RWTH
LTCS–Report

Aachen University of Technology
Research group for
Theoretical Computer Science

Matching under Side Conditions in Description Logics

Franz Baader Sebastian Brandt Ralf Küsters

LTCS-Report 01-02

RWTH Aachen
LuFg Theoretische Informatik
http://www-lti.informatik.rwth-aachen.de

Ahornstr. 55
52074 Aachen
Germany

Matching under Side Conditions in Description Logics

Franz Baader and Sebastian Brandt
LuFG Theoretical Computer Science
RWTH Aachen

Ralf Küsters
Institut für Informatik und Praktische Mathematik
Christian-Albrechts-Universität zu Kiel

Abstract

Whereas matching in Description Logics is now relatively well-investigated, there are only very few formal results on matching under additional side conditions, though these side conditions were already present in the original paper by Borgida and McGuinness introducing matching in DLs. The present report closes this gap for the DL $\mathcal{ALN}$ and its sublanguages.

Contents

1 Introduction

The traditional inference problems (like subsumption) in description logics (DLs) are now well-investigated, which means that there exist complexity results and algorithms for a great variety of DLs of differing expressive power [9] as well as optimized implementations of the algorithms for expressive DLs [11]. In contrast, matching concepts against patterns is a relatively new inference problem in DLs, which has originally been introduced in [6, 13] to help filter out the unimportant aspects of large concepts appearing in knowledge bases of the CLASSIC system [8]. More recently, matching (as well as the more general problem of unification) has been proposed as a tool for detecting redundancies in knowledge bases [3] and to support the integration of knowledge bases by prompting possible interschema assertions [5].

All three applications have in common that one wants to search a large knowledge base for concepts having a certain (not completely specified) form. This "form" can be expressed with the help of so-called *concept patterns*, i.e., concept descriptions containing variables. For example, the pattern $D := X \sqcap \forall \mathsf{child}.(Y \sqcap \mathsf{Female})$ looks for concepts that restrict the child role to fillers that are Female, such as the concept $C := (\geq 1\,\mathsf{child}) \sqcap \forall \mathsf{child}.(\mathsf{Female} \sqcap \mathsf{Rich})$. In fact, applying the substitution $\sigma := \{X \mapsto (\geq 1\,\mathsf{child}), Y \mapsto \mathsf{Rich}\}$ to the pattern D yields a concept equivalent to C, i.e., σ is a solution (matcher) of the matching problem $C \equiv^? D$.[1]

This type of matching problems has been investigated in detail for sublanguages of the DLs $\mathcal{ALN}$ and $\mathcal{ALE}$ in [2] and [1], respectively. In particular, it was shown that, for sublanguages of $\mathcal{ALN}$, solvable matching problems always have a least matcher (w.r.t. subsumption), which can be computed in polynomial time. For sublanguages of $\mathcal{ALE}$, deciding solvability of matching problems modulo equivalence is already NP-complete.

In [6, 13], the expressivity of matching problems was further enhanced by allowing for additional *side conditions* on the variables (through the as-construct): a (strict) subsumption condition is of the form $X \sqsubseteq^? E$ ($X \sqsubset^? E$) where X is a variable and E a pattern, and it restricts the matchers to substitutions satisfying $\sigma(X) \sqsubseteq \sigma(E)$ ($\sigma(X) \sqsubset \sigma(E)$). Using a subsumption condition, the matching problem of the above example can be written more

[1] We restrict our attention to such matching problems *modulo equivalence* since matching *modulo subsumption*, as introduced in [6], can be reduced to matching modulo equivalence [2].

intuitively as $X \sqcap \forall \mathsf{child}.Z \equiv^? (\geq 1\, \mathsf{child}) \sqcap \forall \mathsf{child}.(\mathsf{Female} \sqcap \mathsf{Rich})$ under the subsumption condition $Z \sqsubseteq^? \mathsf{Female}$. One result of this paper is that also more complex sets of subsumption conditions do not extend the expressive power of matching problems (see below). However, they are often more convenient to state. In contrast, strict subsumption conditions cannot always be simulated by pure matching problems. They can, e.g., be used to avoid trivial matches. For example, the pattern $D' := X \sqcap \forall \mathsf{child}.Y$ matches every concept since $\forall \mathsf{child}.\top \equiv \top$ (where the top concept $\top$ stands for the set of all individuals). The additional strict subsumption condition $Y \sqsubset^? \top$ ensures that we can only match concepts with a real restriction on child.

The first (rather restricted) formal results on matching under side conditions were given in [2]: it was shown that matching under strict subsumption conditions in the small DL $\mathcal{FL}_0$ is already NP-hard, and that matching under so-called acyclic subsumption conditions can be reduced to matching without side conditions. However, [2] does not give a complexity upper bound for matching under strict subsumption conditions and the reduction for acyclic subsumption conditions given there is exponential.

This paper investigates in detail matching under side conditions in sublanguages of $\mathcal{ALN}$. We will show that matching under subsumption conditions can be reduced in polynomial time to matching without side conditions. In particular, this implies that solvable matching problems under subsumption conditions in sublanguages of $\mathcal{ALN}$ always have a least matcher, which can be computed in polynomial time. For strict subsumption conditions, matching is shown to be NP-complete in the sublanguages $\mathcal{FL}_\bot$ and $\mathcal{FL}_\neg$ of $\mathcal{ALN}$.

2 Description logics

2.1 Syntax and semantics

Concept descriptions are inductively defined with the help of a set of concept *constructors*, starting with a set N_C of *concept names* and a set N_R of *role names*. In this paper, we consider concept descriptions built from the constructors shown in Table 1. In the description logic $\mathcal{FL}_0$, concept descriptions are formed using the constructors top-concept ($\top$), conjunction ($C \sqcap D$), and value restriction ($\forall r.C$). The description logic $\mathcal{FL}_\bot$ additionally provides us with the bottom concept ($\bot$), and $\mathcal{FL}_\neg$ also allows for primitive negation ($\neg P$). Finally, $\mathcal{ALN}$ extends $\mathcal{FL}_\neg$ with number restrictions ($\geq n\, r$)

Syntax	Semantics	$\mathcal{FL}_0$	$\mathcal{FL}_\bot$	$\mathcal{FL}_\neg$	$\mathcal{ALN}$
$\top$	$\Delta^\mathcal{I}$	x	x	x	x
$C \sqcap D$	$C^\mathcal{I} \cap D^\mathcal{I}$	x	x	x	x
$\forall r.C$	$\{x \in \Delta^\mathcal{I} \mid \forall y\colon (x,y) \in r^\mathcal{I} \rightarrow y \in C^\mathcal{I}\}$	x	x	x	x
$\bot$	$\emptyset$		x	x	x
$\neg P$, $P \in N_C$	$\Delta^\mathcal{I} \setminus P^\mathcal{I}$			x	x
$(\geq n\, r)$, $n \in \mathbb{N}$	$\{x \in \Delta^\mathcal{I} \mid \#\{y \mid (x,y) \in r^\mathcal{I}\} \geq n\}$				x
$(\leq n\, r)$, $n \in \mathbb{N}$	$\{x \in \Delta^\mathcal{I} \mid \#\{y \mid (x,y) \in r^\mathcal{I}\} \leq n\}$				x

Table 1: Syntax and semantics of concept descriptions.

and $(\leq n\, r)$ (see Table 1).

As usual, the semantics of concept descriptions is defined in terms of an *interpretation* $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$. The domain $\Delta^\mathcal{I}$ of $\mathcal{I}$ is a non-empty set and the interpretation function $\cdot^\mathcal{I}$ maps each concept name $P \in N_C$ to a set $P^\mathcal{I} \subseteq \Delta^\mathcal{I}$ and each role name $r \in N_R$ to a binary relation $r^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$. The extension of $\cdot^\mathcal{I}$ to arbitrary concept descriptions is defined inductively, as shown in the second column of Table 1.

One of the most important traditional inference services provided by DL systems is computing the subsumption hierarchy. The concept description C is *subsumed* by the description D ($C \sqsubseteq D$) iff $C^\mathcal{I} \subseteq D^\mathcal{I}$ holds for all interpretations $\mathcal{I}$; C and D are *equivalent* ($C \equiv D$) iff they subsume each other; C is *strictly subsumed* by D ($C \sqsubset D$) iff $C \sqsubseteq D$ and $C \not\equiv D$. For all DLs listed in Table 1, subsumption can be decided in polynomial time using a structural subsumption algorithm [7].

2.1.1 A normal form for concept descriptions

It is easy to see that any $\mathcal{FL}_\bot$-concept description can be transformed into an equivalent description that is either $\top$ or a (nonempty) conjunction of descriptions of the form $\forall r_1. \cdots \forall r_m.A$, where $r_1, \ldots, r_m$ are $m \geq 0$ (not necessarily distinct) roles, and A is the bottom concept $\bot$ or a concept name. We abbreviate $\forall r_1. \cdots \forall r_m.A$ by $\forall r_1 \ldots r_m.A$, where $r_1 \ldots r_m$ is viewed as a word over the alphabet N_R of all role names. If $m = 0$, then this is the empty word ε, and thus $\forall \varepsilon.A$ is our "abbreviation" for A. In addition, instead of $\forall w_1.A \sqcap \ldots \sqcap \forall w_\ell.A$ we write $\forall L.A$ where $L := \{w_1, \ldots, w_\ell\}$ is a finite set of words over N_R; we define $\forall \emptyset.A \equiv \top$. Using these abbreviations, any $\mathcal{FL}_\bot$-

concept description C containing only concept names in the finite set $\mathcal{C} \subseteq N_C$ can be written as

$$C \equiv \forall U_\bot.\bot \sqcap \prod_{A \in \mathcal{C}} \forall U_A.A$$

where U_H for $H \in \mathcal{C} \cup \{\bot\}$ are finite sets of words over N_R (called *role languages*). This representation of C will subsequently be called its *U-labeled normal form*.

As an example consider the $\mathcal{FL}_\bot$-concept description $C_{ex} := \forall r.(\bot \sqcap \forall r.\bot) \sqcap \forall r.\forall s.A \sqcap \forall s.A$. Its $\mathcal{FL}_0$-normal form C'_{ex} is $\forall\{r, rr\}.\bot \sqcap \forall\{rs, s\}.A$.

Similar normal forms exist for concept descriptions in $\mathcal{FL}_\neg$ and $\mathcal{ALN}$. In $\mathcal{FL}_\neg$, an additional role language for every negated atomic concept is necessary; normal forms in $\mathcal{ALN}$ require an additional role language for every negated atomic concept and one for every number restriction.

2.1.2 Characterization of subsumption

Normal forms as introduced in the previous section can be used to characterize subsumption of concept descriptions. The relevant results for $\mathcal{ALN}$ and its sublanguages are provided in [2]. For $\mathcal{FL}_\bot$, we obtain the following lemma:

Lemma 1 *Characterization of subsumption in $\mathcal{FL}_\bot$*
Let C and D be $\mathcal{FL}_\bot$-concept descriptions. Let C be in U-labeled normal form and let D be in V-labeled normal form. Then, $C \sqsubseteq D$ iff the following two conditions hold:
1. $U_\bot \cdot N_R^* \supseteq V_\bot \cdot N_R^*$
2. $U_A \cup U_\bot \cdot N_R^* \supseteq V_A \cup V_\bot \cdot N_R^*$ for all $A \in \mathcal{C}$.

In preparation of the characterization of subsumption in $\mathcal{FL}_\neg$ and $\mathcal{ALN}$, we need to introduce the notion of excluding words.

Definition 2 *Excluding words*
Let C be an $\mathcal{FL}_\neg$-concept description in U-labeled normal form. Let D be an $\mathcal{ALN}$-concept description. For C, define the role language $\widehat{U}_\bot$ as follows:

$$\widehat{U}_\bot := U_\bot \cup \bigcup_{A \in \mathcal{C}} (U_A \cap U_{\neg A})$$

*For D, the set of D-*excluding words *is defined by:*

$$E_D := \{w \in N_R^* \mid D \sqsubseteq \forall w.\bot\}$$

It can be shown that $E_D = \widehat{U}_\bot \cdot N_R^*$ for every $\mathcal{FL}_\neg$-concept description D in U-labeled normal form. Hence, in this case the notion of excluding words can be characterized by $\widehat{U}_\bot$. We shall see in Definition 32 that a characterization of excluding words for $\mathcal{ALN}$-concept descriptions is more complex. Subsumption of $\mathcal{FL}_\neg$-concept description can be characterized as follows.

Lemma 3 *Characterization of subsumption in $\mathcal{FL}_\neg$*
Let C and D be $\mathcal{FL}_\neg$-concept descriptions. Let C be in U-labeled normal form and let D be in V-labeled normal form. Then, $C \sqsubseteq D$ iff the following two conditions hold:
1. $\widehat{U}_\bot \cdot N_R^* \supseteq \widehat{V}_\bot \cdot N_R^*$
2. $U_A \cup \widehat{U}_\bot \cdot N_R^* \supseteq V_A \cup \widehat{V}_\bot \cdot N_R^*$ *for all* $H \in \mathcal{C} \cup \{\neg A | A \in C\}$

Subsumption in $\mathcal{ALN}$ was characterized by Küsters in [12], yielding the following result.

Lemma 4 *Characterization of subsumption in $\mathcal{ALN}$*
Let C, D be $\mathcal{ALN}$-concept descriptions. Let C be in U-labeled normal form. Let D be in V-labeled normal form. Then $C \sqsubseteq D$ iff all of the following conditions hold.
1. $E_C \supseteq E_D$
2. $U_A \cup E_C \supseteq V_A \cup E_D$ *for all* $A \in \mathcal{C}$
3. $U_{\neg A} \cup E_C \supseteq V_{\neg A} \cup E_D$ *for all* $A \in \mathcal{C}$
4. $\bigcup_{m \geq n} U_{(\geq mR)} \cup E_C \supseteq \bigcup_{m \geq n} V_{(\geq mR)} \cup E_D$ *for all* $(\leq nR) \in \mathcal{N}_\leq$ *with* $n \geq 1$
5. $\bigcup_{m \leq n} U_{(\leq mR)} \cup E_C \cdot R^{-1} \supseteq \bigcup_{m \leq n} V_{(\leq mR)} \cup E_D \cdot R^{-1}$ *for all* $(\geq nR) \in \mathcal{N}_\geq$

Note that two concept descriptions are equivalent if they subsume each other. In order to characterize equivalence it is therefore sufficient to replace all $(\supseteq)$-relations by $(=)$ in the above four lemmae.

2.2 Matching in description logics

In order to define concept patterns, we additionally need a set N_X of concept variables, which we assume to be disjoint from $N_C \cup N_R$. Informally, an $\mathcal{ALN}$-concept pattern is an $\mathcal{ALN}$-concept description over the concept names $N_C \cup N_X$ and the role names N_R, with the only exception that primitive

negation must not be applied to variables. More formally, *concept patterns* (denoted D, D') are defined using the following syntax rules:

$$D, D' \longrightarrow X \mid C \mid D \sqcap D' \mid \forall r.D,$$

where $X \in N_X$, $r \in N_R$, and C is an $\mathcal{ALN}$-concept description. For example, if X, Y are concept variables, r a role name, and A, B concept names, then $D := A \sqcap X \sqcap \forall r.(B \sqcap Y)$ is an $\mathcal{ALN}$-concept pattern, but $\neg X$ is *not*.

In analogy to the normal forms defined for concept descriptions, every $\mathcal{ALN}$-concept pattern D over a finite subset $\mathcal{X} \subseteq N_X$ of variables can be written as

$$D \equiv C \sqcap \bigsqcap_{X \in \mathcal{X}} V_X.X,$$

where C is an $\mathcal{ALN}$-concept description in V-labeled normal form. We call this the *V-labeled normal form* of the concept pattern D. The notion of a pattern, the normal form (and also the notions "substitution" and "matching problem" introduced below) can be restricted to sublanguages of $\mathcal{ALN}$ in the obvious way.

A *substitution* σ is a mapping from N_X into the set of all $\mathcal{ALN}$-concept descriptions. This mapping is extended to concept patterns in the usual way by replacing the occurrences of the variables X in the pattern by the corresponding concept description $\sigma(X)$. For example, if we apply the substitution $\sigma := \{X \mapsto A \sqcap B,\ Y \mapsto A\}$ to the pattern D from above, we obtain the description $\sigma(D) = A \sqcap A \sqcap B \sqcap \forall r.(B \sqcap A)$. The result of applying a substitution to an $\mathcal{ALN}$-concept pattern is always an $\mathcal{ALN}$-concept description. Note that this would no longer be the case if negation were allowed in front of concept variables.

Subsumption can be extended to substitutions as follows: the substitution σ is subsumed by the substitution τ ($\sigma \sqsubseteq \tau$) iff $\sigma(X) \sqsubseteq \tau(X)$ for all variables $X \in N_X$.

Definition 5 *Let C be an $\mathcal{ALN}$-concept description and D an $\mathcal{ALN}$-concept pattern. Then, $C \equiv^? D$ is an $\mathcal{ALN}$-*matching problem*. The substitution σ is a* solution *(*matcher*) of $C \equiv^? D$ iff $C \equiv \sigma(D)$.*

In the following, we will abbreviate a matching problem of the form $C \equiv^? C \sqcap D$ as $C \sqsubseteq^? D$. This notation is justified by the fact that σ solves $C \equiv^? C \sqcap D$ iff $C \sqsubseteq \sigma(D)$.

A matching problem can either be viewed as a *decision problem*, where one asks whether the problem is solvable, or as a *computation problem*, where

one asks for actual matchers of this problem (if any). Although the computation problem is usually the more interesting one, the decision problem can serve as a starting point for the complexity analysis. In general, matching problems may have several (even an infinite number of) solutions, and thus the question arises which matcher to compute. Following [6, 2] we will here concentrate on the problem of computing a least matcher (w.r.t. the ordering $\sqsubseteq$ on substitutions).

Instead of a single matching problem, we may also consider finite systems $\{C_1 \equiv^? D_1, \ldots, C_m \equiv^? D_m\}$ of such problems, which must be solved simultaneously. As shown in [2], solving such systems can, however, be reduced to solving the single matching problem

$$\forall r_1.C_1 \sqcap \cdots \sqcap \forall r_m.C_m \equiv^? \forall r_1.D_1 \sqcap \cdots \sqcap \forall r_m.D_m$$

where the r_i are pairwise distinct role names.

How to decide if a given matching problem is solvable and how to compute least matchers has been addressed in [2] and [12]. The next two subsections summarize the relevant results and recall some notions used in this context.

2.2.1 Solving the decision problem

In [2] and [12], matching modulo equivalence in $\mathcal{FL}_\bot$, $\mathcal{FL}_\neg$ and $\mathcal{ALN}$ is reduced to solving equations over formal languages, which we will refer to as *solvability equations*. By assigning appropriate values to the variables occurring in these equations the decision problem can be reduced to testing certain formal languages for equality. The structure of the languages involved guarantees that this test can be done by finite automata in polynomial time.

We begin by introducing solvability equations in $\mathcal{FL}_\bot$. Let $(C \equiv^? D)$ be an $\mathcal{FL}_\bot$-matching problem, where C is in U-labeled normal form and D is in V-labeled normal form.

Definition 6 *Solvability equations for* $(C \equiv^? D)$ *in* $\mathcal{FL}_\bot$

$$U_\bot \cdot N_R^* = V_\bot \cdot N_R^* \cup \bigcup_{X \in \mathcal{X}} V_X \cdot \xi_\bot^X \cdot N_R^* \qquad (\bot)$$

$$U_A \cup U_\bot \cdot N_R^* = V_A \cup U_\bot \cdot N_R^* \cup \bigcup_{X \in \mathcal{X}} V_X \cdot \xi_A^X \qquad (A)$$

for all $A \in \mathcal{C}$.

Solvability of the above system of equations is decided by assigning appropriate formal languages to the occurring variables. The following lemma specifies these formal languages.

Lemma 7 *Testing solvability in* $\mathcal{FL}_\bot$
The system of equations $(\bot), ((A) \mid A \in \mathcal{C})$ *has a solution iff:*

1. *For every* $X \in \mathcal{X}$*, replacing the expression* $\xi^X_\bot \cdot N^*_R$ *by the set* $\widehat{L}^X_\bot := \bigcap_{w \in V_X} w^{-1} \cdot (U_\bot \cdot N^*_R)$ *solves Equation* $(\bot)$*.*

2. *For every* $A \in \mathcal{C}$ *and for every* $X \in \mathcal{X}$*, replacing the variable* ξ^X_A *by the set* $\widehat{L}^X_A := \bigcap_{w \in V_X} w^{-1} \cdot (U_A \cup U_\bot \cdot N^*_R)$ *solves Equation* (A)*.*

Hence, by inserting the languages specified in the above lemma into the referring solvability equations, we obtain variable-free formal language equations valid if and only if the original matching problem is solvable. It is shown in [2] that validity of these equations can be tested in polynomial time using finite automata.

Analogous results exist for $\mathcal{FL}_\neg$ and $\mathcal{ALN}$. Let $(C \equiv^? D)$ be an $\mathcal{FL}_\neg$-matching problem, where C and D are in U-labeled and V-labeled normal forms respectively. Then the relevant solvability equations are defined as follows.

