halt* is reached. For all other states, there is a transition for every possible tape symbol. Clearly, the question whether such a DTM halts when started with the initial state q_0 on the empty tape is undecidable.

Let $M = (Q, \Sigma, \delta, q_0)$ be such a DTM. In order to encode the halting problem for M into a subsumption problem, we assume without loss of generality that $\{r, t\} \subseteq N_R$ (for "right" and "then") and $Q \cup \Sigma \cup \{H, N\} \subseteq N_C$ (the latter two for "halt" and "not-head").

The idea is to construct a set of GCIs that encodes the transition function δ of M such that a model should be a structure like the one shown in Fig. 2, which corresponds to the unique run of the machine started with the initial state q_0

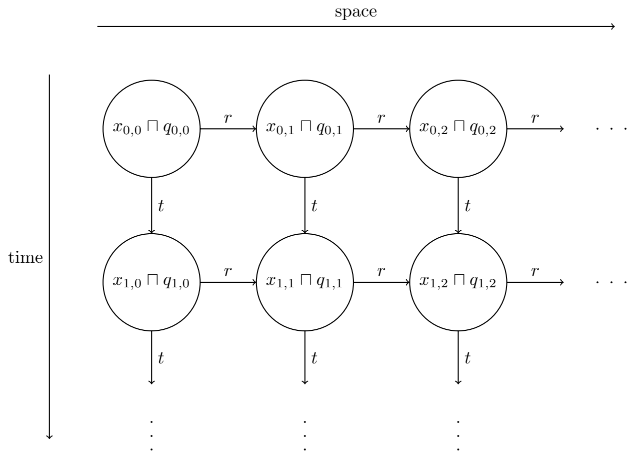

Fig. 2 A model of $\mathcal{T}_M$ corresponding to the unique run of M

on the empty tape (where $x_{i,j}$ is the letter at position j at step i of the run, and $q_{i,j}$ is either N, if the head is not at position j at step i, or the state of the machine at step i otherwise).

More formally, the TBox $\mathcal{T}_M$ consists of the global RVM $tr \sqsubseteq rt$ together with the following GCIs, whose rôle will be explained later:

1. $\forall t.H \sqsubseteq H$
2. $\forall r.H \sqsubseteq H$
3. $B \sqsubseteq \forall r.(B \sqcap N)$
4. $N \sqcap a \sqsubseteq \forall t.a$ for all $a \in \Sigma$
5. $N \sqcap \forall rr.N \sqsubseteq \forall tr.N$
6. $\$ \sqcap \forall r.N \sqsubseteq \forall t.N$
7. $\forall r.(q \sqcap a) \sqsubseteq \forall t.(N \sqcap \forall r.(b \sqcap \forall r.q'))$
   if $\delta(q,a) = (q', b, \rightarrow)$
8. $\forall r.(q \sqcap a) \sqsubseteq \forall t.(q' \sqcap \forall r.(b \sqcap \forall r.N))$
   if $\delta(q,a) = (q', b, \leftarrow)$
9. $(q \sqcap a) \sqsubseteq H$ if $\delta(q,a) = halt$
10. $(q \sqcap \$) \sqsubseteq \forall t.(\$ \sqcap \forall r.q)$

Intuitively, these GCIs have the following meaning:

- The first two GCIs propagate the information that the machine has reached the halting state backwards through time and space (see Fig. 2).
- The third GCI reflects the fact that, if there is a B on a cell, then the machine never went in a position further than the one of this cell (since the machine never writes B). Thus, the letter of any cell to the right of a B should be B too, and the head of the machine cannot be there.
- The fourth GCI reflects the fact that, if the head is not on a cell at step n, then the letter on this cell should be the same at step $n+1$.
- The fifth GCI reflects the fact that the head of the tape can only move one cell at a time, so if the head is not directly to the left or to the right of a cell at step n, it cannot be on this cell at step $n+1$.
- The sixth GCI reflects the same kind of idea: if the head is not directly to the right of the leftmost cell at step n, then it cannot be on this cell at step $n+1$.
- The seventh GCI describe the behavior of the machine when it makes a transition that moves to the right. Note that the left-hand side of this GCI applies to a tape position $j-1$ that is immediately to the left of the position j where the head is. It checks whether the state is q and the letter at position j is a, and then ensures the changes required by the transition $\delta(q,a) = (q', b, \rightarrow)$ are realized for the next time point $t+1$, i.e., the head is not at the current position $j-1$, at the position j we now have the symbol b, the head is two positions to the right (i.e., at position $j+1$) with state q'. Note that the fifth GCI ensures that the head cannot be at position j at time point $t+1$.
- The working of the eighth GCI can be explained similarly.
- The ninth GCI recognizes the fact that the machine halts. This information is then propagate backwards by the first two GCIs.
- The tenth GCI describes the fact that, when the head is on the \$ symbol, the machine has to go right and stay in the same state.

The following lemma shows the correctness of the reduction, and thus yields the undecidability result stated in Theorem 4.