Definition 8 *Solvability equations for* $(C \equiv^? D)$ *in* $\mathcal{FL}_\neg$

$$\widehat{U}_\bot \cdot N^*_R = V_\bot \cdot N^*_R \cup \bigcup_{X \in \mathcal{X}} V_X \cdot \xi^X_\bot \cdot N^*_R \cup \bigcup_{A \in \mathcal{C}} Int(A, \neg A) \cdot N^*_R \qquad (\bot)$$

$$U_A \cup \widehat{U}_\bot \cdot N^*_R = V_A \cup \widehat{U}_\bot \cdot N^*_R \cup \bigcup_{X \in \mathcal{X}} V_X \cdot \xi^X_A \qquad (A)$$

$$U_{\neg A} \cup \widehat{U}_\bot \cdot N^*_R = V_{\neg A} \cup \widehat{U}_\bot \cdot N^*_R \cup \bigcup_{X \in \mathcal{X}} V_X \cdot \xi^X_{\neg A} \qquad (\neg A)$$

for all $A \in \mathcal{C}$*, where*

$$Int(A, \neg A) := \Big(V_A \cup \bigcup_{X \in \mathcal{X}} V_X \cdot \xi^X_A\Big) \cap \Big(V_{\neg A} \cup \bigcup_{X \in \mathcal{X}} V_X \cdot \xi^X_{\neg A}\Big).$$

Note that in the solvability equations for $\mathcal{FL}_\bot$, Equation $(\bot)$ was completely independent of role languages referring to atomic concepts $A \in \mathcal{C}$. For

$\mathcal{FL}_\neg$ this is no longer the case, because the conjunction of an atomic concept and its negation is inconsistent. For that reason, the expression *Int* is included in Equation $(\bot)$. The following lemma provides a test for solvability in $\mathcal{FL}_\neg$.

Lemma 9 *Testing solvability in* $\mathcal{FL}_\neg$
The system of equations $(\bot), ((A) \mid A \in \mathcal{C}), ((\neg A) \mid A \in \mathcal{C})$ *has a solution iff:*

1. *For every* $A \in \mathcal{C}$ *and for every* $X \in \mathcal{X}$*, replacing the variable* ξ_A^X *by the set* $\widehat{L}_A^X := \bigcap_{w \in V_X} w^{-1} \cdot (U_A \cup \widehat{U}_\bot \cdot N_R^*)$ *solves Equation* (A)*.*
2. *For every* $A \in \mathcal{C}$ *and for every* $X \in \mathcal{X}$*, replacing the variable* $\xi_{\neg A}^X$ *by the set* $\widehat{L}_{\neg A}^X := \bigcap_{w \in V_X} w^{-1} \cdot (U_{\neg A} \cup \widehat{U}_\bot \cdot N_R^*)$ *solves Equation* (A)*.*
3. *For every* $X \in \mathcal{X}$*, replacing the expression* $\xi_\bot^X \cdot N_R^*$ *by the set* $\widehat{L}_\bot^X := \bigcap_{w \in V_X} w^{-1} \cdot (U_\bot \cdot N_R^*)$ *together with the assignments proposed in (1) and (2) solves Equation* $(\bot)$*.*

Note that the third condition requires "together with the assignments proposed in (1) and (2)". This is necessary because of the expression *Int*, by which Equation $(\bot)$ becomes dependent on the other assignments. For $\mathcal{ALN}$, we have to introduce some notation first. Let $(C \equiv^? D)$ be an $\mathcal{ALN}$-matching problem, where C and D are in U-labeled and V-labeled normal forms respectively.

Definition 10 *The following tuples of variables are defined for the sake of readability.*

$$\begin{aligned}
\xi_\bot &:= (\xi_\bot^X \mid X \in \mathcal{X}) \\
\xi_\mathcal{C} &:= (\xi_A^X \mid X \in \mathcal{X}, A \in \mathcal{C}) \\
\xi_\neg &:= (\xi_{\neg A}^X \mid X \in \mathcal{X}, A \in \mathcal{C}) \\
\xi_\geq &:= (\xi_{(\geq nR)}^X \mid X \in \mathcal{X}, (\geq nR) \in \mathcal{N}_\geq) \\
\xi_\leq &:= (\xi_{(\leq nR)}^X \mid X \in \mathcal{X}, (\leq nR) \in \mathcal{N}_\leq)
\end{aligned}$$

Denote by α *an arbitrary assignment of finite languages to the variables contained in the tuples, i.e.* $\alpha(\xi_H^X) = L_H^X$ *for all* $X \in \mathcal{X}$ *and* $H \in \{\bot\} \cup \mathcal{C} \cup \{\neg A \mid A \in \mathcal{C}\} \cup \mathcal{N}_\leq \cup \mathcal{N}_\geq =: \mathcal{H}$*. Let* σ *be the substitution corresponding to* α*, so that for every* $X \in \mathcal{X}$ *we have:*

$$\sigma(\xi^X) = \bigsqcap_{H \in \mathcal{H}} \alpha(\xi_H^X).H$$

Denote by $E_D(\xi_\bot, \xi_\mathcal{C}, \xi_\neg, \xi_\geq, \xi_\leq)$ *the set of excluding words obtained for* D *relative to the assignment* α*. Thus, let*

$$E_D\big(\alpha(\xi_\bot), \alpha(\xi_\mathcal{C}), \alpha(\xi_\neg), \alpha(\xi_\geq), \alpha(\xi_\leq)\big) := E_{\sigma(D)},$$

yielding the set of $\sigma(D)$*-excluding words after assigning the occurring variables.*

The above construct is necessary, because the set of excluding words is defined only for concept descriptions and not for concept patterns. Consequently, we must assume some assignment of the concept variables occurring on the right-hand side of the matching problem. With these preparations, the following solvability equations are provided.

Definition 11 *Solvability equations in* $\mathcal{ALN}$
With the notation of the above definition, define the following formal language equations.

$$E_C = E_D(\xi_\bot, \xi_\mathcal{C}, \xi_\neg, \xi_\geq, \xi_\leq) \qquad (\bot)$$

$$U_A \cup E_C = V_A \cup E_C \cup \bigcup_{X\in\mathcal{X}} V_X{\cdot}\xi_A^X \qquad (A)$$

$$U_{\neg A} \cup E_C = V_{\neg A} \cup E_C \cup \bigcup_{X\in\mathcal{X}} V_X{\cdot}\xi_{\neg A}^X \qquad (\neg A)$$

$$\bigcup_{m\geq n} U_{(\geq mR)} \cup E_C = V_{(\geq mR)} \cup E_C \cup \bigcup_{X\in\mathcal{X}} V_X{\cdot}\xi_{(\geq nR)}^X \qquad (\geq nR)$$

$$\bigcup_{m\leq n'} U_{(\leq mR)} \cup E_C{\cdot}R^{-1} = V_{(\leq mR)} \cup E_C{\cdot}R^{-1} \cup \bigcup_{X\in\mathcal{X}} V_X{\cdot}\xi_{(\leq n'R)}^X \qquad (\leq n'R)$$

for all $A \in \mathcal{C}$*,* $n \in \mathbb{N}\setminus\{0\}$*,* $n' \in \mathbb{N}$*,* $(\geq nR) \in \mathcal{N}_\geq$*, and* $(\leq n'R) \in \mathcal{N}_\leq$*.*

Again, Equation $(\bot)$ takes into account role languages referring to other concepts than the $\bot$-concept. However, this property is syntactically hidden in the constructs E_C and E_D, which are defined as $\{w \in N_R^* \mid C \sqsubseteq \forall w.\bot\}$ and analogously for E_D, as introduced in Definition 2.

Lemma 12 *Testing solvability in* $\mathcal{ALN}$
Let $\widehat{L}_\bot^X := \bigcap_{w\in V_X} w^{-1}{\cdot}E_C$*. Then there exists a finite set* $L_\bot^X$ *of polynomial size in the input matching problem with* $L_\bot^X{\cdot}N_R^* = \widehat{L}_\bot^X$*.* [2] *The system of*

[2] As mentioned previously, this is shown in [2].

equations $(\bot), ((A) \mid A \in \mathcal{C}), ((\neg A) \mid A \in \mathcal{C}), ((\geq nR) \mid (\geq nR) \in \mathcal{N}_{\geq}), ((\leq nR) \mid (\leq nR) \in \mathcal{N}_{\leq})$ *then has a solution iff:*

1. *For every* $X \in \mathcal{X}$ *and* $A \in \mathcal{C}$*, replacing the variable* ξ_A^X *by the set* $L_A^X := \left(\bigcap_{w \in V_X} w^{-1} \cdot (U_A \cup E_C)\right) \setminus \widehat{L}_\bot^X$ *solves Equation* (A)*.*

2. *For every* $X \in \mathcal{X}$ *and* $A \in \mathcal{C}$*, replacing the variable* $\xi_{\neg A}^X$ *by the set* $L_{\neg A}^X := \left(\bigcap_{w \in V_X} w^{-1} \cdot (U_{\neg A} \cup E_C)\right) \setminus \widehat{L}_\bot^X$ *solves Equation* $(\neg A)$*.*

3. *For every* $X \in \mathcal{X}$ *and* $(\geq nR) \in \mathcal{N}_{\geq}$*, replacing the variable* $\xi_{\neg A}^X$ *by the set* $L_{(\geq nR)}^X := \left(\bigcap_{w \in V_X} w^{-1} \cdot (\bigcup_{m \geq n} U_{(\geq nR)} \cup E_C)\right) \setminus \widehat{L}_\bot^X$ *solves Equation* $(\geq nR)$*.*

4. *For every* $X \in \mathcal{X}$ *and* $(\leq nR) \in \mathcal{N}_{\leq}$*, replacing the variable* $\xi_{\neg A}^X$ *by the set* $L_{(\leq nR)}^X := \left(\bigcap_{w \in V_X} w^{-1} \cdot (\bigcup_{m \leq n} U_{(\leq nR)} \cup E_C \cdot \mathcal{R}^{-1})\right) \setminus \widehat{L}_\bot^X$ *solves Equation* $(\leq nR)$*.*

5. *For every* $X \in \mathcal{X}$*, replacing the variable* $\xi_\bot^X$ *by the set* $L_\bot^X$ *together with the assignments proposed in (1)–(4) solves Equation* $(\bot)$*.*

Observe that in the above conditions a finite alternative to $\widehat{L}_{j,\bot}$ is provided and that $\widehat{L}_{j,\bot}$ is subtracted from the other languages, thus producing polynomially large languages as solutions to the equations. This is an immediate consequence of [2], where it was shown that the above solution languages can be computed in polynomial time.

2.2.2 Solving the computation problem

Apart from testing solvability, [2] also proposes solutions to be assigned to the variables occurring in a matching problem and discusses their correctness and complexity in detail. The following lemma simmarizes the relevant results.

Lemma 13 *Solving matching problems*
Let $\mathcal{L} \in \{\mathcal{FL}_0, \mathcal{FL}_\bot, \mathcal{FL}_\neg, \mathcal{ALN}\}$*. Let* M *be an* $\mathcal{L}$*-matching problem. Then there exists an algorithm* $\textsc{match}_\mathcal{L}$ *with the following properties.*

1. $\textsc{match}_\mathcal{L}(M)$ *decides in polynomial time, whether the input matching problem* M *has a solution or not. If* M *is solvable, then* $\textsc{match}_\mathcal{L}(M)$ *in polynomial time in the size of* M *computes a solution* σ *which is minimal in regard to the ordering* $\sqsubseteq$ *on substitutions.*

2. MATCH$_\mathcal{L}$ *does not introduce atomic concepts or number restrictions which do not occur in the input matching problem* M.

3. MATCH$_\mathcal{L}$ *also accepts a finite system of* $\mathcal{L}$*-matching problems.*

PROOF. 1. It remains to be shown that computing the actual solution to a solvable matching problem also requires only polynomial time.

Solution in $\mathcal{ALN}$: To show this for $\mathcal{ALN}$-matching problems, we can refer to results provided in [2]. It is shown that the languages L_*^X used for the solvability test in Lemma 12 in fact are least solutions to the matching problem, which can be computed in polynomial time by finite automata. Therefore, a solution σ with the desired properties can be defined by assigning

$$X \longmapsto \forall L_\bot^X.\bot \sqcap \bigsqcap_{A\in\mathcal{C}} \forall L_A^X.A \sqcap \bigsqcap_{A\in\mathcal{C}} \forall L_{\neg A}^X.\neg A \\ \sqcap \bigsqcap_{(\leq nR)\in\mathcal{N}_\leq} \forall L_{(\leq nR)}^X.(\leq nR) \sqcap \bigsqcap_{(\geq nR)\in\mathcal{N}_\geq} \forall L_{(\geq nR)}^X.(\geq nR)$$

for every $X \in \mathcal{X}$. It can be shown that the assigned concept descriptions are of polynomial size in the size of the original matching problem. Since every role language of the form L_*^X can be represented by a treelike automaton [2], it takes only polynomial time to read off the languages represented by these automata, i.e. to actually return the computed solution.

Solutions in $\mathcal{FL}_\bot$ and $\mathcal{FL}_\neg$: For these sublanguages of $\mathcal{ALN}$, we must first restrict the languages used in the solvability test to finite ones. The rest of the argument then is identical to that for $\mathcal{ALN}$. For $\mathcal{FL}_\bot$ and $\mathcal{FL}_\neg$, [2] again provides us with the necessary results: Finite solution languages L_A^X can be obtained in the following way. Since $\widehat{L}_\bot^X$ can be represented by a treelike automaton [2] for every X, we read off a finite language $L_\bot^X$ with $L_\bot^X \cdot N_R^* = \widehat{L}_\bot^X$. Analogous to the languages defined for $\mathcal{ALN}$ in Lemma 12 we now define languages L_A^X by subtracting $\widehat{L}_\bot^X$ from $\widehat{L}_A^X$. We can then assign to the variable X the conjunction

$$X \longmapsto \forall L_\bot^X.\bot \sqcap \bigsqcap_{A\in\mathcal{C}} \forall L_A^X.A \sqcap \bigsqcap_{A\in\mathcal{C}} \forall L_{\neg A}^X.\neg A$$

for every $X \in \mathcal{X}$. Again, we yield a solution of polynomial size in polynomial time. The argument for $\mathcal{FL}_\bot$ is identical except for negated atomic concept missing in the concept descriptions finally assigned.

Solutions in $\mathcal{FL}_0$: Two arbitrary $\mathcal{FL}_0$-concept descriptions are equivalent if and only if their $\mathcal{FL}_0$-normal forms agree on all role languages involved. Therefore, infinite sets are not necessary at any step when solving matching problems. It can be shown that the solvability equation and solution languages for $\mathcal{FL}_0$ are equivalent to those for $\mathcal{FL}_\bot$ after removing any constructs relating to the bottom-concept or its role languages. The task of deciding solvability and computing solutions to a given matching problem then apparently is of of polynomial complexity.

2. It is shown in [2], that the solution specified above already has the desired property. Especially, this implies that the solution of a matching problem can be represented with the same set of role languages as the matching problem.

3. In Section 2.2, we have already seen that systems of matching equations can be represented by a single matching problem modulo subsumption which is polynomial in the size of the original system. Thus, with the results from (1) the proposition follows immediately. ∎

Hence, matching problems can be solved in polynomial time. Furthermore, we can find minimal solutions without introducing new atomic concepts or number restrictions and we can admit systems of matching problems as input. The following theorem summarizes the results obtained.

Theorem 14 *Let $\mathcal{L} \in \{\mathcal{FL}_0, \mathcal{FL}_\bot, \mathcal{FL}_\neg, \mathcal{ALN}\}$. Then there exists a polynomial time matching algorithm, called* $\text{MATCH}_\mathcal{L}$ *in the sequel, that computes the least matcher of a given system of $\mathcal{L}$-matching problems, if this system has a solution, and returns "fail" otherwise.*

2.3 Matching under side conditions

In this report, we focus on more general matching problems, those that allow for additional *side conditions*.

Definition 15 *A* subsumption condition *is of the form $X \sqsubseteq^? E$ where X is a concept variable and E is a pattern; a* strict subsumption condition *is of the form $X \sqsubset E$ where X and E are as above. A* side condition *is either a subsumption condition or a strict subsumption condition. The substitution σ satisfies the side condition $X \ \rho \ E$ for $\rho \in \{\sqsubseteq, \sqsubset\}$ iff $\sigma(X) \ \rho \ \sigma(E)$.*

A matching problem under side conditions *is a tuple* $M := \langle C \equiv^? D, S\rangle$, *where* $C \equiv^? D$ *is a matching problem and* S *is a finite set of side conditions. If the set* S *contains only subsumption conditions, then* M *is called* matching problem under subsumption conditions. *The substitution* σ *is a* solution *(*matcher*) of* M *iff it is a matcher of* $C \equiv^? D$ *that satisfies every side condition in* S.

In the next section, we will restrict the attention to matching problems under subsumption conditions. Section 4 then treats general matching problems under side conditions. There it is useful to distinguish between cyclic and acyclic sets of side conditions. In order to define matching problems under acyclic side conditions, we say that a variable X *directly depends* on a variable Y in S iff S contains a side condition $X\ \rho\ E$ such that Y occurs in E. If there are $n \geq 1$ variables $X_1, \ldots, X_n$ such that X_i directly depends on X_{i+1} in S $(1 \leq i \leq n-1)$, then we say that X_1 *depends* on X_n in S. The set of side conditions S is *cyclic* iff there is a variable X that depends on itself in S; otherwise, S is *acyclic*.

3 Matching under subsumption conditions

Let $\mathcal{L}$ be one of the DLs $\mathcal{FL}_\bot, \mathcal{FL}_\neg, \mathcal{ALN}$. We present a polynomial time algorithm that, given an $\mathcal{L}$-matching problems under subsumption conditions, returns a least matcher (w.r.t. the ordering $\sqsubseteq$ on substitutions) if the problem is solvable, and "fail" otherwise.

3.1 The algorithm handling subsumption conditions

In principle, the algorithm iterates the application of $\text{MATCH}_\mathcal{L}$ until a fixpoint is reached. However, the matcher computed in one step is used to modify the matching problem to be solved in the next step. Given an $\mathcal{L}$-matching problem under subsumption conditions $M := \langle C \equiv^? D, S\rangle$ and a substitution σ, we define

$$M_\sigma := \{C \equiv^? D\} \cup \{\sigma(X) \sqsubseteq^? E \mid X \sqsubseteq^? E \in S\}.$$

Recall that $\sigma(X) \sqsubseteq^? E$ abbreviates the matching problem $\sigma(X) \equiv^? \sigma(X) \sqcap E$. Thus M_σ is a system of $\mathcal{L}$-matching problems without side conditions, to which $\text{MATCH}_\mathcal{L}$ can be applied.

Algorithm 16 *Let $M := \langle C \equiv^? D, S\rangle$ be an $\mathcal{L}$-matching problem under subsumption conditions. Then, the algorithm* $\text{MATCH}^{\sqsubseteq}_{\mathcal{L}}(M)$ *works as follows:*

1. *$\sigma(X) := \bot$ for all variables X;*
2. *If* $\text{MATCH}_{\mathcal{L}}(M_\sigma)$ *returns "fail", then return "fail";*
   else if $\sigma \equiv \text{MATCH}_{\mathcal{L}}(M_\sigma)$*, then return σ;*
   else $\sigma := \text{MATCH}_{\mathcal{L}}(M_\sigma)$*; continue with 2.*

Let σ_0 denote the substitution defined in step 1 of the algorithm, and σ_t ($t \geq 1$) the matcher computed in the t-th iteration of Step 2. Note that σ_t is undefined if $\text{MATCH}_{\mathcal{L}}$ returns "fail" in the t-th iteration or if the algorithm has stopped before the t-th iteration.

To show that the algorithm is correct, we must show soundness, completeness, and termination, i.e., i) if the algorithm terminates and returns a substitution, then this substitution in fact solves the problem; ii) if the algorithm terminates and returns "fail", then there indeed is no solution; and iii) the algorithm halts on every input.

Soundness and completeness are addressed below in Section 3.2. Proving termination of the algorithm is more involved, and the exact argument depends on the DL $\mathcal{L}$ under consideration. The proof is given in Section 3.4. It depends on the so-called reduced normal form of concept descriptions, which has to be introduced beforehand in Section 3.3.

3.2 Soundness and Completeness

The following lemma proves soundness and completeness of Algorithm 16. The first two items establish a loop invariant.

Lemma 17 *Let $M := \langle C \equiv^? D, S\rangle$ be an $\mathcal{L}$-matching problem under subsumption conditions.*

1. *If σ_t is defined and τ is a solution of M, then $\sigma_t \sqsubseteq \tau$.*
2. *If σ_t, σ_{t+1} are defined, then $\sigma_t \sqsubseteq \sigma_{t+1}$.*
3. *If* $\text{MATCH}^{\sqsubseteq}_{\mathcal{L}}(M)$ *returns the substitution σ, then σ solves M (soundness).*
4. *If* $\text{MATCH}^{\sqsubseteq}_{\mathcal{L}}(M)$ *returns "fail", then M has no solution (completeness).*

PROOF. 1. Obviously, the claim is true for σ_0. Assume that $\sigma_t \sqsubseteq \tau$, and that σ_{t+1} is defined. To prove $\sigma_{t+1} \sqsubseteq \tau$, it is sufficient to show that τ solves M_{σ_t} since σ_{t+1} is the least solution of M_{σ_t}. Since τ solves M, we know that it solves $C \equiv^? D$ and that $\tau(X) \sqsubseteq \tau(E)$ for all $X \sqsubseteq^? E \in S$. The induction assumption $\sigma_t \sqsubseteq \tau$ implies $\sigma_t(X) \sqsubseteq \tau(X)$, and thus $\sigma_t(X) \sqsubseteq \tau(E)$, which shows that τ solves M_{σ_t}.

2. Obviously, $\sigma_0 \sqsubseteq \sigma_1$. Now assume that $\sigma_{t-1} \sqsubseteq \sigma_t$. Together with the fact that σ_t solves $M_{\sigma_{t-1}}$, this implies that σ_{t+1} solves the system $M_{\sigma_{t-1}}$. Since σ_t is the least solution of $M_{\sigma_{t-1}}$, we can conclude $\sigma_t \sqsubseteq \sigma_{t+1}$.

3. Assume that $\sigma = \sigma_t$. By definition of $\text{MATCH}^{\sqsubseteq}_{\mathcal{L}}$, $C \equiv \sigma_t(D)$. It remains to show that σ_t solves the side conditions. We know that $\sigma_t \equiv \sigma_{t+1}$ and σ_{t+1} solves M_{σ_t}. Thus, $\sigma_t(X) \sqsubseteq \sigma_{t+1}(E) \equiv \sigma_t(E)$ for every $X \sqsubseteq^? E \in S$.

4. Assume that $\text{MATCH}^{\sqsubseteq}_{\mathcal{L}}(M)$ returns "fail," and that σ_t is the last substitution computed by the algorithm. Now assume that τ solves M. As in the proof of 1. we can show that τ solves M_{σ_t}. Consequently, M_{σ_t} is solvable, and thus $\text{MATCH}(M_{\sigma_t})$ returns the least matcher of this system, in contradiction to the assumption that $\text{MATCH}^{\sqsubseteq}_{\mathcal{L}}(M)$ returns "fail" in this step of the iteration. ∎

3.3 Reduced normal forms

Role languages occurring in concept descriptions may contain redundant words, i.e., words that, when removed, yield equivalent concept descriptions. For instance, in $\mathcal{FL}_\bot$ it holds that: i) since $\forall w.\bot \sqsubseteq \forall wv.\bot$ for every $w, v \in N_R^*$, we can require $U_\bot$ to be prefix-free, i.e., $w, wv \in U_\bot$ implies $v = \varepsilon$; and ii) since $\forall w.\bot \sqsubseteq \forall wv.A$, we can require $U_A \cap (U_\bot \cdot N_R^*) = \emptyset$. A normal form satisfying these conditions is called *reduced normal form*. A formal definition of reduced normal forms for concept descriptions in $\mathcal{FL}_\bot$, $\mathcal{FL}_\neg$, and $\mathcal{ALN}$ is provided in Section 3.3.2. In preparation, we discuss some properties of so-called *prefix-free* formal languages.

3.3.1 Prefix free languages

We define prefix free languages as a specialization of formal languages [10] by introducing a unary function to make a given formal language prefix free.

Definition 18 *Prefix free languages*

$$\begin{aligned} pf\colon \mathfrak{P}(N_R^*) &\to \mathfrak{P}(N_R^*) \\ L &\mapsto L \setminus (L{\cdot}N_R^+) \end{aligned}$$

A language $U \subseteq N_R^*$ is called prefix free *if and only if $U = pf(U)$.*

Intuitively, $pf(L)$ for every word $w \in L$ removes all nontrivial continuations of w. The result is that for every word $w \in pf(L)$, all nontrivial prefixes of w are missing in $pf(L)$. To examine the properties of prefix free sets in greater detail, we must first introduce an appropriate order over finite languages. The definition of multiset orders is taken from [4], where their properties are discussed in depth. However, we employ multiset orders over formal languages and do not need to introduce multisets, which generalize the notion of sets by admitting multiple occurrences of elements.

Definition 19 *Multiset order for finite languages*
Define $(\succ)$ as a multiset order with $(>_{pr})$ on N_R^*. Thus, for finite languages $U, V \subseteq N_R^*$ it holds that $V \succ U$ if and only if there exist finite languages $X, Y \subseteq N_R^*$ such that:

1. $\emptyset \neq X \subseteq V$
2. $U = (V \setminus X) \cup Y$
3. $\forall y \in Y \exists x \in X\colon x <_{pr} y$

According to the definition, finite languages U and V are in prefix order, i.e. $U \succ V$, if and only if U can be transformed into V by performing a modification of the following type one or more times: remove a word u from U and replace it by a finite number of words from $\{u\}{\cdot}N_R^+$. Thus, u is replaced by a finite number of (nontrivial) continuations of u. Note that in this modification, u may be removed without substituting any words. This is allowed because in the definition above, the language Y may be empty. The following example illustrates this.

Example 20 Multiset order
Let $N_R := \{a, b, c\}$. Then $\{a, ab, c\} \succ \{ab, ac, caa, cab, ccc\}$. The definition of the multiset order is satisfied by taking $X := \{a, c\}$ and $Y := \{ac, caa, cab, ccc\}$. On the other hand, we also obtain $\{a, ab, c\} \succ \{ca\}$ by

taking $X := \{a, ab, c\}$ and $Y := \{ca\}$. Observe that the relation $U \succ V$ does not imply an obvious relation for the cardinality of the languages or for the length of the longest word contained in them.

The multiset order can be used to simplify comparing the N_R^*-closure of two given languages. This is addressed by the following lemma.

Lemma 21 *N_R^*-closures and prefix free languages*
Let $U, V \subseteq N_R^*$ be finite languages over N_R. Then,
1. $U \cdot N_R^* = pf(U) \cdot N_R^*$
2. $U \cdot N_R^* \subset V \cdot N_R^*$ *iff* $pf(U) \prec pf(V)$
3. $U \cdot N_R^* = V \cdot N_R^*$ *iff* $pf(U) = pf(V)$.

Proof. For the sake of brevity, denote $pf(U)$ by U' throughout this lemma. Analogously, denote $pf(V)$ by V'.

1. Since U' is a subset of U and since the sets on both sides of the equation are N_R^*-closed, it is sufficient to show that $U \setminus U'$ is a subset of $U' \cdot N_R^*$. Thus, consider $w \in U \setminus U'$. Then, by definition of prefix free sets, $w \in U \cdot N_R^+$. This implies, that in U there exists a word $u \in U$ of minimal length and a word $v \in N_R^+$ so that $w = uv$. Consequently, $u \notin U \cdot N_R^+$, because in this case the length of u would not be minimal. So we have $u \in U'$, implying that $w = uv \in U' \cdot N_R^*$.

2. ("$\Leftarrow$") If $U' \prec V'$ then, by Definition 18, there exist finite sets $X, Y \subseteq N_R^*$ with:

   (a) $\emptyset \neq X \subseteq V'$

   (b) $U' = (V' \setminus X) \cup Y$

   (c) $\forall y \in Y \exists x \in X : x <_{pr} y$.

   We first prove the non-strict version of the claim, i.e. $U \cdot N_R^* \subset V \cdot N_R^*$, and then show that the inclusion is strict.

   Nonstrict inclusion: As U' equals $(V' \setminus X) \cup Y$, it is sufficient to show that $Y \subseteq V' \cdot N_R^*$. Thus, consider an arbitrary $y \in Y$. Because of Property 3 of multiset orders it holds that there is an $x \in X \subseteq V'$ so that $x <_{pr} y$. Being less in regard to the prefix order implies, that we

obtain $y = xw$ for an appropriate $w \in N_R^*$. Since $x \in V'$, this yields $y = xw \in V' \cdot N_R^*$, completing the proof.

Strictness of the inclusion: Consider an arbitrary $x \in X \subseteq V'$. According to Property 1 of multiset orders, such an x in fact exists. x is no element of $(V' \setminus X)$, because V' is prefix free and thus contains no prefix of x. Now, if $x \in Y$ then Property 3 demands that there is another word $x' \in X$ so that $x' <_{pr} X$. This would be a contradiction to V' being prefix free, and therefore: $x \notin U' \cdot N_R^*$.

("$\Rightarrow$") Assume $U' \cdot N_R^* \subset V' \cdot N_R^*$. Taking advantage of (1), this is equivalent to the original proposition. Define finite languages X, Y in the following way: $X := V' \setminus U'$ and $Y := U' \setminus V'$. We will show that these languages match conditions 1, 2, and 3 stated in the definition of multiset orders.

Property 1: Trivial. X is obviously defined as a subset of V'. If X is empty, then $U' \supseteq V'$, which would rule out $U' \cdot N_R^* \subset V' \cdot N_R^*$, conflicting with the assumption above.

Property 2: Applying the definitions of X and y, we can expand $(V' \setminus X) \cup Y$ to the expression $(V' \setminus (V' \setminus U')) \cup U' \setminus V'$, which simplifies to $(U' \cap V') \cup U' \setminus V'$. This is obviously equivalent to U'.

Property 3: Consider an arbitrary $y \in Y = U' \setminus V'$. From Property 2 of the multiset order we know that $Y \subseteq U' \subset V' \cdot N_R^*$. Thus, there are words $v \in V'$ and $w \in N_R^*$ such that $y = vw$. This implies $w \neq \varepsilon$, because otherwise y, being equal to v, would be an element of V'. If w is not empty, then v and y are in prefix relation: $v <_{pr} y$. Consequently, v is no element of U', because then U' would not be prefix free. This implies $v \in V' \setminus U'$, which by definition is equivalent to $v \in X$.

3. ("$\Leftarrow$") This is an immediate consequence of (1). If U' equals V', then obviously $U' \cdot N_R^* = V' \cdot N_R^*$, which implies $U \cdot N_R^* = V \cdot N_R^*$, as shown in (1). ("$\Rightarrow$") Reversely assume that $U' \cdot N_R^* = V' \cdot N_R^*$. According to (1), this is equivalent to the original proposition. It is sufficient to prove the inclusion $U' \subseteq V'$, since the reverse inclusion follows by symmetry.

   Consider an arbitrary $u \in U'$. According to the above assumption we have $U' \subseteq V' \cdot N_R^*$, which implies the existence of words $v \in V'$ and $w \in N_R^*$ with $u = vw$. It reversely holds that $V' \subseteq U' \cdot N_R^*$, again implying words $u' \in U'$ and $w' \in N_R^*$ so that $v = u'w'$. Therefore, we

yield $u = vw = u'w'w$. This implies $w = w' = \varepsilon$, because otherwise U' would not be prefix free, containing a prefix of u. With w equal to ε, we finally obtain $u \in V'$, which had to be shown. ∎

Observe, that the N_R^*-closure of a language L is uniquely defined by the prefix free version of L. We can also use prefix free languages to guarantee a suffix condition when representing the left quotient of the N_R^*-closure of a language:

Lemma 22 *Left quotients and prefix free languages*
Let $U \subseteq N_R^*$ be a finite language and let $w \in N_R^*$. Then there exists a finite language $L \subset N_R^*$ such that,
1. $L{\cdot}N_R^* = w^{-1}(U{\cdot}N_R^*)$ and
2. L is prefix free and
3. L contains only suffixes of words in U.

Proof. According to [2], there exists a finite language L' with $L'{\cdot}N_R^* = w^{-1}{\cdot}(U{\cdot}N_R^*)$. Due to Lemma 21, we know that this also holds for $L := pf(L')$. We now show that L contains only suffixes of U, which is sufficient for our claim. Assume a word $v \in L$, which is no suffix of any word in U. Observe, that this implies $v \neq \varepsilon$ because otherwise v would be a trivial suffix of any word in U. By definition of L, we know that v is an element of $w^{-1}{\cdot}(U{\cdot}N_R^*)$. Thus, there exists a word $u \in U$ and a word $x \in N_R^+$ such that $wv = ux \in U{\cdot}N_R^*$. We exclude $x = \varepsilon$, because then v would be a suffix of u. Denote by s the last character of v, i.e. take $s \in N_R$ and $v' \in N_R^*$ such that $v = v's$. Analogously, let $x = x's$ for an appropriate $x' \in N_R^*$. Then we can conclude that $v' \in L$, because $wv' = ux'$ is an element of $U{\cdot}N_R^*$. This implies a contradiction to the language L being prefix free. ∎

3.3.2 Reduced normal forms

In $\mathcal{FL}_\bot$, $\mathcal{FL}_\neg$, and $\mathcal{ALN}$, equivalent concept descriptions in normal form can differ in size to an arbitrary extent. For instance, $\forall\{\varepsilon\}.\bot \sqcap \forall U_A.A$ is equivalent to $\forall\{\varepsilon\}.\bot$ for every role language U_A. For a simplified proof of termination, we require normal forms which impose stronger limitations on the size of concept descriptions equivalent to or subsuming each other. For this purpose, *reduced* normal forms for $\mathcal{FL}_\bot$, $\mathcal{FL}_\neg$, and $\mathcal{ALN}$ are introduced.

Reduced normal forms for $\mathcal{FL}_\bot$

The reduced normal form of $\mathcal{FL}_\bot$-concept descriptions is defined by specifying an operation to transform an arbitrary $\mathcal{FL}_\bot$-concept description into its corresponding reduced normal form.

Definition 23 *Let C be an $\mathcal{FL}_\bot$-concept description in U-labeled normal form. Its corresponding $U^\downarrow$-*labeled reduced normal form *$C^\downarrow$ is defined as follows:*

$$C^\downarrow := \forall U_\bot^\downarrow.\bot \sqcap \prod_{A \in \mathcal{C}} \forall U_A^\downarrow.A$$

where for $A \in \mathcal{C}$:

$$U_\bot^\downarrow := pf(U_\bot)$$
$$U_A^\downarrow := U_A \setminus U_\bot^\downarrow \cdot N_R^*$$

A concept description C is called reduced*, if C is in normal form and if it coincides with $C^\downarrow$ in every occurring role language. The notion of reduction can be extended to substitutions. For a substitution σ, the reduced substitution $\sigma^\downarrow$ is established by defining $\sigma^\downarrow(X) := \sigma(X)^\downarrow$ for every variable X in the domain of σ.*

The above definition implies as immediate consequences the following simple properties, which are stated without proof.

Corollary 24 *Properties*
Let C be an $\mathcal{FL}_\bot$-concept descriptions in U-labeled normal form. Then,
1. $U_\bot^\downarrow$ is prefix free and $U_A^\downarrow \cap U_\bot^\downarrow \cdot N_R^*$ is empty for every $A \in \mathcal{C}$
2. The reduced normal form $C^\downarrow$ can be computed in polynomial time in the size of C.

It will be particularly useful that there is no overlap between the role language $U_\bot^\downarrow$ and the N_R^*-closure of $U_A^\downarrow$. The role languages for $C^\downarrow$ can be constructed in polynomial time using treelike automata, for which the complement and the N_R^*-closure can be computed in linear time. It also takes only polynomial time to make a given finite role language prefix free. The ability to compute reduced normal forms in polynomial time will not be required in the remainder of this chapter. Nevertheless, it might be an important property in the context of presenting the output of matching algorithms in a compact way.

Recall that *pf* in Section was defined to make the input language prefix free. The purpose of reduced normal forms is to simplify the characterization of subsumption and equivalence. One can see that in the above definition exactly those languages are made prefix free, whose N_R^*-closure appears in the characterization of the subsumption proposed in Lemma 1. Furthermore, by subtracting the N_R^*-closure from the other role languages, we make sure that all unions in the characterising conditions are disjoint. In the next lemma we will see that this is sufficient to reduce equivalence to equality.

Lemma 25 *Properties*
Let B, C, D be $\mathcal{FL}_\bot$-concept descriptions. Let B be in W-labeled normal form, let C be in U-labeled reduced normal form, and D in V-labeled reduced normal form. Then,
1. $B \equiv B^\downarrow$
2. $C \equiv D$ iff $U_H = V_H$ for all $H \in \{\bot\} \cup \mathcal{C}$
3. $C \sqsubset D$ iff one of the following conditions holds:
(a) $U_\bot \succ V_\bot$ and $V_A \subseteq U_A \cup U_\bot \cdot N_R^*$ for all $A \in \mathcal{C}$
(b) $U_\bot = V_\bot$ and $U_A \supseteq V_A$ for all $A \in \mathcal{C}$ and there exists an $A \in \mathcal{C}$ with $U_A \supset V_A$.

Proof. 1. We have seen in Lemma 1 that it is sufficient to prove the following two conditions:

- $W_\bot \cdot N_R^* = W_\bot^\downarrow \cdot N_R^*$
- $W_A \cup W_\bot \cdot N_R^* = W_A^\downarrow \cup W_\bot^\downarrow \cdot N_R^*$ for all $A \in \mathcal{C}$.

The first condition was shown as a property of prefix free languages in Lemma 21. For the second condition, we can therefore conclude for every A that $W_A^\downarrow \cup W_\bot^\downarrow \cdot N_R^*$ is equal to $W_A^\downarrow \cup W_\bot \cdot N_R^*$. We may add $(W_A \cap W_\bot \cdot N_R^*)$, which is a subset of $W_\bot \cdot N_R^*$, thus yielding $W_A^\downarrow \cup (W_A \cap W_\bot \cdot N_R^*) \cup W_\bot \cdot N_R^*$. According to the definition of reduced normal forms, W_A equals $W_A^\downarrow \cup (W_A \cap W_\bot \cdot N_R^*)$. Therefore, $W_A^\downarrow \cup (W_A \cap W_\bot \cdot N_R^*) \cup W_\bot \cdot N_R^*$ equals $W_A \cup W_\bot \cdot N_R^*$.

2. ("$\Leftarrow$") is trivial. ("$\Rightarrow$") Assume $C \equiv D$. Due to Lemma 1, this again is equivalent to $U_\bot \cdot N_R^* = V_\bot \cdot N_R^*$ and $U_A \cup U_\bot \cdot N_R^* = V_A \cup V_\bot \cdot N_R^*$ for all $A \in \mathcal{C}$. Since C and D are assumed to be reduced, this implies $U_\bot = V_\bot$, according to the properties of prefix free sets. Furthermore, due to

the definition of reduced normal forms, U_A and $U_\perp \cdot N_R^*$ are disjoint for every A. The same applies to V_A and $V_\perp \cdot N_R^*$. Therefore, $U_A \cup U_\perp \cdot N_R^* = V_A \cup V_\perp \cdot N_R^*$ implies $U_A = V_A$ for all A, which was to be shown.

3. ("$\Rightarrow$") Assume $C \sqsubset D$. Then we again have $U_\perp \cdot N_R^* \supseteq V_\perp \cdot N_R^*$. We distinguish two cases depending on whether the inclusion is strict or not.

   Strict inclusion: If $U_\perp \cdot N_R^* \supset V_\perp \cdot N_R^*$, we can infer $U_\perp \succ V_\perp$, as shown in Lemma 21. We know from the characterization of the subsumption that $U_A \cup U_\perp \cdot N_R^* \supseteq V_A \cup V_\perp \cdot N_R^*$ for all $A \in \mathcal{C}$. We may remove $V_\perp \cdot N_R^*$ from the right-hand side of the inclusion, yielding the assertion for case (a), $V_A \subseteq U_A \cup U_\perp \cdot N_R^*$.

   Equality: If $U_\perp \cdot N_R^* = V_\perp \cdot N_R^*$, we have $U_\perp = V_\perp$, because C and D are reduced and therefore $U_\perp$ and $V_\perp$ are prefix free. The subsumption $C \sqsubset D$ also implies that $U_A \cup U_\perp \cdot N_R^* \supseteq V_A \cup V_\perp \cdot N_R^*$ for every A. The unions on both sides of the inclusion are disjoint, as stated in Corollary 24. Taking advantage of the equality of $U_\perp \cdot N_R^*$ and $V_\perp \cdot N_R^*$, we obtain $U_A \supseteq V_A$ for every $A \in \mathcal{C}$. There has to be one A with a strict inclusion $U_A \supset V_A$. Otherwise, C and D would agree on all role languages, implying equivalence as shown in (2). Thus, the assertion for case (b) holds.