Lemma 1 *The DTM M halts when started with the initial state q_0 on the empty tape iff*

$$\$ \sqcap \forall r.(B \sqcap q_0) \sqsubseteq_{\mathcal{T}_M} H.$$

Proof (Sketch) "⇐" If M does not halt then one can use its run to create a model of $\mathcal{T}_M$ that looks like the structure depicted in Fig. 2, and where the element in the upper left corner belongs to the left-hand side $\$ \sqcap \forall r.(B \sqcap q_0)$ of the subsumption statement. Since the machine does not halt, one can interpret H as the empty set without violating the GCIs (1), (2), and (9). In fact, if H is empty then $\forall t.H$ is also empty since every element has a t-successor, and thus cannot satisfy this value restriction. This shows that GCI (1) is satisfied. The same argument applies to GCI (2). GCI (9) is satisfied since the machine does not reach the halting state by assumption. Thus, we have a counterexample to the subsumption.

"⇒" The converse direction is a bit trickier. Basically, we show that a counterexample $(x, \mathcal{I})$ to the subsumption contains the structure induced by the run of M and depicted in Fig. 2 as a kind of substructure. To be more precise, we can show by induction on n that, for all i, if $(x, y) \in (t^n r^i)^{\mathcal{I}}$, then

- $y \in a^{\mathcal{I}}$ where a is the letter on the i-th cell at the n-th step of the run,
- $y \in N^{\mathcal{I}}$ if the head of the machine is not on the i-th cell at the n-th step, and
- $y \in q^{\mathcal{I}}$ if the head is on the i-th cell and the state of the machine is q at the n-th step.

Note that the implications in the other direction need not hold, i.e., it can well be that y also belongs to other states q' or letters a' in $\mathcal{I}$, and that y may also belong to N if the head is actually there.

Nevertheless, we can then argue that, since the right-hand side H of the subsumption does not contain the individual x, the run must be a non-halting run. Otherwise, an element $y \in H^{\mathcal{I}}$ reachable from x via the roles r and t would be forced to exist due to GCI (9), and H would then be propagated back to x due to the first two GCIs. □

Using the fact that GCIs and local RVMs can express global RVMs, or the trick employed in the proof of Corollary 2, we can transfer the undecidability result stated in Theorem 4 also to local RVMs and TBoxes of a restricted form.

Corollary 3 *Subsumption $C \sqsubseteq_{\mathcal{T}} D$ in $\mathcal{FL}_0$ extended with local role-value maps is undecidable even if*

1. *C, D are $\mathcal{FL}_0$ concepts and $\mathcal{T}$ contains GCIs between $\mathcal{FL}_0$ concepts and a single GCI of the form $\top \sqsubseteq (tr \subseteq rt)$ involving a local RVM, or*
2. *D is an $\mathcal{FL}_0$ concept, $\mathcal{T}$ contains only GCIs between $\mathcal{FL}_0$ concepts, and $C = C' \sqcap E$ for an $\mathcal{FL}_0$ concept C' and a fixed concept E of $\mathcal{FL}_0$ extended with local role-value maps.*

6 Conclusion

In this paper we have, on the one hand, given a more direct proof of the known fact that subsumption in $\mathcal{FL}_0$ w.r.t. GCIs is ExpTime-hard. We believe that the ideas underlying the reduction employed in this proof may turn out to be helpful for showing ExpTime-hardness for other inexpressive DLs. On the other hand, we have determined decidable and undecidable cases for $\mathcal{FL}_0$ extended with role-value maps. For the case without a TBox, we have shown that admissible global RVMs leave the subsumption problem decidable. What remains open is the question whether the same is true for admissible local RVMs. For the decidable cases, it would also be interesting to investigate the complexity of the subsumption problem, depending on the form of the available RVMs.

Acknowledgements Open Access funding provided by Projekt DEAL. We thank the reviewers for their pertinent comments, which helped us to improve the paper. Franz Baader was partially supported by the Deutsche Forschungsgemeinschaft (DFG), Grant 389792660, as part of the collaborative research center cpec (TRR 248).

References

1. Baader F (1996) Using automata theory for characterizing the semantics of terminological cycles. Ann Math Artif Intell 18:175–219
2. Baader F (2003) Description logic terminology. In: [6], pp. 485–495
3. Baader F (2003) Restricted role-value-maps in a description logic with existential restrictions and terminological cycles. In: Calvanese D, De Giacomo G, Franconi E (eds) Proceedings of the 2003 description logic workshop (DL 2003), CEUR Workshop Proceedings, vol 81. CEUR-WS.org
4. Baader F, Brandt S, Lutz C (2005) Pushing the *EL* envelope. In: Kaelbling LP, Saffiotti A (eds) Proceedings of the 19th international joint conference on artificial intelligence (IJCAI 2005). Morgan Kaufmann, Los Altos, pp 364–369
5. Baader F, Bürckert HJ, Nebel B, Nutt W, Smolka G (1993) On the expressivity of feature logics with negation, functional uncertainty, and sort equations. J Logic Lang Inf 2:1–18
6. Baader F, Calvanese D, McGuinness D, Nardi D, Patel-Schneider PF (eds) (2003) The description logic handbook: theory, implementation, and applications. Cambridge University Press, Cambridge
7. Baader F, Gil OF, Pensel M (2018) Standard and non-standard inferences in the description logic FL_0 using tree automata. In: Lee D, Steen A, Walsh T (eds) Proceedings of the 4th global conference on artificial intelligence (GCAI-2018), EPiC Series in Computing, vol 55. EasyChair, pp 1–14
8. Baader F, Horrocks I, Lutz C, Sattler U (2017) An introduction to description logic. Cambridge University Press, Cambridge
9. Baader F, Horrocks I, Sattler U (2002) Description logics for the semantic web. KI 16(4):57–59
10. Baader F, Lutz C, Turhan AY (2010) Small is again beautiful in description logics. KI 24(1):25–33