   ("$\Leftarrow$") We have to show that both conditions for the subsumption as stated in Lemma 1 are met. Assuming case (b), this can be seen immediately. Consider case (a). If $U_\perp \succ V_\perp$ holds, the first condition for the subsumption is met as a consequence of Lemma 21, obtaining $U_\perp \cdot N_R^* \supset V_\perp \cdot N_R^*$. We have assumed that $V_A \subseteq U_A \cup U_\perp \cdot N_R^*$. Adding $V_\perp \cdot N_R^*$ on both sides yields $V_A \cup V_\perp \cdot N_R^* \subseteq U_A \cup U_\perp \cdot N_R^* \cup V_\perp \cdot N_R^*$. As $V_\perp \cdot N_R^*$ is a subset of $U_\perp \cdot N_R^*$, this is equivalent to $V_A \cup V_\perp \cdot N_R^* \subseteq U_A \cup U_\perp \cdot N_R^*$. Thus, the second condition of the subsumption is met for every $A \in \mathcal{C}$. We yield strict subsumption $C \sqsubset D$, because otherwise $U_\perp = V_\perp$. ∎

Reduced normal forms for $\mathcal{FL}_\neg$

For $\mathcal{FL}_\neg$, we follow the same pattern as seen in the previous section. Firstly, the reduction operation is expanded in such a way that it works with negated atomic concepts as well.

Definition 26 *Reduced normal form*
Let C be an $\mathcal{FL}_\neg$-concept description in U-labeled normal form. Like in Definition 23, define its corresponding reduced normal form $C^\downarrow$ by modifying the role languages:

$$C^\downarrow := \forall U_\bot^\downarrow.\bot \sqcap \bigsqcap_{A\in\mathcal{C}} \forall U_A^\downarrow.A \sqcap \bigsqcap_{A\in\mathcal{C}} \forall U_{\neg A}^\downarrow.\neg A$$

where for $A \in \mathcal{C}$:

$$U_\bot^\downarrow := pf\,(U_\bot \cup \bigcup_{A\in\mathcal{C}} U_A \cap U_{\neg A})$$
$$U_A^\downarrow := U_A \setminus U_\bot^\downarrow \cdot N_R^*$$

Again, if C is reduced, then its role languages are identical to those of $C^\downarrow$. We extend the notion of reduction to substitutions as in Definition 23.

Observe that in this definition the role language $U_\bot$ referring to the bottom concept may increase in size when normalized. Contrary to $\mathcal{FL}_\bot$, it is possible to have inconsistencies without involving the bottom concept. The reduced normal form for $\mathcal{FL}_\neg$ aims at making all implicit inconsistencies explicit, i.e. whenever an expression like $\forall w.(A \sqcap \neg A)$ occurs, w is removed from the role languages referring to A and $\neg A$ and is included in the language for the bottom concept. The definition of excluding words again implies some inportant properties, which are stated below without proof.

Corollary 27 *Properties*
Let C be an $\mathcal{FL}_\neg$-concept descriptions in U-labeled normal form. Then,
1. $U_\bot^\downarrow$ is prefix free and $U_\bot^\downarrow = (U_\bot^\downarrow)^\frown$.
2. $U_H^\downarrow \cap (U_\bot^\downarrow)^\frown \cdot N_R^*$ is empty for every $H \in \mathcal{C} \cup \{\neg A \mid A \in \mathcal{C}\}$.
3. $U_A^\downarrow \cap U_{\neg A}^\downarrow$ is empty for every $A \in \mathcal{C}$.
4. The reduced normal form $C^\downarrow$ can be computed in polynomial time in the size of C.

Since $(U_\bot^\downarrow)^\frown$ is defined as $U_\bot^\downarrow \cup \bigcup_{A\in\mathcal{C}}(U_A^\downarrow \cap U_{\neg A}^\downarrow)$, the above assertions are readily obtained from the definition of reduced normal forms. Computing the reduced normal form in polynomial time can again be accomplished by employing treelike automata. By virtue of these properties, we again achieve the desired simplification for the characterization of the subsumption. In the next lemma it is shown that the results obtained for $\mathcal{FL}_\neg$ resemble those for $\mathcal{FL}_\bot$ seen in the last section.

Lemma 28 *Properties*
Let B, C, D be $\mathcal{FL}_\neg$-concept descriptions. Let B be in W-labeled normal form, let C be in U-labeled reduced normal form, and D in V-labeled reduced normal form. Let $\mathcal{H} := \mathcal{C} \cup \{\neg A \mid A \in \mathcal{C}\}$. Then,
1. $B \equiv B^{\downarrow}$
2. $C \equiv D$ *iff* $U_H = V_H$ *for all* $H \in \{\bot\} \cup \mathcal{H}$
3. $C \sqsubset D$ *iff one of the following conditions holds:*
   (a) $U_\bot \succ V_\bot$ *and* $V_H \subseteq U_H \cup U_\bot \cdot N_R^*$ *for all* $H \in \mathcal{H}$
   (b) $U_\bot = V_\bot$ *and* $U_H \supseteq V_H$ *for all* $H \in \mathcal{H}$ *and there exists an* $H \in \mathcal{H}$ *with* $U_A \supset V_A$.

Proof. 1. Due to Lemma 3, it is sufficient to prove that the following conditions hold:

- $\widehat{W_\bot} \cdot N_R^* = (W_\bot^{\downarrow})^\frown \cdot N_R^*$
- $W_H \cup \widehat{W_\bot} \cdot N_R^* = W_H^{\downarrow} \cup (W_\bot^{\downarrow})^\frown \cdot N_R^*$ for all $H \in \mathcal{H}$.

First condition: By definition, $\widehat{W_\bot} \cdot N_R^*$ equals $(W_\bot \cup \bigcup_{A \in \mathcal{C}} W_A \cap W_{\neg A}) \cdot N_R^*$, which is equivalent to the prefix free version $pf(W_\bot \cup \bigcup_{A \in \mathcal{C}} W_A \cap W_{\neg A}) \cdot N_R^*$, as we have seen in Lemma 21. Applying the definition of reduced normal forms, this is equivalent to $W_\bot^{\downarrow} \cdot N_R^*$. The intersection of $W_A^{\downarrow}$ and $W_{\neg A}^{\downarrow}$ is empty for every $A \in \mathcal{C}$, as stated in Corrolary 27. We may therefore add $(\bigcup_{A \in \mathcal{C}} W_A^{\downarrow} \cap W_{\neg A}^{\downarrow})$ to the expression, so that we end up with $(W_\bot^{\downarrow} \cup \bigcup_{A \in \mathcal{C}} W_A^{\downarrow} \cap W_{\neg A}^{\downarrow}) \cdot N_R^*$. This equals $(W_\bot^{\downarrow})^\frown \cdot N_R^*$, as can be verified from the definition.

Second condition: Taking advantage of (1), we can see that $W_H^{\downarrow} \cup (W_\bot^{\downarrow})^\frown \cdot N_R^*$ is equal to $W_H^{\downarrow} \cup \widehat{W_\bot} \cdot N_R^*$ for every $H \in \mathcal{H}$. We may add a subset of the second term, yielding the expression $W_H^{\downarrow} \cup (W_H \cap \widehat{W_\bot} \cdot N_R^*) \cup \widehat{W_\bot} \cdot N_R^*$. The language $W_H^{\downarrow}$ is defined as $W_H \setminus W_\bot^{\downarrow} \cdot N_R^*$. As stated in Corollary 27, this equals $W_H \setminus (W_\bot^{\downarrow})^\frown \cdot N_R^*$, which in (1) is shown equal to $W_H \setminus \widehat{W_\bot} \cdot N_R^*$. The expression $W_H^{\downarrow} \cup (W_H \cap \widehat{W_\bot} \cdot N_R^*) \cup \widehat{W_\bot} \cdot N_R^*$ can therefore be simplified to $W_H \cup \widehat{W_\bot} \cdot N_R^*$, yielding the desired result.

2. ("$\Leftarrow$") Trivial. ("$\Rightarrow$") According to Corollary 27, we have $\widehat{U_\bot} = U_\bot$ and $\widehat{V_\bot} = V_\bot$. When replacing these role languages, the proposition and the characterization of the subsumption are analogous to those for

$\mathcal{FL}_\bot$. Consequently, the proof is identical to (2) in the previous Lemma 25.

3 Again, taking into account that $\widehat{U}_\bot = U_\bot$ and $\widehat{V}_\bot = V_\bot$, we can prove the proposition in the same way as seen in (3) in the previous lemma. ∎

One can see that the additional complexity of concept descriptions in $\mathcal{FL}_\neg$ is hidden in the reduced normal form.

Reduced normal forms for $\mathcal{ALN}$

When introducing reduced normal forms for $\mathcal{ALN}$-concept descriptions, we have to face two additional problems. Firstly, the set of all inconsistencies explicitly occurring or implicitly included in a concept description cannot be obtained in such a straightforward way as in the previous two logics. Secondly, we also have to cope with number restrictions. In the following definition, we utilize the notion of excluding words, which have been introduced in Definition 2 in the context of the characterization of subsumption.

Definition 29 *Reduced normal form*
Let C be an $\mathcal{ALN}$-concept description in U-labeled normal form. Define the reduced normal form of C by modifying its role languages. It has been stated in [2] that there exists a finite language U_{E_C} with $E_C = U_{E_C} \cdot N_R^*$. Using this language, define $C^\downarrow$ as:

$$C^\downarrow := \forall U_\bot^\downarrow.\bot \sqcap \bigsqcap_{A \in \mathcal{C}} \forall U_A^\downarrow.A \sqcap \bigsqcap_{A \in \mathcal{C}} \forall U_{\neg A}^\downarrow.\neg A$$
$$\sqcap \bigsqcap_{(\geq nR) \in \mathcal{N}_\geq} \forall U_{(\geq nR)}^\downarrow.(\geq nR) \sqcap \bigsqcap_{(\leq nR) \in \mathcal{N}_\leq} \forall U_{(\leq nR)}^\downarrow.(\leq nR)$$

where for $A \in \mathcal{C}$, $(\leq nR) \in \mathcal{N}_\leq$, and $(\geq nR) \in \mathcal{N}_\geq$:

$$U_\bot^\downarrow := pf(U_{E_C})$$
$$U_A^\downarrow := U_A \setminus E_C$$
$$U_{\neg A}^\downarrow := U_{\neg A} \setminus E_C$$
$$U_{(\geq nR)}^\downarrow := \bigcup_{m \geq n} U_{(\geq mR)} \setminus E_C$$
$$U_{(\leq nR)}^\downarrow := \bigcup_{m \leq n} U_{(\leq mR)} \setminus E_C \cdot R^{-1}$$

Analogous to the previous cases, the notion of reduction is extended to substitutions.

In spite of the formally more complex definition, the objective of the above normal form is equal to those seen before. Inconsistencies are made explicit by augmenting the role language of the bottom concept and the other role languages are minimized as much as possible. Observe that the reduced role language $U_\perp^\downarrow$ in fact is well-defined, because for languages of the form $L{\cdot}N_R^*$ the set $pf(L)$ is unique. The definition of reduced normal forms again implies some basic properties, which are presented in the corollary below.

Corollary 30 *Properties*
Let C be an $\mathcal{ALN}$-concept descriptions in U-labeled normal form. Then,
1. $U_\perp^\downarrow$ is prefix free

2. $U_H^\downarrow \cap E_{C^\downarrow}$ is empty for every $\mathcal{H} := \mathcal{C} \cup \{\neg A \mid A \in \mathcal{C}\} \cup \mathcal{N}_\geq$.
Furthermore, $U_{(\leq nR)}^\downarrow \cap E_{C^\downarrow}{\cdot}R^{-1}$ is empty for every $(\leq nR) \in \mathcal{N}_\leq$

3. $\bigcup_{m \geq n} U_{(\geq mR)}^\downarrow = U_{(\geq nR)}^\downarrow$ for all $(\geq nR) \in \mathcal{N}_\geq$ and analogously for all $(\leq nR) \in \mathcal{N}_\leq$

4. The reduced normal form $C^\downarrow$ can be computed in polynomial time in the size of C.

As stated in [2], a role language U_{E_C} with $E_C = U_{E_C}{\cdot}N_R^*$ can be computed in polynomial time. With the aid of treelike automata, it therefore takes only polynomial time to compute the reduced normal form of C. In order to examine the properties of our normal form closer, we have to procure a better characterization for the set of excluding words from [12]. The following definition is necessary in preparation.

Definition 31 *Required words*
Let C be an $\mathcal{ALN}$-concept description in U-labeled normal form. Let v and v' be words over N_R. Let $|v| =: m$ and $|vv'| =: n$ and $v' =: R_{m+1} \ldots R_n$. Then vv' is required by C starting from v iff for all $i \in \{m, \ldots, n-1\}$ there exist positive integers $k_{i+1} \geq 1$ such that $vR_{m+1} \ldots R_i \in U_{(\geq k_{i+1} R_{i+1})}$.

Intuitively, the continuation vv' is required by a concept description C starting from v, iff there is a sequence of $(\geq)$-number restrictions for every prefix of vv' between v and vv' demanding the presence of the respective

following prefix. For example, assume $N_R := \{R, S\}$ and let $C := A \sqcap \forall\{RS, RSR\}.(\geq 1R) \sqcap \forall\{RSR\}.(\geq 2S)$. Then the words $RSRR$ and $RSRS$ are required by C starting from RS.

With the notion of required words we can characterize excluding words for $\mathcal{ALN}$-concept descriptions by the following lemma.

Lemma 32 *Characterization of excluding words*
Let C be an $\mathcal{ALN}$-concept description in U-labeled normal form. Let w be a word over N_R. Then $w \in E_C$ iff
1. there exists a prefix $v \in N_R^*$ of w and a word $v' \in N_R^*$ such that vv' is required by C starting from v and
   (a) $vv' \in U_\bot$, or
   (b) there is an atomic concept $A \in \mathcal{C}$ with $vv' \in U_A \cap U_{\neg A}$, or
   (c) there are number restrictions $(\geq lR) \in \mathcal{N}_\geq$ and $(\leq rR) \in \mathcal{N}_\geq$ such that $l > r$ and $v \in U_{(\geq lR)} \cap U_{(\leq rR)}$; or
2. there exists a prefix vR of w (with $v \in N_R^*, R \in N_R$) such that $v \in U_{(\leq 0R)}$.

Now we are set to examine reduced normal forms in detail. Before addressing the standard questions of correctness, equivalence, and subsumption, however, we first introduce one auxiliary result regarding the notion of excluding words, which will be required in Lemma 35. In the next lemma, it is shown that transforming a concept description into reduced normal forms does not change its properties in respect to required words.

Lemma 33 *Required words and reduced normal forms*
Let C be an $\mathcal{ALN}$-concept description in U-labeled normal form and let v, v' be words over N_R. Then, if vv' is required by $C^\downarrow$ starting from v then vv' is required by C starting from v.

Proof. To simplify the notation throughout this proof, denote $|v| =: s$, $|vv'| =: t$, and $vv' =: R_1R_2 \ldots R_t$. If vv' is required by $C^\downarrow$ starting from v, then by definition it holds for all $i \in \{s, \ldots, t-1\}$ that there exists a positive integer $k \geq 1$, so that $R_1 \ldots R_i \in U^\downarrow_{(\geq kR_{i+1})}$. By definition of reduced normal forms, this implies that $R_1 \ldots R_i \in \bigcup_{n \geq k} U_{(\geq nR_{i+1})} \setminus E_C$. No n under the union is smaller than k. Consequently, there exists an integer $k' \geq k$ so that $R_1 \ldots R_i$ is an element of $U_{(\geq k'R_{i+1})} \setminus E_C$. Obviously, we can include all the words subtracted by E_C, thus obtaining that $R_1 \ldots R_i \in U_{(\geq k'R_{i+1})}$. This is equivalent to vv' being required by C starting from v, which was to be shown ∎

A simplified characterization for the set of excluding words is now proposed for concept descriptions in reduced normal form. It is shown by the next lemma that only case (1a) of the characterization given in Lemma 32 is relevant for the reduced normal form of concept descriptions.

Lemma 34 *Excluding words and reduced normal forms*
Let C be an $\mathcal{ALN}$-concept description in U-labeled normal form. Let w be a word over N_R. Then, $w \in E_{C\downarrow}$ iff there exists a prefix $v \in N_R^*$ of w and a word $v' \in N_R^*$ with: vv' is required by $C^\downarrow$ starting from v and $vv' \in U_\bot$.

Proof. Consider a word $w \in E_{C\downarrow}$. It is sufficient to prove that the Cases (1b), (1c), or (2) specified in the characterization of $E_{C\downarrow}$ do not apply.

Case (1b): Then there exists a prefix $v \in N_R^*$ of w, a word $v' \in N_R^*$, and an atomic concept $A \in \mathcal{C}$, so that vv' is required by $C^\downarrow$ starting from v and $vv' \in (U_A^\downarrow \cap U_{\neg A}^\downarrow)$. Applying the definition of reduced normal forms, this implies that vv' is an element of $U_A \cap U_{\neg A}$, but no element of E_C. By Definition of the semantics of $\mathcal{ALN}$-concept descriptions, this implies $C \sqsubseteq \forall vv'.\bot$. As a consequence of Definition 2, this implies $vv' \in E_C$, in contradiction to the above finding that $vv' \notin E_C$.

Case (1c): Then we have an analogous word vv' and nonnegative numbers $l > r$ with $vv' \in U_{(\geq lR)} \cap U_{(\leq rR)}$. Again by definition of reduced normal forms, we conclude that vv' is an element of the intersection $\bigcup_{l' \geq l} U_{(\geq l'R)} \cap \bigcup_{r' \leq r} U_{(\leq r'R)}$, but it is not in E_C. Therefore, we can find integers $l' \geq l$ and $r' \leq r$ such that $vv' \in U_{(\geq l'R)} \cap U_{(\leq r'R)}$. Analogous to case (1b), the semantics of $\mathcal{ALN}$ then implies $C \sqsubseteq \forall vv'.\bot$. Due to Definition 2, this entails $vv' \in E_C$, contradicting the above statement.

Case (2): We prove that in the reduced normal form $C^\downarrow$ the role language $U_{(\leq 0R)}^\downarrow$ is empty for every atomic role $R \in N_R$. As 0 is the least nonnegative integer, for every atomic role $R \in N_R$ the definition of $U_{(\leq 0R)}^\downarrow$ can be simplified to $U_{(\leq 0R)} \setminus E_C{\cdot}R^{-1}$, omitting the union. Therefore, if $U_{(\leq 0R)}^\downarrow$ is not empty, it contains an element of $U_{(\leq 0R)}$. Thus, assume $w \in U_{(\leq 0R)}$ for a word w. According to the definition of number restrictions, this implies that w has no successors in regard to R. Consequently, $wR \in E_C$. Obviously, we can infer $w \in E_C{\cdot}R^{-1}$. In the definition of $U_{(\leq 0R)}^\downarrow$, the set $E_C{\cdot}R^{-1}$ is subtracted from the rest, implying $w \notin U^\downarrow{}_{(\leq 0R)}$. Case (2) does therefore not apply to $C^\downarrow$. ∎

The above result suggests a simpler proof of the correctness of the normal form. The standard questions, correctness and modified characterizations for equivalence and subsumption, are addressed in the next lemma.

Lemma 35 *Properties*
Let B, C, D be $\mathcal{ALN}$-concept descriptions. Let B be in W-labeled normal form, let C be in U-labeled reduced normal form, and D in V-labeled reduced normal form. Let $\mathcal{H} := \mathcal{C} \cup \{\neg A \mid A \in \mathcal{C}\} \cup \mathcal{N}_{\leq} \cup \mathcal{N}_{\geq}$. Then,
1. $B \equiv B^{\downarrow}$

2. $C \equiv D$ *iff* $U_H = V_H$ *for all* $H \in \{\bot\} \cup \mathcal{H}$

3. $C \sqsubset D$ *iff one of the following conditions holds:*
(a) $U_\bot \succ V_\bot$ *and* $V_H \subseteq U_H \cup U_\bot \cdot N_R^*$ *for all* $H \in \mathcal{H} \setminus \mathcal{N}_{\leq}$ *and* $V_H \subseteq U_H \cup U_\bot \cdot N_R^* \cup U_\bot \cdot R^{-1}$ *for all* $(\leq nR) := H \in \mathcal{N}_{\leq}$
(b) $U_\bot = V_\bot$ *and* $U_H \supseteq V_H$ *for all* $H \in \mathcal{H}$ *and there exists an* $H \in \mathcal{H}$ *with* $U_A \supset V_A$.

Proof. 1. In Lemma 4, equivalence of $\mathcal{ALN}$-concept decriptions was is characterized by the following conditions. For $A \in \mathcal{C}$, $(\leq mR) \in \mathcal{N}_{\leq}$, and $(\geq mR) \in \mathcal{N}_{\geq}$:

(a) $E_{B^\downarrow} = E_B$
(b) $W_A^\downarrow \cup E_{B^\downarrow} = W_A \cup E_B$
(c) $W_{\neg A}^\downarrow \cup E_{B^\downarrow} = W_{\neg A} \cup E_B$
(d) $\bigcup_{m \geq n} W_{(\geq mR)}^\downarrow \cup E_{B^\downarrow} = \bigcup_{m \geq n} W_{(\geq mR)} \cup E_B$
(e) $\bigcup_{m \leq n} W_{(\leq mR)}^\downarrow \cup E_{B^\downarrow} \cdot R^{-1} = \bigcup_{m \leq n} W_{(\leq mR)} \cup E_B \cdot R^{-1}$

Condition 1: Prove $E_{B^\downarrow} \subseteq E_B$. Consider an arbitrary $w \in E_{B^\downarrow}$. Due to the simplified characterization of exclusion for reduced normal forms, this implies that there exists a prefix $v \in N_R^*$ of w and a word $v' \in N_R^*$ such that vv' is required by B starting from v and $vv' \in W_\bot^\downarrow$. According to Definition 26, this implies that vv' is in $pf(W_{E_B}) \subseteq E_B$ for an appropriate finite language W_{E_B} with $E_B = W_{E_B} \cdot N_R^*$. Due to Lemma 33, we know that vv' is required by B starting from v. Since $vv' \in E_B$, this implies $v \in E_B$. As E_B is N_R^*-closed and as v is a prefix of w, we obtain $w \in E_B$.