11. Baader F, Narendran P (2001) Unification of concept terms in description logics. J Symb Comput 31(3):277–305
12. Baader F, Théron C (2019) Role-value maps and general concept inclusions in the description logic FL_0. LTCS-Report 19-08, Chair of Automata Theory, Institute of Theoretical Computer Science, Technische Universität Dresden, Dresden, Germany. https://tu-dresden.de/inf/lat/reports#BaTh-LTCS-19-08
13. Book RV, Otto F (1993) String-rewriting systems. Springer, New York
14. Borgida A, Patel-Schneider PF (1994) A semantics and complete algorithm for subsumption in the CLASSIC description logic. J Artif Intell Res 1:277–308
15. Brachman RJ, Levesque, HJ (1984) The tractability of subsumption in frame-based description languages. In: Proceedings of the 4th national conference on artificial intelligence (AAAI'84), pp 34–37
16. Brachman RJ, Levesque HJ (eds) (1985) Readings in knowledge representation. Morgan Kaufmann, Los Altos
17. Brachman RJ, Schmolze JG (1985) An overview of the KL-ONE knowledge representation system. Cogn Sci 9(2):171–216
18. Donini FM, Lenzerini M, Nardi D, Nutt W (1991) The complexity of concept languages. In: Allen J, Fikes R, Sandewall E (eds) Proceedings of the 2nd international conference on the principles of knowledge representation and reasoning (KR'91). Morgan Kaufmann, Los Altos, pp 151–162
19. Glimm B, Stuckenschmidt H (2016) 15 years of semantic web: an incomplete survey. KI 30(2):117–130
20. Hoehndorf R, Schofield PN, Gkoutos GV (2015) The role of ontologies in biological and biomedical research: a functional perspective. Brief Bioinform 16(6):1069–1080
21. Hofmann M (2005) Proof-theoretic approach to description-logic. In: Panangaden P (ed) Proceedings of the 20th IEEE symposium on logic in computer science (LICS 2005). IEEE Computer Society Press, pp 229–237
22. Horrocks I, Kutz O, Sattler U (2006) The even more irresistible $SROIQ$. In: Doherty P, Mylopoulos J, Welty CA (eds.) Proceedings of the 10th international conference on principles of knowledge representation and reasoning (KR 2006). AAAI Press/The MIT Press, Lake District, UK, pp 57–67
23. Jurdzinski M, Sproston J, Laroussinie F (2008) Model checking probabilistic timed automata with one or two clocks. Logic Methods Comput Sci 4:3
24. Kazakov Y (2008) RIQ and $SROIQ$ are harder than $SHOIQ$. In: Brewka G, Lang J (eds) Proceedings of the 11th international conference on principles of knowledge representation and reasoning (KR 2008). AAAI Press, pp 274–284
25. Kazakov Y, de Nivelle H (2003) Subsumption of concepts in FL_0 for (cyclic) terminologies with respect to descriptive semantics is PSPACE-complete. In: Proceedings of the 2003 description logic workshop (DL 2003). CEUR electronic workshop proceedings. http://CEUR-WS.org/Vol-81/
26. Minsky M (1975) A framework for representing knowledge. In: Haugeland J (ed) Mind design. The MIT Press (1981). A longer version appeared in the psychology of computer vision. Republished in [16]
27. Nebel B (1990) Terminological reasoning is inherently intractable. Artif Intell 43:235–249
28. Quillian MR (1967) Word concepts: a theory and simulation of some basic capabilities. Behav Sci 12:410–430 **(Republished in [16])**
29. Quillian MR (1968) Semantic memory. In: Minsky M (ed) Semantic information processing. The MIT Press, London, pp 216–270
30. Schmidt-Schauß M (1989) Subsumption in KL-ONE is undecidable. In: Brachman RJ, Levesque HJ, Reiter R (eds) Proceedings of the 1st international conference on the principles of knowledge representation and reasoning (KR'89). Morgan Kaufmann, Los Altos, pp 421–431
31. Walukiewicz I (2001) Pushdown processes: games and model-checking. Inf Comput 164(2):234–263