Prove $E_B \subseteq E_{B^\downarrow}$. If $w \in E_B$ then there exists a prefix w' of w and a word $w'' \in N_R^*$, so that $w = w'w''$ and w' is an element of $pf(W_{E_B})$. Applying the definition of reduced normal forms, we have $w' \in W_\bot^\downarrow$. This implies $B^\downarrow \sqsubseteq \forall w'.\bot$, which is subsumed by $\forall w'w''.\bot$, according

to the semantics of $\bot$. Due to the definition of E_B, this yields $w'w'' = w \in E_{B^\downarrow}$.

Combining the above two results, we obtain $E_{B^\downarrow} = E_B$, which was to be shown.

Condition 2 and 3: Taking into account the result of (1), it holds that $W_A^\downarrow \cup E_{B^\downarrow}$ is equal to $W_A^\downarrow \cup E_B$ for every $A \in \mathcal{C}$. Applying the definition of $W_A^\downarrow$ yields the expression $(W_A \setminus E_B) \cup E_B$, which is obviously equal to $W_A \cup E_B$. The same argument holds for negated atomic concepts $\neg A$.

Condition 4 and 5: Again, the result of (1) and the definition of $W_{(\geq mR)}^\downarrow$ enable us to expand $\bigcup_{m \geq n} W_{(\geq mR)}^\downarrow \cup E_{B^\downarrow}$ to the expression $\bigcup_{m \geq n} (\bigcup_{p \geq m} W_{(\geq pR)} \setminus E_B) \cup E_B$. By applying distributivity over the union, we obtain $(\bigcup_{m \geq n} \bigcup_{p \geq m} W_{(\geq pR)}) \setminus E_B \cup E_B$, which can be simplified to $(\bigcup_{m \geq n} W_{(\geq mR)}) \setminus E_B \cup E_B$. We can omit subtracting E_B before adding it again, so that we finally have $(\bigcup_{m \geq n} W_{(\geq mR)}) \cup E_B$.

In (1) we have seen that $E_{B^\downarrow} = E_B$. This implies $E_{B^\downarrow} \cdot R^{-1} = E_B \cdot R^{-1}$ for every atomic role R. Consequently, the above argument applies to condition 5 as well.

2. ("$\Leftarrow$") Trivial. ("$\Rightarrow$") If $C \equiv D$, then the characterization of the subsumption allows us to conclude the following conditions again:

   (a) $E_C = E_D$

   (b) $U_A \cup E_C = V_A \cup E_D$

   (c) $U_{\neg A} \cup E_C = V_{\neg A} \cup E_D$

   (d) $\bigcup_{m \geq n} U_{(\geq mR)} \cup E_C = \bigcup_{m \geq n} V_{(\geq mR)} \cup E_D$

   (e) $\bigcup_{m \leq n} U_{(\leq mR)} \cup E_C \cdot R^{-1} = \bigcup_{m \leq n} V_{(\leq mR)} \cup E_D \cdot R^{-1}$

   Taking advantage of Lemma 21, we can infer from condition 1 that $pf(U_{E_C}) = pf(V_{E_D})$, which is equivalent to $U_\bot = V_\bot$, since both concept descriptions are assumed to be reduced. Due to reduction, it also holds that $U_A = U_A \setminus E_C$ and analogously $V_A = V_A \setminus E_D$. Therefore, the unions in condition 2 are disjoint. Because of condition 1 we may replace E_D by E_C in condition 2, which yields $U_A = V_A$. The same argument applies to condition 3. Because C and D are reduced, the role languages $U_{(\leq mR)}$ and $U_{(\geq mR)}$ already contain the union over all

lesser and the union over all greater numbers respectively, as stated in Corollary 30. In condition 4 and 5, we may therefore ommit the unions over m. Moreover, the role languages in condition 4 and 5 are defined as disjoint to E_C and E_D respectively, so that finally the argument for conditions 2 and 3 also applies, yielding $U_{(\leq nR)} = V_{(\leq nR)}$ for every number restriction $(\leq nR) \in \mathcal{N}_{\leq}$ and analogously $U_{(\geq nR)} = V_{(\geq nR)}$ for every $(\geq nR) \in \mathcal{N}_{\geq}$.

3. ("$\Rightarrow$") If $C \sqsubset D$, then from the characterization of subsumption we know that $E_C \supseteq E_D$. We first consider the case that this inclusion is strict, then the case of equality of the languages.

   $E_C \supset E_D$: Then, as stated in [2], there are finite languages U_{E_C} and V_{E_D} such that $pf(U_{E_C})\cdot N_R^* \supset pf(V_{E_D})\cdot N_R^*$. Due to the definition of reduced normal forms, this is equivalent to the inclusion $U_\perp \cdot N_R^* \supset V_\perp \cdot N_R^*$. According to Lemma 21, we can then infer $U_\perp \succ V_\perp$. Since $C \sqsubset D$, we know from the characterization of subsumption that $U_H \cup E_C \supseteq V_H \cup E_D$ for all $H \in \mathcal{C} \cup \{\neg A \mid A \in \mathcal{C}\}$. As mentioned above, this inclusion is equivalent to $U_H \cup U_\perp \cdot N_R^* \supseteq V_H \cup V_\perp \cdot N_R^*$. We may drop the term $V_\perp \cdot N_R^*$ on the right-hand side, obtaining the desired result for all $H \in \mathcal{C} \cup \{\neg A \mid A \in \mathcal{C}\}$.

   For $(\geq nR) \in \mathcal{N}_{\geq}$, we similarly yield

   $$\bigcup_{m \geq n} U_{(\leq mR)} \cup U_\perp \cdot N_R^* = \bigcup_{m \leq n} V_{(\leq mR)} \cup V_\perp \cdot N_R^*.$$

   As mentioned before, the union over all $m \geq n$ can be omitted. Dropping the term $V_\perp \cdot N_R^*$ on the right-hand side of the inclusion afterwards analogously produces $V_H \subseteq U_H \cup U_\perp \cdot N_R^*$, which was to be shown.

   This analogy does not hold for $\leq$-number restrictions, where we need to cope with the right quotient $(\cdot R^{-1})$ in the respective equations: For every $(\leq nR) := H \in \mathcal{N}_{\leq}$, we obtain $U_H \cup U_\perp \cdot N_R^* \cdot R^{-1} \supseteq V_H \cup V_\perp \cdot N_R^* \cdot R^{-1}$. We may drop the expression $V_\perp \cdot N_R^* \cdot R^{-1}$ on the right-hand side of the inclusion. Furthermore, as stated in [2], $U \cdot N_R^* \cdot R^{-1}$ equals $U \cdot N_R^* \cup U \cdot R^{-1}$ for every finite language U over N_R and $R \in N_R$. Consequently, the inclusion can be simplified to $U_H \cup U_\perp \cdot N_R^* \cup U_\perp \cdot R^{-1} \supseteq V_H$, which we wanted to show.

   $E_C = E_D$: As shown in (2), the reduced normal form of C and D then allows us to infer $U_\perp \cdot N_R^* = V_\perp \cdot N_R^*$, which yields $U_\perp = V_\perp$, as both

languages are prefix free. The characterization of the subsumption furthermore allows us to conclude that $U_H \supset V_H$ for every $H \in \mathcal{H}$. Obviously, C and D cannot agree on all role languages, since this would imply $C \equiv D$, in contradiction to the assumption. Consequently, there is one $H \in \mathcal{H}$ such that $U_H \supset V_H$.

("$\Leftarrow$") In case (b), it is not difficult to verify that the conditions for subsumption stated in Lemma 4 are met. Assume Case (a). From $U_\perp \succ V_\perp$ we can infer by Lemma 21 that $U_\perp \cdot N_R^* \supset V_\perp \cdot N_R^*$. Since C and D are reduced, this implies $E_C \supset E_D$, matching the first condition for subsumption. As assumed, for every $H \in \mathcal{H} \setminus \mathcal{N}_\leq$ it holds that $V_H \subseteq U_H \cup U_\perp \cdot N_R^*$. We have already seen in (3) that $U_\perp \cdot N_R^*$ equals E_C. Therefore, after adding the language E_D on both sides of the inclusion we have $V_H \cup E_D \subseteq U_H \cup E_C \cup E_D$. Since E_D is a subset of E_C, we obtain $V_H \cup E_D \subseteq U_H \cup E_C$. For $H \in \mathcal{C} \cup \{\neg A \mid A \in \mathcal{C}\}$, this equals conditions 2 and 3 for the subsumption as stated in Lemma 4.

According to Corollary 30, for all $(\geq nR) \in \mathcal{N}_\geq$ the language $U_{(\geq nR)}$ is equal to the union $\bigcup_{m \geq n} U_{(\geq nR)}$, so that the inclusion $V_H \cup E_D \subseteq U_H \cup E_C$ can be expanded to $\bigcup_{m \geq n} V_{(\geq mR)} \cup E_D \subseteq \bigcup_{m \geq n} U_{(\geq mR)} \cup E_C$, which meets condition 4 for the subsumption.

For $(\leq nR) \in \mathcal{N}_\leq$, we have assumed $V_{(\leq nR)} \subseteq U_{(\leq nR)} \cup U_\perp \cdot N_R^* \cup U_\perp \cdot R^{-1}$. As mentioned above for the reverse direction of (3), we can replace $U_\perp \cdot N_R^* \cup U_\perp \cdot R^{-1}$ by $U_\perp \cdot N_R^* \cdot R^{-1}$, which is equal to $E_C \cdot R^{-1}$. Following a similar line as for the $\geq$-number restrictions, $E_D \cdot R^{-1}$ is added on both sides of the inclusion, yielding $V_{(\leq nR)} \cup E_D \cdot R^{-1} \subseteq U_{(\leq nR)} \cup E_C \cdot R^{-1} \cup E_D \cdot R^{-1}$. As E_C is a superset of E_D and as also both languages are of the form $L \cdot N_R^*$ for some finite language L, it is easy to see that $E_C \cdot R^{-1}$ is a superset of $E_C \cdot R^{-1}$ for every $R \in N_R$. The inclusion therefore simplifies to $V_{(\leq nR)} \cup E_D \cdot R^{-1} \subseteq U_{(\leq nR)} \cup E_C \cdot R^{-1}$. Exploiting Corollary 30, the languages $U_{(\leq nR)}$ and $V_{(\leq nR)}$ can be replaced by the respective unions over all $m \leq n$, thus matching condition 5 of the subsumption conditions of Lemma 4. Consequently, all conditions for subsumption are met. We obtain strict subsumption, because (2) would otherwise imply $U_\perp = V_\perp$, contradicting $U_\perp \succ V_\perp$. ∎

3.4 Termination

Let the substitutions σ_t be defined in Section 3.1. We assume that every $\sigma_t(X)$ is given in $U^{t,X}$-labeled reduced normal form, and that C (as defined in Algorithm 16) is in U-labeled reduced normal form. Then, termination follows from the fact that every solvable matching problem under subsumption conditions has a matcher that only uses concept names already contained in the matching problem M, denoted by the set $\mathcal{C} \subseteq N_C$, and the following three properties of the languages $U_H^{t,X}$ and U_H for $H \in \mathcal{C} \cup \{\bot\}$. In the formulation of these properties we implicitly assume that the substitution σ_t is defined whenever we talk about one of the languages $U_H^{t,X}$.

1. *Suffix property*
   For every variable X and every $H \in \mathcal{C} \cup \{\bot\}$, the set $U_H^{t,X}$ contains only suffices of U_H.

2. *Deletion property*
   For every word w, if $w \in U_H^{t,X} \setminus U_H^{t+1,X}$, then $w \notin U_H^{t',X}$ for any $t' > t$.

3. *Strictness property*
   If σ_t and σ_{t+1} are defined and $\sigma_t \not\equiv \sigma_{t+1}$, then there exists an $H \in \mathcal{C} \cup \{\bot\}$, a variable X, and a word w such that $w \in U_H^{t,X} \setminus U_H^{t+1,X}$.

Note that these properties would not hold if we did not use *reduced* normal forms. In the following three subsections the above termination conditions are shown valid individually for $\mathcal{FL}_\bot$, $\mathcal{FL}_\neg$, and $\mathcal{ALN}$. With these prerequisites we can provide a general proof of termination in Section 3.4.4, yielding a polynomial time upper bound for the three logics under consideration.

3.4.1 Termination properties in $\mathcal{FL}_\bot$

Let us briefly recall our point of departure. We consider the algorithm $\text{MATCH}^{\sqsubseteq}_{\mathcal{FL}_\bot}$, applied to an $\mathcal{FL}_\bot$-matching problem under subsumption conditions M of the form $\langle C \equiv^? D, S\rangle$. M is defined over a finite set $\mathcal{X}$ of variables. We assume C in U-labeled reduced normal form and D in V-labeled normal form. For every subsumption condition $X \sqsubseteq^? E$ in S, we assume E in V^X-labeled normal form. Denote by $T(\text{MATCH}^{\sqsubseteq}_{\mathcal{FL}_\bot}, M)$ the index set of all substitutions computed during the execution of the algorithm $\text{MATCH}^{\sqsubseteq}_{\mathcal{FL}_\bot}$ upon input M. For every $t \in T(\text{MATCH}^{\sqsubseteq}_{\mathcal{FL}_\bot}, M)$ and for every variable X, assume $\sigma_t(X)$ in $U^{t,X}$-labeled reduced normal form.

In order to show the validity of the suffix property, the solution languages introduced in Definition 7 are used to derive a recursive relationship with respect to t between the role languages occurring in consecutive substitutions σ_t. We can then infer the desired properties from σ_0 upward by induction.

Lemma 36 *Suffix property in $\mathcal{FL}_\bot$*
For all $t \in T(\text{MATCH}^{\sqsubseteq}_{\mathcal{FL}_\bot}, M)$ and for all $X \in \mathcal{X}$ it holds that:
1. $U_\bot^{t,X}$ contains only suffixes of $U_\bot$.
2. $U_A^{t,X}$ contains only suffixes of U_A for every $A \in \mathcal{C}$.

PROOF. 1. When performing step t of the algorithm $\text{MATCH}^{\sqsubseteq}_{\mathcal{FL}_\bot}(M)$, the following system of matching problems must be solved.

$$\forall U_\bot.\bot \sqcap \bigsqcap_{A\in\mathcal{C}} \forall U_A.A \equiv^? \forall V_\bot.\bot \sqcap \bigsqcap_{A\in\mathcal{C}} \forall V_A.A \sqcap \bigsqcap_{X\in\mathcal{X}} \forall V_X.X$$
$$\forall U_\bot^{t,X}.\bot \sqcap \bigsqcap_{A\in\mathcal{C}} \forall U_A^{t,X}.A \sqsubseteq^? \forall V_\bot^X.\bot \sqcap \bigsqcap_{A\in\mathcal{C}} \forall V_A^X.A \sqcap \bigsqcap_{X'\in\mathcal{X}} \forall V_{X'}^X.X',$$

where the second line represents one equation for every $X \in \mathcal{X}$. As stated in Section 2.2, this system can be combined into a single matching problem with little difficulty. For the resulting matching problem, setting up the solvability equations proposed in Definition 6 and applying Lemma 7, we yield the following solution language for the bottom-concept.

$$U_\bot^{t+1,X}\cdot N_R^* = \bigcap_{w\in V_X} w^{-1}(U_\bot\cdot N_R^*) \cap \bigcap_{X'\in\mathcal{X}} \bigcap_{w\in V_{X'}^X} w^{-1}(U_\bot^{t,X}\cdot N_R^*) \qquad (*)$$

Due to the notation introduced for the solutions σ_t, here $U_\bot^{t+1,X}\cdot N_R^*$ takes the place of $\widehat{L}_\bot^X$ used in Lemma 7 to denote the solution language for the $\bot$-concept. We have to show that the $U_\bot^{t+1,X}$ contains only suffixes of $U_\bot$.

According to Lemma 22, for every finite language U and for every word w there exists a finite prefix free language L such that firstly, $L\cdot N_R^* = w^{-1}(U\cdot N_R^*)$; and secondly, L contains only suffixes of U. Using this result we now show the proposition for $U_\bot^{t,X}$ by induction over the number of steps t the algorithm $\text{MATCH}^{\sqsubseteq}_{\mathcal{FL}_\bot}(M)$ takes.

$(t = 0)$: According to equation $(*)$, it holds that

$$U_\bot^{0,X}\cdot N_R^* = \bigcap_{w\in V_X} w^{-1}(U_\bot\cdot N_R^*). \qquad (*')$$

At first, we show that the suffix property does not get lost when intersecting languages of the form $L{\cdot}N_R^*$ with that property. It is shown in [2] that for finite languages L and L' the intersection $L{\cdot}N_R^* \cap L'{\cdot}N_R^*$ is equal to $((L \cap L'{\cdot}N_R^*) \cup (L' \cap L{\cdot}N_R^*)){\cdot}N_R^*$.

Obviously, $(L \cap L'{\cdot}N_R^*) \cup (L' \cap L{\cdot}N_R^*)$ is a subset of the union $L \cup L'$. This implies that the intersection $L{\cdot}N_R^* \cap L'{\cdot}N_R^*$ can be represented as $L''{\cdot}N_R^*$ such that every element of L'' comes from L or from L'.

Because of Lemma 22, it holds for every $X \in \mathcal{X}$ and for every $w \in V_X$ that the language $w^{-1}(U_\perp{\cdot}N_R^*)$ can be represented as $L{\cdot}N_R^*$, where L contains only suffixes of $U_\perp$. We have just seen that the suffix property is respected by the intersection. Thus, the entire right-hand side of equation $(*')$ is of the form $L{\cdot}N_R^*$, where L contains only suffixes of $U_\perp$. $pf(L)$ is a subset of L and therefore contains only suffixes as well. $pf(L){\cdot}N_R^*$ also represents the right-hand side of $(*')$, as we know from Lemma 21. From the definition of reduced normal forms in $\mathcal{FL}_\perp$ we also know that $U_\perp^{0,X}$ is prefix free. Lemma 21 now implies that $U_\perp^{0,X}$ is equal to $pf(L)$, completing our argument.

$(t > 0)$: Due to induction, we may assume that all role languages on the right-hand side of equation $(*)$ contain only suffixes of $U_\perp$. Analogous to the argument for the case $t = 0$, the suffix property is valid for $U_\perp^{t+1,X}$ as well.

2. Consider $U_A^{t,X}$ for an arbitrary $A \in \mathcal{C}$. Starting again with the system of matching equations proposed in (1) and taking into account the definition of the solution languages in Lemma 13, we obtain the following

result for $U_A^{t,X}$.

$$\begin{aligned} U_A^{t+1,X} &= \bigcap_{w\in V_X} w^{-1}(U_A \cup U_\perp{\cdot}N_R^*) \cap \bigcap_{X'\in\mathcal{X}} \bigcap_{w\in V_{X'}^X} w^{-1}(U_A^{t,X} \cup U_\perp^{t,X}{\cdot}N_R^*) \\ &\quad \setminus U_\perp^{t+1,X}{\cdot}N_R^* \\ &= \underbrace{\bigcap_{w\in V_X} w^{-1}(U_A \cup U_\perp{\cdot}N_R^*) \cap \bigcap_{X'\in\mathcal{X}} \bigcap_{w\in V_{X'}^X} w^{-1}(U_A^{t,X} \cup U_\perp^{t,X}{\cdot}N_R^*)}_{M_1} \\ &\quad \setminus \underbrace{\bigcap_{w\in V_X} w^{-1}(U_\perp{\cdot}N_R^*) \cap \bigcap_{X'\in\mathcal{X}} \bigcap_{w\in V_{X'}^X} w^{-1}(U_\perp^{t,X}{\cdot}N_R^*)}_{M_2} \\ &\overset{!}{\subseteq} \bigcup_{w\in V_X} w^{-1}(U_A) \cup \bigcup_{X'\in\mathcal{X}} \bigcup_{w\in V_{X'}^X} w^{-1}(U_A^{t,X}) \end{aligned}$$

The equality to $M_1 \setminus M_2$ is obtained by replacing $U_\perp^{t+1,X}{\cdot}N_R^*$ with the right-hand side of equation $(*)$. The last step in the above sequence remains to be shown. Consider an arbitrary word v in $U_A^{t+1,X} = M_1 \setminus M_2$. Since v is not an element of M_2, there exists a word $w \in V_X$ or a word $w' \in V_{X'}^X$ such that v is no element of $w^{-1}(U_\perp{\cdot}N_R^*)$ or no element of $w^{-1}(U_\perp^{t,X}{\cdot}N_R^*)$. Assume the first case, i.e. $v \notin w^{-1}(U_\perp{\cdot}N_R^*)$. As v is an element of M_1, obviously $v \in w^{-1}(U_A \cup U_\perp{\cdot}N_R^*)$, which implies $v \in w^{-1}(U_A)$. Thus, v is a suffix of a word in U_A. The second case is analogous, yielding that v is a prefix of a word in $U_A^{t,X}$. Thus, the inclusion claimed above holds.

Since U_A and all $U_A^{t,X}$ are finite languages, it is not difficult to see that the left quotients $w^{-1}(U_A)$ and $w^{-1}(U_H^{t,X})$ for every word w only contain suffixes of U_A and $U_A^{t,X}$ respectively. We still have to ensure that the suffix property is respected by the union. This can be shown inductively similar to the proof seen in (1) for the intersection. In case of the union, however, the induction argument is by far simpler, since for finite languages L, L' the union $L{\cdot}N_R^* \cup L'{\cdot}N_R^*$ is equal to $(L \cup L'){\cdot}N_R^*$. ∎

For the proof of the deletion property, the characterization of the subsumption for reduced normal forms can be utilized to rule out words reappearing after being deleted. A subsumption argument, of course, can only

be used since we know from the proof of correctess that the solutions σ_t in fact are subsumed by its respective successors σ_{t+1}.

Lemma 37 *Deletion property in $\mathcal{FL}_\bot$*
$\textsc{match}^{\sqsubseteq}_{\mathcal{FL}_\bot}(M)$ *meets the deletion property.*

Proof. We first prove the deletion property for role languages referring to the $\bot$-concept and then consider those referring to atomic concepts $A \in \mathcal{C}$.

$\bot$-concept: Assume that contrary to our claim a word w can reappear for greater values of t after being deleted from a role language at a certain point during the execution of the algorithm. Thus, assume for $w \in N_R^*$ that $w \in U_\bot^{t,X}$ and $w \notin U_\bot^{t',X}$ but finally $w \in U_\bot^{t'+1,X}$ for some $X \in \mathcal{X}$ and for nonnegative integers $t < t' \in T(\textsc{match}^{\sqsubseteq}_{\mathcal{FL}_\bot}, M)$.

We know from Lemma 17 that $\sigma_t \sqsubseteq \sigma_{t'} \sqsubseteq \sigma_{t'+1}$. As all substitutions are reduced we further know due to our assumption, that $\sigma_t(X_j) \not\equiv \sigma_{t'}(X_j) \not\equiv \sigma_{t'+1}(X_j)$. From this we can infer by means of Lemma 25 that $U_\bot^{t,X} \succ U_\bot^{t',X} \succ U_\bot^{t'+1,X}$.

We have assumed that $w \in U_\bot^{t'+1,X}$. The above relation then for $U_\bot^{t',X}$ demands that $U_\bot^{t',X}$ contains a prefix w' of w. As w is no element of $U_\bot^{t',X}$, this is a nontrivial prefix. Similarly we find that $U_\bot^{t,X}$ contains a prefix of w' or w' itself. The language $U_\bot^{t,X}$, however, initially was assumed to contain w as well, yielding a contradiction to $U_\bot^{t,X}$ being prefix free.

A-concept: Assume similarly for a word $w \in N_R^*$ that $w \in U_A^{t,X}$ and $w \notin U_A^{t',X}$ but finally $w \in U_A^{t'+1,X}$ for some $X \in \mathcal{X}$, for $A \in \mathcal{C}$, and for nonnegative integers $t < t' \in (\textsc{match}^{\sqsubseteq}_{\mathcal{FL}_\bot}, M)$. Since $\sigma_t \sqsubseteq \sigma_{t'} \sqsubseteq \sigma_{t'+1}$ and as also all substitutions are reduced we obtain as a consequence of Lemma 25:

$$U_A^{t,X} \dot\cup U_\bot^{t,X} \cdot N_R^* \supseteq U_A^{t',X} \dot\cup U_\bot^{t',X} \cdot N_R^* \supseteq U_A^{t'+1,X} \dot\cup U_\bot^{t'+1,X} \cdot N_R^*.$$

We have assumed that $w \in U_A^{t'+1,X}$. Since w is no element of $U_A^{t',X}$, the subset relation implies that $w \in U_\bot^{t',X} \cdot N_R^*$. From the characterization of the subsumption we know that $U_\bot^{t,X} \cdot N_R^* \supseteq U_\bot^{t',X} \cdot N_R^*$, which in our case implies $w \in U_\bot^{t,X} \cdot N_R^*$. This contradicts the disjointedness of the union with $U_A^{t,X}$, which was shown in Lemma 25. ∎

As the next lemma will show, the strictness property is obtained as an immediate consequence of Lemma 17 (soundness and completeness) and the caracterization of strict subsumption for reduced normal forms.

Lemma 38 *Strictness property*
$\text{MATCH}^{\sqsubseteq}_{\mathcal{FL}_\bot}(M)$ *meets the strictness property.*

PROOF. It is shown in Lemma 17 that $\sigma_t \sqsubseteq \sigma_{t+1}$ for every $t \in T(\text{MATCH}^{\sqsubseteq}_{\mathcal{FL}_\bot}, M)$. Since the fixed point iteration in $\text{MATCH}^{\sqsubseteq}_{\mathcal{FL}_\bot}$ terminates in case $\sigma_t \equiv \sigma_{t+1}$, we have $\sigma_t \sqsubset \sigma_{t+1}$ for every t as long as the iteration does not terminate. The strict subsumption of the substitutions implies that for every t there is a variable $X \in \mathcal{X}$ such that $\sigma_t(X) \sqsubset \sigma_{t+1}(X)$.

Due to the characterization of strict subsumption for reduced normal forms (Lemma 25), this implies that either $U_\bot^{t,X} \succ U_\bot^{t+1,X}$ or $U_A^{t,X} \supset U_A^{t+1,X}$ for some $A \in \mathcal{C}$. In both cases at least one word in one role language is removed at the transition from σ_t to σ_{t+1}. ∎

3.4.2 Termination properties in $\mathcal{FL}_\neg$

For $\mathcal{FL}_\neg$, a separate proof of termination is omitted, because we can exploit the analogy to $\mathcal{FL}_\bot$. Verifying the termination properties again yields a positive result, which is stated below without proof. Let M be an $\mathcal{FL}_\neg$-matching problem under subsumption conditions.

Lemma 39 *Termination properties in* $\mathcal{FL}_\neg$
$\text{MATCH}^{\sqsubseteq}_{\mathcal{FL}_\neg}(M)$ *meets the suffix, deletion and strictness property.*

Let us discuss briefly why we can expect to gain the same result for $\mathcal{FL}_\neg$ in exactly the same way as seen for $\mathcal{FL}_\bot$. The idea is to show that due to the reduced normal form of all substitutions σ_t occurring during the execution of $\text{MATCH}^{\sqsubseteq}_{\mathcal{FL}_\neg}(M)$, the validity of the three termination properties can be shown analogous to the proof for $\mathcal{FL}_\bot$. Recall that the prerequisites for the existence of a solution in $\mathcal{FL}_\neg$ are stronger than in $\mathcal{FL}_\bot$. Nevertheless, once the matching problem is solvable, the solution assigned by σ_t is syntactically similar to that in $\mathcal{FL}_\bot$—the only difference being the construct $\widehat{U}$ instead of U. This can be found when comparing Lemma 7 and Lemma 9, where the solution languages are introduced. In the presence of reduced normal forms the difference between languages of the form $\widehat{U}$ and U disappears, as stated in Corollary 27. Furthermore, a comparison of Lemma 25 and Lemma 28 yields the same characerization of equivalence and subsumption for reduced normal forms in $\mathcal{FL}_\bot$ and $\mathcal{FL}_\neg$. Hence, the results obtained for $\mathcal{FL}_\neg$ are analogous to those for $\mathcal{FL}_\bot$.

3.4.3 Termination properties in $\mathcal{ALN}$

The overall task of solving matching problems in $\mathcal{ALN}$ is significantly more involved than in its sublanguages. However, most of the additional complexity is hidden in the notion of excluding words, which has been studied in depth in [12]. Once we know that sets of excluding words are of the form $L \cdot N_R^*$ for some finite language L, we do not need to introduce new ideas to prove the termination properties. By virtue of the reduced normal forms we again find a situation analogous to $\mathcal{FL}_\bot$, though consisting of considerably larger equations. Let M denote an $\mathcal{ALN}$-matching problem under subsumption conditions analogous to that defined in Section 3.4.1.

Lemma 40 *Suffix property in $\mathcal{ALN}$*
For all $t \in T(\text{MATCH}^{\sqsubseteq}_{\mathcal{ALN}}, M)$ and for all $X \in \mathcal{X}$ it holds that:
1. $U_\bot^{t,X}$ contains only suffixes of $U_\bot$.
2. $U_A^{t,X}$ contains only suffixes of U_A for every $A \in \mathcal{C}$ and $U_{\neg A}^{t,X}$ contains only suffixes of $U_{\neg A}$ for every $A \in \mathcal{C}$.
3. $U_{(\geq nR)}^{t,X}$ contains only suffixes of $U_{(\geq nR)}$ for every $(\geq nR) \in \mathcal{N}_\geq$.
4. $U_{(\leq nR)}^{t,X}$ contains only suffixes of $U_{(\leq nR)} \cup U_\bot \cdot R^{-1}$ for every $(\leq nR) \in \mathcal{N}_\leq$.

PROOF. • At step t of the algorithm $\text{MATCH}^{\sqsubseteq}_{\mathcal{ALN}}(M)$, the following system of matching problems has to be solved:

$$\forall U_\bot.\bot \sqcap \bigsqcap_{A\in\mathcal{C}} \forall U_A.A \sqcap \bigsqcap_{A\in\mathcal{C}} \forall U_{\neg A}.\neg A$$
$$\sqcap \bigsqcap_{(\geq nR)\in\mathcal{N}_\geq} \forall U_{(\geq nR)}.(\geq nR) \sqcap \bigsqcap_{(\leq nR)\in\mathcal{N}_\leq} \forall U_{(\leq nR)}.(\leq nR)$$
$$\equiv^?$$
$$\forall V_\bot.\bot \sqcap \bigsqcap_{A\in\mathcal{C}} \forall V_A.A \sqcap \bigsqcap_{A\in\mathcal{C}} \forall V_{\neg A}.\neg A$$
$$\sqcap \bigsqcap_{(\geq nR)\in\mathcal{N}_\geq} \forall V_{(\geq nR)}.(\geq nR) \sqcap \bigsqcap_{(\leq nR)\in\mathcal{N}_\leq} \forall V_{(\leq nR)}.(\leq nR)$$
$$\sqcap \bigsqcap_{X\in\mathcal{X}} \forall V_X.X$$

and for every $X \in \mathcal{X}$:

$$\forall U_\bot^{t,X}.\bot \sqcap \bigsqcap_{A\in\mathcal{C}} \forall U_A^{t,X}.A \sqcap \bigsqcap_{A\in\mathcal{C}} \forall U_{\neg A}^{t,X}.\neg A$$
$$\sqcap \bigsqcap_{(\geq nR)\in\mathcal{N}_\geq} \forall U_{(\geq nR)}^{t,X}.(\geq nR) \sqcap \bigsqcap_{(\leq nR)\in\mathcal{N}_\leq} \forall U_{(\leq nR)}^{t,X}.(\leq nR)$$
$$\sqsubseteq^?$$
$$\forall V_\bot^{X}.\bot \sqcap \bigsqcap_{A\in\mathcal{C}} \forall V_A^{X}.A \sqcap \bigsqcap_{A\in\mathcal{C}} \forall V_{\neg A}^{X}.\neg A$$
$$\sqcap \bigsqcap_{(\geq nR)\in\mathcal{N}_\geq} \forall V_{(\geq nR)}^{X}.(\geq nR) \sqcap \bigsqcap_{(\leq nR)\in\mathcal{N}_\leq} \forall V_{(\leq nR)}^{X}.(\leq nR)$$
$$\sqcap \bigsqcap_{X'\in\mathcal{X}} \forall V_{X'}^{X}.X'$$

This system can be combined into a single matching problem. For the solution to this problem, Lemma 12 provides us with appropriate solution languages. Regarding the $\bot$-concept, we obtain the following result for the solution language $U_\bot^{t+1,X}$ assigned by $\sigma_{t+1}(X_j)$:

$$U_\bot^{t+1,X}\cdot N_R^* = \bigcap_{w\in V_X} w^{-1}(E_C) \cap \bigcap_{X'\in\mathcal{X}} \bigcap_{w\in V_{X'}^X} w^{-1}(E_C^{t,X}) \qquad (*)$$

Again, due to our notation $U_\bot^{t+1,X}\cdot N_R^*$ takes the place of $\widehat{L}_\bot^X$ as used in Lemma 12. Furthermore, E_C denotes the set of C-excluding words and analogously $E_C^{t,X}$ the set of excluding words for the matching problem corresponding to the variable X the above system of matching problems.

We may assume C to be in reduced normal form. Consequently, it holds that $U_\bot\cdot N_R^* = E_C$, as seen in Definition 29. As σ_t is also in reduced normal form, we furthermore obtain that $U_\bot^{t,X}\cdot N_R^* = E_C^{t,X}$ for every $t \in T(\textsc{match}_{\mathcal{ALN}}^{\sqsubseteq}, M)$. In Equation $(*)$, we may therefore replace E_C by $U_\bot\cdot N_R^*$ and $E_C^{t,X}$ by $U_\bot^{t,X}\cdot N_R^*$. This reveals the inductive relation of the role languages:

$$U_\bot^{t+1,X}\cdot N_R^* = \bigcap_{w\in V_X} w^{-1}(U_\bot\cdot N_R^*) \cap \bigcap_{X'\in\mathcal{X}} \bigcap_{w\in V_{X'}^X} w^{-1}(U_\bot^{t,X}\cdot N_R^*) \qquad (*')$$

It is to prove that $U_\bot^{t+1,X}$ contains only suffixes of $U_\bot$. Equation $(*')$ is only a syntactic variant of Equation $(*)$ established in Lemma 36. As $U_\bot^{t+1,X}$ is prefix free, we can prove the claim following exactly the same pattern as seen for $\mathcal{FL}_\bot$ in Lemma 36.

- From the system of matching problems introduced in (1), we now derive solutions for role languages of the form $U_A^{t+1,X}$ referring to the atomic concept A in $\sigma_{t+1}(X_j)$. By virtue of Lemma 12 we obtain:

$$U_A^{t+1,X} = \bigcap_{w \in V_X} w^{-1}(U_A \cup E_C) \cap \bigcap_{X' \in \mathcal{X}} \bigcap_{w \in V_{X'}^X} w^{-1}(U_A^{t,X} \cup E_C^{t,X}) \setminus U_\perp^{t+1,X} \cdot N_R^*$$

  Taking into account that $U_\perp \cdot N_R^* = E_C$ and that $U_\perp^{t,X} \cdot N_R^* = E_{t,j,C}$, we can apply the argument of Lemma 36 and replace the expression $U_\perp^{t+1,X} \cdot N_R^*$ with the right-hand side of Equation $(*')$. Again, we can obtain an upper bound for the resulting expression, yielding that

$$U_A^{t+1,X} \subseteq \bigcup_{w \in V_X} w^{-1}(U_A) \cup \bigcup_{X' \in \mathcal{X}} \bigcup_{w \in V_{X'}^X} w^{-1}(U_A^{t,X}).$$

  Because U_A and every $U_A^{t+1,X}$ is finite, it is not difficult to prove that $w^{-1}(U_A)$ and every $w^{-1}(U_A^{t+1,X})$ contain only suffixes of U_A. We know from Lemma 36, that this property is respected by the union, thus completing the proof. For role languages $U_{\neg A}^{t,X}$ referring to negated atomic concepts $\neg A$, exactly the same argument holds.

- We already know that σ_t is in reduced normal form for every $t \in T(\text{MATCH}_{\mathcal{ALN}}^{\sqsubseteq}, M)$. Thus, we have for every number restriction $(\geq nR) \in \mathcal{N}_\geq$ that $\bigcup_{m \geq n} U_{(\geq mR)}^{t,X}$ is equal to $U_{(\geq nR)}^{t,X}$, i.e. the union can be omitted. The same holds for C, which is in reduced normal form as well. Therefore, the expression $\bigcup_{m \geq n} U_{(\geq mR)}$ similarly can be replaced by $U_{(\geq nR)}$. This observation enables us to simplify the solution language derived from the system of matching problems proposed in (1). By means of Lemma 12, we can infer for $U_{(\geq nR)}^{t+1,X}$ that:

$$\begin{aligned} U_{(\geq nR)}^{t+1,X} &= \bigcap_{w \in V_X} w^{-1}(\bigcup_{m \geq n} U_{(\geq mR)} \cup E_C) \cap \bigcap_{X' \in \mathcal{X}} \bigcap_{w \in V_{X'}^X} w^{-1}(\bigcup_{m \geq n} U_{(\geq mR)}^{t,X} \cup E_C^{t,X}) \\ &\quad \setminus U_\perp^{t+1,X} \cdot N_R^* \\ &= \bigcap_{w \in W_j} w^{-1}(U_{(\geq nR)} \cup E_C) \cap \bigcap_{X' \in \mathcal{X}} \bigcap_{w \in V_{X'}^X} w^{-1}(U_{(\geq nR)}^{t,X} \cup E_C^{t,X}) \\ &\quad \setminus U_\perp^{t+1,X} \cdot N_R^* \end{aligned}$$

We can see that after removing the unions for the number restrictions, the above equation is syntactically identical to the one derived for $A \in \mathcal{C}$ in (2). The rest of the argument therefore is identical to what has been proposed there.

- For $(\leq)$-number restrictions, we can again remove the union-operator in the same fashion as done in (3). However, we obtain slightly different results for the solution languages derived from the system of matching problems introduced in (1). For $U^{t+1,X}_{(\leq nR)}$ we can infer that:

$$\begin{aligned} U^{t+1,X}_{(\leq nR)} = & \bigcap_{w \in V_X} w^{-1}(\bigcup_{m \geq n} U_{(\leq mR)} \cup E_C \cdot R^{-1}) \\ & \cap \bigcap_{X' \in \mathcal{X}} \bigcap_{w \in V^X_{X'}} w^{-1}(\bigcup_{m \geq n} U^{t,X}_{(\leq mR)} \cup E^{t,X}_C \cdot R^{-1}) \\ & \setminus U^{t+1,X}_{\perp} \cdot N^*_R \\ = & \bigcap_{w \in V_X} w^{-1}(U_{(\leq nR)} \cup (U_\perp \cdot N^*_R) \cdot R^{-1}) \\ & \cap \bigcap_{X' \in \mathcal{X}} \bigcap_{w \in V^X_{X'}} w^{-1}(U^{t,X}_{(\leq nR)} \cup (U^{t,X}_\perp \cdot N^*_R) \cdot R^{-1}) \\ & \setminus \underbrace{\left(\bigcap_{w \in V_X} w^{-1}(U_\perp \cdot N^*_R) \cap \bigcap_{X' \in \mathcal{X}} \bigcap_{w \in V^X_{X'}} w^{-1}(U^{t,X}_\perp \cdot N^*_R) \right)}_{=:M_2} \end{aligned}$$

Observe, that in the second step we could replace E_C by $U_\perp \cdot N^*_R$ and $E^{t,X}_C$ by $U^{t,X}_\perp \cdot N^*_R$. This replacement is valid because C and σ_t are in reduced normal form. However, the result deviates from the pattern seen in the previous cases of this proof—the right-quotients of $U_\perp \cdot N^*_R$ and $U^{t,X}_\perp \cdot N^*_R$ occur instead of the original languages. Nevertheless, we can simplify the right quotient thanks to the finiteness of $U_\perp$ and $U^{t,X}_\perp$: $(U_\perp \cdot N^*_R) \cdot R^{-1}$ equals $U_\perp \cdot R^{-1} \cup U_\perp \cdot N^*_R$ and similarly $(U^{t,X}_\perp \cdot N^*_R) \cdot R^{-1}$ can be simplified to $U^{t,X}_\perp \cdot R^{-1} \cup U^{t,X}_\perp \cdot N^*_R$ for all t and X. Since after this transformation all right quotients refer to finite languages, we can subtract M_2 and follow the argument familiar from Lemma 36.

Consequently, we obtain:

$$U^{t+1,X}_{(\leq nR)} \subseteq \bigcup_{w \in V_X} w^{-1}(U_{(\leq nR)} \cup U_\bot \cdot R^{-1}) \cup \bigcup_{X' \in \mathcal{X}} \bigcup_{w \in V^X_{X'}} w^{-1}(U^{t,X}_{(\leq nR)} \cup U^{t,X}_\bot \cdot R^{-1})$$

Finally, we can again employ an induction argument to prove that every $U^{t+1,X}_{(\leq nR)}$ contains only suffixes of $U_{(\leq nR)} \cup U_\bot \cdot R^{-1}$. ∎

After eliminating the union over number restrictions and the right-quotient for $(\leq)$-number restrictions in the above equations, the resulting situation appeared very similar to the analogous problems for $\mathcal{FL}_\bot$. Recalling the characterizations of equivalence and subsumption for reduced normal forms in $\mathcal{FL}_\bot$ and $\mathcal{ALN}$, this is not surprising. By comparing Lemma 25 and Lemma 35, we find almost the same conditions for subsumption. Note that we again assumed C to be in reduced normal form.

Lemma 41 *Deletion property in $\mathcal{ALN}$*
$\textsc{match}^{\sqsubseteq}_{\mathcal{ALN}}(M)$ *meets the deletion property.*

Proof. At first, the assertion is proved for role languages referring to the $\bot$-concept and then for the remaining cases.

$\bot$-concept: Assume that a word w can appear in a role language for greater t after having been deleted, i.e. there exists a word $w \in N^*_R$, and indices $t < t' \in T(\textsc{match}^{\sqsubseteq}_{\mathcal{ALN}}, M)$ and an $X \in \mathcal{X}$ such that $w \in U^{t,X}_\bot$ and $w \notin U^{t',X}_\bot$ but $w \in U^{t'+1,X}_\bot$. We can now infer a contradiction to $U^{t,X}_\bot$ being prefix free, as already done for $\mathcal{FL}_\bot$ in Lemma 37.

As the substitutions σ_t, $\sigma_{t'}$ and $\sigma_{t'+1}$ are reduced, we can infer from the assumptions by virtue of the properties of reduced normal forms in $\mathcal{ALN}$ that $U^{t,X}_\bot \succ U^{t',X}_\bot \succ U^{t'+1,X}_\bot$. The rest of the argument is analogous to Lemma 37. We apply the definition of the multiset order $(\succ)$ and infer that $U^{t,X}_\bot$ must contain a nontrivial prefix of w as well as w itself.

Other cases: Assume similarly for a word $w \in N^*_R$ that $w \in U^{t,X}_A$ and $w \notin U^{t',X}_A$, but $w \in U^{t'+1,X}_A$ for an atomic concept $A \in \mathcal{C}$, for some $X \in \mathcal{X}$, and for nonnegative integers $t < t' \in T$. Since again $\sigma_t \sqsubseteq \sigma_{t'} \sqsubseteq \sigma_{t'+1}$ and since all substitutions are reduced, we yield by Lemma 35:

$$U^{t,X}_A \dot\cup U^{t,X}_\bot \cdot N^*_R \supseteq U^{t',X}_A \dot\cup U^{t',X}_\bot \cdot N^*_R \supseteq U^{t'+1,X}_A \dot\cup U^{t'+1,X}_\bot \cdot N^*_R$$

Now we can follow the argument employed in Lemma 37 to infer a contradiction to the disjointness of the unions. It is shown in Lemma 35 that the argument of disjoint unions also applies for negated atomic concept and number restrictions. ∎

The proof of the strictness property for $\mathcal{ALN}$ is identical to the previous case for $\mathcal{FL}_\bot$. This can be readily seen—firstly, Lemma 17 (soundness and completeness) is valid for $\mathcal{ALN}$ as well; and secondly, the caracterization of strict subsumption for $\mathcal{ALN}$-concept descriptions in Lemma 35 yields the same superset relation for the role languages as used in Lemma 38. Since no other argument was necessary there, the same strategy works for $\mathcal{ALN}$ as well. We may therefore state the result without proof, concluding the proofs of the termination properties:

Lemma 42 *Strictness property*
$\text{MATCH}^{\sqsubseteq}_{\mathcal{ALN}}(M)$ *meets the strictness property.*

3.4.4 General result

Given the three termination properties, it is now easy to show that the algorithm halts after a polynomial number of steps. In fact, Property 1 (suffix property) yields a polynomial upper bound on the size of the role languages $U_H^{t,X}$. Property 3 (strictness property) shows that in every step of the iteration at least one word is removed from one of these languages, and Property 2 (deletion property) ensures that words that have been removed cannot reappear. To sum up, we have shown the following theorem.

Theorem 43 *Let* $\mathcal{L} \in \{\mathcal{FL}_\bot, \mathcal{FL}_\neg, \mathcal{ALN}\}$. *The algorithm* $\text{MATCH}^{\sqsubseteq}_{\mathcal{L}}$ *is a polynomial time algorithm that, given an* $\mathcal{L}$*-matching problem with subsumption conditions, returns a least matcher of this problem if it is solvable, and "fail" otherwise.*

It should be noted that the algorithm $\text{MATCH}^{\sqsubseteq}_{\mathcal{L}}$ does not work for $\mathcal{L} = \mathcal{FL}_0$. In the following section we will therefore briefly discuss the additional conditions necessary to extend Theorem 43 to $\mathcal{FL}_0$.

3.5 Matching under subsumption conditions in $\mathcal{FL}_0$

The language $\mathcal{FL}_0$ does not allow for the bottom concept, and thus the initialization step (Step 1) of Algorithm 16 is not possible. Instead of starting with $\sigma(X) := \bot$, the algorithm can also start from the least matcher of $C \equiv^? D$. In case the side conditions do not introduce new variables (i.e., variables not contained in D), this modification works and yields a polynomial time matching algorithm. In contrast, if new variables are introduced, then we can show that the size of the least matcher may grow exponentially in the size of the matching problem. The following example, which has also been dicussed in [2], illustrates this.

Example 44 Let $N_R = \{R, S\}$. For some $n \in \mathbb{N}$, assume $\mathcal{X} = \{X_1, \ldots, X_n\}$. Consider the (trivial) $\mathcal{FL}_0$-matching problem $\top \equiv^? \top$ under the subsumption conditions $\{X_0 \sqsubseteq^? A\} \cup \{X_{i+1} \sqsubseteq^? \forall\{R, S\}.X_i \mid 0 \leq i \leq n-1\}$.

Combining the first subsumption condition with the second one yields that every solution to the matching problem has to respect the subsumption condition $X_1 \sqsubseteq \forall\{R, S\}.A$. It is easy to see by induction that for every $i \in \{1, \ldots, n\}$ we have

$$X_i \sqsubseteq \forall\{R, S\}^{2^i}.A,$$

denoting by $\{R, S\}^{2^i}$ the the set of all words of length 2^i over the alphabet $\{R, S\}$. Hence, for every solution σ to the matching problem it holds that $\sigma(X_n)$ must assign a role language of exponential size in n corresponding to the atomic concept A.

The above example suggests a solution strategy for $\mathcal{FL}_0$-matching problems $M =: \langle C \equiv^? D, S_0 \rangle$ with new variables occurring in subsumption conditions. The strategy comprises six steps which are explained below.

1. It is shown in [2] that we can transform S_0 into an equivalent set S_1 of acyclic subsumption conditions whose size is polynomial in the size of S_0.

2. Analogous to the above example, S_1 is then transformed into an equivalent set S_2 such that every variable occurring in S_2 either occurs only on left-hand sides of subsumption conditions or only on right-hand sides. To this end the substitution $\{X \mapsto E \mid X \sqsubseteq^? E \in S_1\}$ is applied to the right-hand side E' of every subsumption condition $X' \sqsubseteq^? E' \in S_1$. After at most $|S_1|$ iterations the set of subsumption conditions has the

required form. As shown by the example, the size of S_2 may be exponential in that of S_1. Note that this modification would not preserve equivalence in case of strict subsumption conditions.

3. Every variable neither occurring in $C \equiv^? D$ nor on left-hand sides of subsumption conditions in S_2 is now substituted by $\top$, yielding S_3.

4. Finally, S_4 is obtained from S_3 by removing every subsumption condition $X \sqsubseteq^? E$ where X occurs neither in $C \equiv^? D$ nor on any right-hand side of any subsumption condition in S_3. Obviously, every variable occurring in S_4 also occurs in the original matching problem $C \equiv^? D$.

5. The problem $\langle C \equiv^? D, S_4 \rangle$ is then solved with the modified algorithm $\text{MATCH}^{\sqsubseteq}_{\mathcal{FL}_0}$ starting by solving $C \equiv^? D$ instead of assigning $\bot$ to every variable in D. Denote by σ the solution returned in case of a successful computation.

6. σ assigns values only to variables occurring in D. For a solution θ including all variables in M we proceed as follows. For every variable X occurring only on right-hand sides of subsumption conditions in S_3 (and not in $C \equiv^? D$), define $\theta(X) := \top$. For those variables X occurring only on left-hand sides, let $\theta(X) := \prod_{X \sqsubseteq^? E \in S_3} \sigma(E)$.

One can see that the possible exponential blow-up in Step 2 makes the above strategy an exponential time algorithm.

Nevertheless, the size of the substitutions for variables in D can still be bounded polynomially, and if one is only interested in substitutions for these variables, then these can still be computed in polynomial time.

4 Matching under general side conditions

Matching under general side conditions (i.e., strict and non-strict subsumption conditions) is more complex than matching under subsumption conditions for two reasons.

First, as already shown in [2], deciding the solvability of an $\mathcal{FL}_0$-matching problem under strict (and acyclic) subsumption conditions is NP-hard. It is easy to see that the same reduction works for the DLs $\mathcal{FL}_\bot$, $\mathcal{FL}_\neg$, and $\mathcal{ALN}$. Thus, assuming that P $\neq$ NP, there cannot exist a polynomial time algorithm computing matchers of matching problems under general side conditions.

Second, as shown by the following example, solvable matching problems under strict subsumption conditions no longer need to have a *least* matcher (but rather finitely many *minimal* matchers).

Example 45 Consider the $\mathcal{FL}_\bot$-matching problem

$$A_1 \sqcap \ldots \sqcap A_n \equiv^? X_1 \sqcap \ldots \sqcap X_n$$

under the strict subsumption conditions

$$\{X_{i+1} \sqsubset^? X_i \mid 1 \leq i \leq n-1\} \cup \{X_1 \sqsubset^? \top\}.$$

The pure matching problem enforces that each X_i must be replaced by a (possibly empty) conjunction of concept names from $\{A_1, \ldots, A_n\}$. Thus, the strict subsumption conditions can only be satisfied if X_1 is replaced by one of these names, X_2 by a conjunction of this name with an additional one, etc. From this it is easy to derive that the matchers of the problem are of the following form: given a permutation $P := (p_1, \ldots, p_n)$ of $(1, \ldots, n)$, the substitution σ^P is defined by $\sigma^P(X_i) := A_{p_1} \sqcap \ldots \sqcap A_{p_i}$ $(1 \leq i \leq n)$. Thus, there are $n!$ non-equivalent matchers, and it is easy to see that each of them is minimal.

The new contribution of this section is a (non-deterministic) algorithm, $\text{MATCH}^{\sqsubset}_{\mathcal{L}}$, that computes matchers of $\mathcal{L}$-matching problems under general side conditions for $\mathcal{L} \in \{\mathcal{FL}_\bot, \mathcal{FL}_\neg\}$. (We strongly conjecture that a similar algorithm can also be used for $\mathcal{ALN}$.) This non-deterministic algorithm matches the lower complexity bound (NP hard) for the decision problem in the following sense. The length of every computation path of this algorithm is polynomially bounded in the size of the given matching problem. In case the problem is not solvable, every computation returns "fail". Otherwise, the successful computation paths yield all minimal matchers. The algorithm proceeds in two steps: first it eliminates cycles and then solves the resulting matching problem with acyclic side conditions.

4.1 Eliminating cycles

In [2], $\mathcal{FL}_0$-matching problems with cyclic subsumption conditions are transformed into equivalent ones with acyclic subsumption conditions.

In this context, ε-cycles and role cycles must be distinguished. We say that X directly ε-depends on Y iff there is a side condition $X \; \rho \; E$ such that

Y occurs in the top-level conjunction of E. Now, the notion "ε-dependence" is defined in the obvious way, and X lies on an ε-cycle iff it ε-depends on itself. For example, w.r.t. $S := \{X \sqsubseteq^? X \sqcap \forall r.Y\}$, the variable X ε-depends on itself, and it depends on Y (but does not ε-depend on Y).

If an ε-cycle involves a strict subsumption condition, then the problem is unsolvable. Otherwise, ε-cycles can be removed by first replacing all variables occurring on such a cycle by the same variable. The remaining ε-cycles are due to subsumption conditions of the form $X \sqsubseteq^? X \sqcap E$. But such a condition is equivalent to $X \sqsubseteq^? E$.

If X is a variable on a role cycle (i.e., a cycle that is not an ε-cycle), the we can show that solutions (in $\mathcal{FL}_\bot, \mathcal{FL}_\neg$) must replace X by either $\top$ or $\bot$. The next lemma provides the relevant result.

Lemma 46 *Solutions to role cycles*
Let $X\ \rho^?\ \forall\{v\}.X$ be a subsumption condition in an $\mathcal{FL}_\bot$-matching problem M, where $v \neq \varepsilon$. Let σ be a solution to M respecting the side condition. Then,

1. *If $\rho = \sqsubseteq$, then $\sigma(X) \equiv \bot$ or $\sigma(X) \equiv \top$.*
2. *If $\rho = \sqsubset$, then $\sigma(X) \equiv \bot$.*

Proof. • Without loss of generality we may assume that σ is reduced. Denote $\sigma(X)$ in U-labeled reduced normal form. If σ respects the side condition, then we have $\sigma(X) \sqsubseteq \forall\{v\}.\sigma(X)$. The characterization of subsumption (Lemma 1) implies that the conditions

$$U_\bot \cdot N_R^* \supseteq \{v\} \cdot U_\bot \cdot N_R^*$$
$$U_A \cup U_\bot \cdot N_R^* \supseteq \{v\} \cdot U_A \cup \{v\} \cdot U_\bot \cdot N_R^*$$

hold for all $A \in \mathcal{C}$. We have to show that i) $\sigma(X) = \bot$ and $\sigma(X) = \top$ solves $X\ \rho^?\ \forall W.X$ and that ii) these are the only valid solutions.

i) If $\sigma(X) = \bot$, then the reduced normal form implies that $U_\bot = \{\varepsilon\}$. This yields the strict inclusion $U_\bot \cdot N_R^* \supset \{v\} \cdot U_\bot \cdot N_R^*$, since $v \neq \varepsilon$, and also respects the second condition, since $U_A \supseteq U_A$ holds for any choice of U_A. Consequently, we find that $\sigma(X) \equiv \bot$ solves the side condition for $\rho = \sqsubseteq$.

If $\sigma(X) = \top$, then we have $U_\bot = U_A = \emptyset$ which for the first condition yields $\emptyset \supseteq \emptyset$. Hence, the first condition for subsumption obviously

holds, while for the second one we get $U_A \supseteq W \cdot U_A$ for all $A \in \mathcal{C}$. This holds, since $U_A = \emptyset$. Hence, $\sigma(X) \equiv \bot$ solves the side condition for $\rho = \sqsubseteq$.

Note that in case $U_\bot = \emptyset$ the assignment $U_A = \emptyset$ is the only valid solution for $U_A \supseteq W \cdot U_A$ and that no possible value for some U_A yields a strict inclusion $U_A \supset W \cdot U_A$.

ii) The previous remark implies that for any reduced solution $\sigma(X) \notin \{\bot, \top\}$ solving the side condition, the role language $U_\bot$ cannot be empty. Thus, assume $u \in U_\bot \setminus \{\varepsilon\}$. it can be shown that this implies a contradiction to $U_\bot$ being finite and prefix free.

• ∎

Consequently, we can (non-deterministically) guess such a substitution for variables on role cycles. Note that the side conditions with such a variable as left-hand side are either obviously unsolvable or give rise to additional matching problems. For example, if we replace X in $X \sqsubset^? Y \sqcap \forall r.X$ by $\bot$ then the condition $\bot \sqsubset^? Y \sqcap \forall r.\bot$ can be expressed by the matching problem $\bot \equiv^? Z$ under the side condition $Z \sqsubset^? Y \sqcap \forall r.\bot$.

4.2 The algorithm handling acyclic side conditions

In the following, let $M = \langle C \equiv^? D, S\rangle$ be an $\mathcal{L}$-matching problem ($\mathcal{L} \in \{\mathcal{FL}_\bot, \mathcal{FL}_\neg\}$) under acyclic side conditions. Let $S = \{X_1 \ \rho_1^? \ E_1, \ldots, X_\ell \ \rho_\ell^? \ E_\ell\}$ for distinct variables $X_1, \ldots, X_\ell$ and patterns $E_1, \ldots, E_\ell$ such that E_i does not contain the variables $X_i, \ldots, X_\ell$. (The case where not all the left-hand side variables are distinct can be treated similarly.) We denote by $S_\sqsubseteq$ the set of side conditions obtained from S by replacing every ρ_i by $\sqsubseteq$.

Applied to input M, the algorithm $\text{MATCH}_\mathcal{L}^\sqsubseteq$ first calls $\text{MATCH}_\mathcal{L}^\sqsubseteq(\langle C \equiv^? D, S_\sqsubseteq\rangle)$. If this yields "fail", then M is also unsolvable. Otherwise, the computed substitution σ solves $C \equiv^? D$, but may still violate some of the strict subsumption conditions. Starting with the violated side condition with the largest index, the algorithm tries to modify σ such that this side condition is satisfied.

Assume that $X_k \sqsubset^? E_k$ is this side condition. Since σ solves $X_k \sqsubseteq^? E_k$, we thus know that $\sigma(X_k) \equiv \sigma(E_k)$. Thus, we must either make $\sigma(X_k)$ more specific or $\sigma(E_k)$ more general. Since $\text{MATCH}_\mathcal{L}^\sqsubseteq$ computes the least solution, the first option cannot lead to a solution of the overall system. Hence, we

must try the second one. The idea (which will be explained in more detail later) is that we consider the reduced normal form of $\sigma(E_k)$. We try to make $\sigma(E_k)$ more general by (non-deterministically) choosing one word from one of its role languages and by removing this word by appropriately modifying the role languages of the variables occurring in E_k. Since we want to compute minimal matchers, we make as little changes as possible in order to keep the substitution as specific as possible.

The new substitution σ' obtained this way solves $X_k \sqsubset^? E_k$, and since we only modified variables occurring in E_k, the side conditions with larger index are still satisfied. However, the side conditions with smaller index (even the non-strict ones) as well as the matching problem need no longer be solved by σ'. To overcome this problem, $\text{MATCH}_{\mathcal{L}}^{\sqsubseteq}$ is used to compute the least substitution that (i) solves $\langle C \equiv^? D, S_{\sqsubseteq}\rangle$, and (ii) subsumes σ'. It can be shown that the second condition (which can be expressed by a system of matching problems) makes sure that the computed substitution still solves the strict subsumption conditions from index k to ℓ. We can now continue the modification process with this substitution.

Algorithm 47 *Let $M = \langle C \equiv^? D, S\rangle$ be an $\mathcal{L}$-matching problem under acyclic side conditions. Then,* $\text{MATCH}_{\mathcal{L}}^{\sqsubseteq}$ *works as follows:*

1. *If* $\text{MATCH}_{\mathcal{L}}^{\sqsubseteq}(\langle C \equiv^? D, S_{\sqsubseteq}\rangle)$ *returns "fail", then return "fail";*
2. $k := \ell$; $\sigma := \text{MATCH}_{\mathcal{L}}^{\sqsubseteq}\langle\{C \equiv^? D\}, S_{\sqsubseteq}\rangle$;
3. *If $k = 0$, then return σ;*
   If $\sigma(X_k)\ \rho_k\ \sigma(E_k)$, then continue with 5.
4. *Guess modification σ' of σ for $X_k \sqsubset^? E_k$;*
   If $\sigma'(E_k) \equiv \sigma(E_k)$, then return "fail";
   $M' := \langle\{C \equiv^? D\} \cup \{\sigma'(X_j) \sqsubseteq^? X_j \mid 1 \leq j \leq \ell\}, S_{\sqsubseteq}\rangle$;
   If $\text{MATCH}_{\mathcal{L}}^{\sqsubseteq}(M')$ *returns "fail", then return "fail";*
   $\sigma := \text{MATCH}_{\mathcal{L}}^{\sqsubseteq}(M')$
5. $k := k - 1$; *continue with 3.*

4.3 How to guess modifications

In order to introduce modifications, we first sketch the underlying idea for $\mathcal{FL}_\bot$. Recall that the goal is to make $\sigma(E_k)$ more general by (non-determinis-

tically) choosing one word w from one of its role languages and by removing this word by appropriately modifying the role languages of the variables occurring in E_k.

We call this a *$\mathcal{C}$-modification* if w is picked from a role language corresponding to some atomic concept A. In this case, removing certain words from role languages of the variables in E suffices to obtain a minimal modification.

In case of a *$\bot$-modification*, where w is picked from the role language corresponding to the $\bot$-concept, the removal of some word v in the role language of a variable implicitly removes every continuation vv' of v. To correct this effect, every word in $\{v\}\cdot N_R$ is put back whenever some v is removed. In addition, since v is also implicitly removed from role languages corresponding to atomic concepts, it is also transferred to such role languages. This ensures that the computed substitution is as specific as possible. This is vital both for the proof of correctness and to obtain all minimal solutions.

Before dealing with modifications in terms of a formal definition (see Definition 49), the following example illustrates in more detail how the modifications work.

Example 48 Consider the $\mathcal{FL}_\bot$-matching problem $A \sqcap \forall\{r,s\}.\bot \equiv^? X_1 \sqcap \forall r.X_2 \sqcap \forall r.X_3$ under the strict subsumption conditions $X_2 \sqsubset^? X_1,\ X_3 \sqsubset^? X_2$.

Executing the above algorithm, we obtain in Step 2 as initial solution σ the following substitution:

$$\{X_1 \mapsto \forall\{r,s\}.\bot \sqcap \forall\{\varepsilon\}.A, X_2 \mapsto \forall\{\varepsilon\}.\bot, X_3 \mapsto \forall\{\varepsilon\}.\bot\}.$$

The iteration begins in Step 3 by checking the second side condition, which is violated. Choosing a $\bot$-modification in Step 4, we must choose a word from the role language $\{\varepsilon\}$ corresponding to $\bot$ in $\sigma(X_2) = \sigma(X_3)$. In this case, we can only pick ε. To keep the change minimal, we do not simply remove it, but rather replace it by $\{r,s\}$ in the role language corresponding to $\bot$ in $\sigma(X_2)$. In addition, we transfer ε to the role language corresponding to A. This yields $\sigma'(X_2) = \forall\{r,s\}.\bot \sqcap \forall\{\varepsilon\}.A$. The other variables remain unchanged.

In this case, the substitution σ' itself solves the matching problem M' considered in Step 4, and thus $\text{MATCH}^{\sqsubseteq}_{\mathcal{FL}_\bot}(M')$ returns σ'.

In the second iteration, we find in Step 3 that the first side condition $X_2 \sqsubset^? X_1$ no longer holds. In Step 4, we again choose a $\bot$-modification,

and choose the word r from the role language $\{r, s\}$ corresponding to $\bot$ in $\sigma(X_1)$. The modification replaces r by rr, rs and adds r to the role language corresponding to A. This yields $\sigma'(X_1) := \forall\{rr, rs, s\}.\bot \sqcap \forall\{\varepsilon, r\}.A$. Again, this substitution solves M', and thus the new value of σ is σ'.

In the next iteration we have $k = 0$, ending the iteration in Step 3. The algorithm finally returns the substitution

$$\{X_1 \mapsto \forall\{rr, rs, s\}.\bot \sqcap \forall\{\varepsilon, r\}.A, X_2 \mapsto \forall\{r, s\}.\bot \sqcap \forall\{\varepsilon\}.A, X_3 \mapsto \forall\{\varepsilon\}.\bot\}.$$

Note that, in the first iteration step, it was not possible to apply a $\mathcal{C}$-modification since the role language corresponding to A was empty. In the second step, we could have applied a $\mathcal{C}$-modification, removing ε from the role language corresponding to A in $\sigma(X_1)$. Then, however, the system M' obtained this way would not have been solvable. In fact, it is easy to see that the two matching problems $A \sqcap \forall\{r, s\}.\bot \equiv^? X_1 \sqcap \forall r.X_2 \sqcap \forall r.X_3$ and $\forall\{r, s\}.\bot \sqsubseteq^? X_1$ occurring in M' cannot be solved simultaneously.

Modifications in $\mathcal{FL}_\bot$

In the following definition, modifications in $\mathcal{FL}_\bot$ are defined formally. Recall that in our matching problem $M := (C \equiv^? D, \{X_j \; \rho_j^? \; E_j \mid 1 \leq j \leq \ell\})$, C is assumed in U-labeled reduced normal form and D is assumed in V-labeled normal form. Furthermore, for the k-th side condition $X_k \; \rho_k^? \; E_k$, the concept description E_k is assumed in V^{X_k}-labeled normal form. Let σ_0 denote the substitution computed in Step 2 of the algorithm and denote by σ'_t $(t \geq 1)$ the respective modification computed in Step 4. Denote by σ_{t+1} the solution of $\text{MATCH}^{\sqsubseteq}_{\mathcal{FL}_\bot}(M')$ computed in the t-th iteration of the algorithm. For every t and for every variable X_k, assume $\sigma_t(X_k)$ in U^{t,X_k}-labeled reduced normal form. In the following definition, modifications need not be defined for the first side condition, because the acyclic structure implies that E_1 contains no variables.

Definition 49 *Guessing modifications in $\mathcal{FL}_\bot$*
Let $\mathcal{H} = \{\bot\} \cup \mathcal{C}$ and let $k \in \{2, \ldots, \ell\}$, where $\rho_k = \sqsubset$. Consider a reduced substitution σ_t with $\sigma_t(X_k) \equiv \sigma_t(E_k)$. A modification *$\sigma'_t$ of σ_t is defined by executing one of the following alternatives:*

- $\bot$-modification
  (Non-deterministically) guess one word $\hat{u} \in U_\bot^{t,X_k}$. For all $j \in \{1, \ldots, k-$

$1\}$, *compute*

$$W_\perp^j := \bigcup_{w \in V_{X_j}^{X_k}} w^{-1} \cdot \{\hat{u}\}$$

Thus, $W_\perp^j$ *contains all suffixes of* $\hat{u}$ *which yield* $\hat{u}$ *in the product* $V_j^{X_k} \cdot W_\perp^j$. *Define* σ'_t *by specifying the relevant role languages* $U_H'^{t,X_j}$ *for* $H \in \{\perp\} \cup \mathcal{C}$, *i.e. denote every* $\sigma'(X_j)$ *in* U'^{t,X_j}*-labeled normal form.*

1. $U_\perp'^{t,X_j} := (U_\perp^{t,X_j} \setminus W_\perp^j) \cup (U_\perp^{t,X_j} \cap W_\perp^j) \cdot N_R$
2. *For all* $A \in \mathcal{C}$, *define:* $U_A'^{t,X_j} := U_A^{t,X_j} \cup (W_\perp^j \cap U_\perp^{t,X_j})$

- $\mathcal{C}$-modification
  (Non-deterministically) guess one atomic concept $\hat{A} \in \mathcal{C}$. *For* $\hat{A}$, *guess one word* $\hat{u} \in U_{\hat{A}}^{t,X_k}$. *Using* $\hat{u}$, *for all* $j \in \{1, \ldots, k-1\}$ *compute* $W_A^j := \bigcup_{w \in V_{X_j}^{X_k}} w^{-1} \cdot \{\hat{u}\}$. *Then define:*
  $U_{\hat{A}}'^{t,X_j} := U_{\hat{A}}^{t,X_j} \setminus W_A^j$ *and* $U_H'^{t,X_j} := U_H^{t,X_j}$ *for all* $H \in \{\perp\} \cup \mathcal{C} \setminus \{\hat{A}\}$.

Soundness and completeness for $\mathcal{FL}_\perp$ and $\mathcal{FL}_\neg$ is proved in Section 4.4. NP-completeness is proved in Section **??**. We give two more examples in order to show that i) modifications deleting only one word do not always suffice and ii) matching in Step 4 of the algorithm MATCH$_\mathcal{L}^\sqsubseteq$ is necessary. For our examples, let $N_C = \{A\}$ and $N_R = \{R, S\}$.

Example 50 Consider the matching problem

$$\forall\{rrr, rrs, rs, srr\}.\perp \sqcap \forall\{rr, sr\}.A \equiv^? \forall rr.X_1 \sqcap \forall sr.X_2 \sqcap \forall r.X_3 \sqcap \forall r.X_4$$

under the following set of subsumption conditions.

$$\begin{aligned} &\{X_1 \sqsubseteq^? \forall\{r, s\}.\perp, \\ &\ X_3 \sqsubset^? \forall\{rs, s\}.X_1 \sqcap \forall r.X_2, \\ &\ X_4 \sqsubset^? \forall s.\perp \sqcap \forall\{\varepsilon, r\}.X_3\} \end{aligned}$$

Executing algorithm MATCH$_{\mathcal{FL}_\perp}^\sqsubseteq$ yields as initial solution σ in Step 2

$$\begin{aligned} &\{X_1 \mapsto \forall\{r, s\}.\perp \sqcap \forall\{\varepsilon\}.A, \\ &\ X_2 \mapsto \forall r.\perp \sqcap \forall\{\varepsilon\}.A, \\ &\ X_3 \mapsto \forall\{rr, rs, s\}.\perp \sqcap \forall r.A, \\ &\ X_4 \mapsto \forall\{rr, rs, s\}.\perp \sqcap \forall r.A\}. \end{aligned}$$

which violates the third side condition, as the test in Step 3 shows: $\sigma(X_4)$ is equivalent to $\sigma(\forall s.\bot \sqcap \forall\{\varepsilon, r\}.X_3)$. In Step 4, we choose a $\bot$-modification and pick the word rs from the role language $\{rr, rs, s\}$ corresponding to $\bot$ in $\sigma(X_4)$. Hence, we have $W^3_\bot = \{rs, s\}$, according to the definition of $\bot$-modifications. Thus, rs and s must be changed in the role language corresponding to $\bot$ in $\sigma(X_3)$. The modified solution σ' now yields

$$\sigma'(X_3) = \forall\{rr, rsr, rss, sr, ss\}.\bot \sqcap \forall\{r, rs, s\}.A,$$

while the other variables remain unchanged. We find that σ' solves the matching problem M' in Step 4, and thus $\text{MATCH}^{\sqsubseteq}_{\mathcal{FL}_\bot}(M')$ yields σ'.

In the second iteration we find in Step 3 that the second side condition is violated, since $\sigma(X_3)$ is equivalent to $\sigma(\forall\{rs, s\}.X_1 \sqcap \forall r.X_2)$. We choose a $\mathcal{C}$-modification and pick the word rs from the role language $\{r, rs, s\}$ corresponding to A in $\sigma(X_3)$. This yields $W^1_A = \{\varepsilon\}$ and $W^2_A = \{s\}$. Nevertheless, the role language $\{\varepsilon\}$ corresponding to A in $\sigma(X_2)$ does not contain the word s, while $\{\varepsilon\}$ corresponding to A in $\sigma(X_1)$ obviously contains ε. We therefore have

$$\sigma'(X_1) = \forall\{r, s\}.\bot,$$

while the other variables remain unchanged. Again σ' solves the matching problem M' in Step 4, so that we have σ' as new substitution σ. In the third iteration, we now find in Step 3 that the first side condition holds, so that the final result is the following.

$$\begin{aligned}
&\{X_1 \mapsto \forall\{r, s\}.\bot,\\
&\ X_2 \mapsto \forall r.\bot \sqcap \forall\{\varepsilon\}.A,\\
&\ X_3 \mapsto \forall\{rr, rsr, rss, sr, ss\}.\bot \sqcap \forall\{r, rs, s\}.A,\\
&\ X_4 \mapsto \forall\{rr, rs, s\}.\bot \sqcap \forall r.A\}
\end{aligned}$$

A closer examination reveals that for the third side condition, neither picking any word other than rs from $\{r, rs, s\}$ in the $\bot$-modification, nor performing a $\mathcal{C}$-modification would have been successful. Similarly, in the second side condition only a $\mathcal{C}$-modification is successful. Nevertheless, here we could have picked the word s instead of rs, which would not have altered the solution, though.

The previous two examples might raise the question whether or not solving the matching problem M' in Step 4 of every iteration of the the algorithm

$\text{MATCH}^{\sqsubseteq}_{\mathcal{FL}_\bot}$ is necessary at all. The following example shows that there are cases where matching is needed.

Example 51 We examine the matching problem

$$\forall\{rrr, rrs\}.\bot \sqcap \forall rr.A \equiv^? \forall rr.X_1 \sqcap \forall r.X_2 \sqcap X_3 \sqcap X_4$$

under the following set of subsumption conditions.

$$\{X_3 \sqsubset^? \forall rr.X_1 \sqcap \forall r.X_2,$$
$$X_4 \sqsubset^? X_3\}$$

Executing algorithm $\text{MATCH}^{\sqsubseteq}_{\mathcal{FL}_\bot}$ again begins by computing an initial solution σ in Step 2, yielding the following substitution.

$$\{X_1 \mapsto \forall\{r, s\}.\bot \sqcap A,$$
$$X_2 \mapsto \forall\{rr, rs\}.\bot \sqcap \forall r.A,$$
$$X_3 \mapsto \forall\{rrr, rrs\}.\bot \sqcap \forall\{rr\}.A,$$
$$X_4 \mapsto \forall\{rrr, rrs\}.\bot \sqcap \forall\{rr\}.A\}$$

Obviously, in Step 3 we find that the second side condition is violated, making it necessary to modify the role languages of $\sigma(X_3)$, so that $\sigma(X_3) \sqsubset \sigma'(X_3)$.

Nevertheless, for the initial solution σ we also find that the first side condition is violated as well, since $\sigma(X_3)$ is equivalent to $\sigma(\forall rr.X_1 \sqcap \forall r.X_2)$. As a consequence, any successful modification will result in a substitution σ' with $\sigma'(X_3) \not\equiv \sigma'(\forall rr.X_1 \sqcap \forall r.X_2)$. Hence, σ' can be no solution to the matching problem M' in Step 4.

The above examples may suffice to give a rough impression of the algorithm $\text{MATCH}^{\sqsubseteq}_{\mathcal{FL}_\bot}$. We now introduce modifications for $\mathcal{FL}_\neg$.

Modifications in $\mathcal{FL}_\neg$

The modification strategy for $\text{MATCH}^{\sqsubseteq}_{\mathcal{FL}_\neg}$ differs from the previous definition for $\mathcal{FL}_\bot$ in three ways. Here, inconsistencies can not only be introduced by role languages referring to the $\bot$-concept, but also by interactions between role languages referring to an atomic concept A and its negation $\neg A$.

Consequently, removing the set $W^j_\bot$ from role languages referring to the $\bot$-concept alone does not suffice for $\bot$-modifications. Furthermore, a $\bot$-modification can no longer add the intersection $W^j_\bot \cap U^{t,X_j}_\bot$ to every role

language of the form U_H^{t,X_j}, where $H \neq \bot$. In this case, $W_\bot^j \cap U_\bot^{t,X_j}$ would appear in U_A^{t,X_j} as well as in $U_{\neg A}^{t,X_j}$ for every $A \in \mathcal{C}$, rendering the removal from all role languages referring to the $\bot$-concept useless. For $\bot$-modifications in $\mathcal{FL}_\neg$, we non-deterministically choose a subset of $W_\bot^j \cap U_\bot^{t,X_j}$ to be added to the role languages of the form U_H^{t,X_j}.

For $\mathcal{C}$-modifications, the non-deterministic choice of an atomic concept $\hat{A}$ must be generalized to all concepts in $\mathcal{C} \cup \{\neg A \mid A \in \mathcal{C}\}$. With these two changes we obtain the following definition for modifications in $\mathcal{FL}_\neg$.

Definition 52 *Guessing modifications in $\mathcal{FL}_\neg$*
Let $\mathcal{H} = \{\bot\} \cup \mathcal{C} \cup \{\neg A \mid A \in \mathcal{C}\}$ and let

$$X_k \sqsubset^? \underbrace{\bigsqcap_{H \in \mathcal{H}} \forall V_H^{X_k}.H \;\sqcap\; \bigsqcap_{j=1}^{k-1} \forall V_{X_j}^{X_k}.X_j}_{E_k}$$

be the k-th side condition in an $\mathcal{FL}_\neg$-matching problem with strict acyclic side conditions over the variables $\{X_1, \ldots, X_\ell\}$, where $H \in \mathcal{H}$ and $1 \leq k \leq \ell$. We again consider a reduced substitution σ_t with $\sigma_t(X_k) \equiv \sigma_t(E_k)$, where every $\sigma_t(X_j)$ is denoted in U^{t,X_j}-labeled normal form. A modification *σ_t' of σ_t is defined by executing one of the following alternatives:*

- $\bot$-modification
  (Non-deterministically) guess one word $\hat{u} \in U_\bot^{t,X_k}$. For all $j \in \{1, \ldots, k-1\}$, compute

$$W_\bot^j := \bigcup_{w \in V_{X_j}^{X_k}} w^{-1} \cdot \{\hat{u}\}$$

  Thus, $W_\bot^j$ contains all suffixes of $\hat{u}$ which yield $\hat{u}$ in the product $V_j^{X_k} \cdot W_\bot^j$. Define σ_t' by specifying the relevant role languages $U_H'^{t,X_j}$ for $H \in \{\bot\} \cup \mathcal{C}$, i.e. denote every $\sigma'(X_j)$ in U'^{t,X_j}-labeled normal form.

  1. $U_\bot'^{t,X_j} := (U_\bot^{t,X_j} \setminus W_\bot^j) \cup (U_\bot^{t,X_j} \cap W_\bot^j) \cdot N_R$
  2. *For all $H \in \mathcal{H} \setminus \{\bot\}$, (non-deterministically) choose a subset $\hat{W}^j \subseteq W_\bot^j \cap U_\bot^{t,X_j}$. Then define:*
     $U_H'^{t,X_j} := \big(U_H^{t,X_j} \setminus (U_H^{t,X_j} \cap U_{\neg H}^{t,X_j} \cap W_\bot^j)\big) \cup \hat{W}^j$

- $\mathcal{C}$-modification
  (Non-deterministically) guess one atomic concept $\hat{H} \in \mathcal{C} \cup \{\neg A \mid A \in \mathcal{C}\}$. *For* $\hat{H}$, *guess one word* $\hat{u} \in U_{\hat{H}}^{t,X_k}$. *Using* $\hat{u}$, *for all* $j \in \{1, \ldots, k-1\}$ *compute* $W_{\hat{H}}^{j} := \bigcup_{w \in V_{X_j}^{X_k}} w^{-1} \cdot \{\hat{u}\}$. *Then define:*
  $U_{\hat{H}}^{\prime t,X_j} := U_{\hat{H}}^{t,X_j} \setminus W_{\hat{H}}^{j}$ *and*
  $U_{H}^{\prime t,X_j} := U_{H}^{t,X_j}$ *for all* $H \in \mathcal{H} \setminus \{\hat{H}\}$.

In the next section we will prove the algorithms $\text{MATCH}^{\sqsubseteq}_{\mathcal{FL}_\bot}$ and $\text{MATCH}^{\sqsubseteq}_{\mathcal{FL}_\neg}$ to be correct. Knowing that the algorithm $\text{MATCH}^{\sqsubseteq}_{\mathcal{L}}$ always terminates it is easy to see that termination also holds for $\text{MATCH}^{\sqsubseteq}_{\mathcal{L}}$, where a fixed number of matching problems under subsumption conditions are solved. For this reason we do not need to address the question of termination separately.

4.4 Soundness and completeness

With a formal definition of modifications, we are now ready to prove soundness and completess of the algorithm. We first address the case $\mathcal{L} = \mathcal{FL}_\bot$.

Soundness and completeness in $\mathcal{FL}_\bot$

In preparation, we need to introduce some notation which simplifies denoting the role words assigned to a concept pattern for some atomic concept.

Definition 53 *Notation*
Let E *be an* $\mathcal{FL}_\bot$*-concept pattern in* V*-labeled normal form over the role alphabet* N_R *and the set* $\mathcal{X}$ *of variables, i.e.*

$$E := \bigsqcap_{H \in \mathcal{H}} \forall U_H.H \sqcap \bigsqcap_{X \in \mathcal{X}} \forall V_X.X,$$

where $\mathcal{H} := \{\bot\} \cup \mathcal{C}$. *For a substitution* σ *and for all* $X \in \mathcal{X}$, *denote* $\sigma(X)$ *in* U^X*-labeled normal form. For every* $H \in \mathcal{H}$, *define*

$$\sigma(E)|_H := U_H \cup \bigcup_{X \in \mathcal{X}} V_X \cdot U_H^X$$

With the above notation, we can write $\sigma(E)$ as $\forall \sigma(E)|_\bot.\bot \sqcap \bigsqcap_{A \in \mathcal{C}} \forall \sigma(E)|_A.A$.

It is shown next that the modification strategy defined for $\mathcal{FL}_\bot$ in Definition 49 does produce a strict solution for the relevant side condition. Hence,

if for some side condition $X_k \sqsubset^? E_k$ occurring in a solvable matching problem it holds that $\sigma_t(X_k) \equiv \sigma_t(E_k)$, then there is a modification yielding $\sigma \sqsubset \sigma'$ and $\sigma'(X_k) \sqsubset \sigma'_t(E_k)$.

Lemma 54 *Strictness of modifications in $\mathcal{FL}_\bot$*
Let $\sigma_\sqsubset$ be a reduced solution to M, let σ_t be a reduced substitution with $\sigma_t(X_k) \equiv \sigma_t(E_k)$ for some $k \in \{2, \ldots, \ell\}$ with $\rho_k = \sqsubset$. Let $\sigma_t \sqsubset \sigma_\sqsubset$. Then (non-deterministically) modifying σ_t to σ'_t yields $\sigma \sqsubset \sigma'$ and $\sigma'_t(X_k) \sqsubset \sigma'_t(E_k)$.

Proof. Two steps are sufficient to prove the claim: i) every modification in accordance with Definition 49 yields $\sigma \sqsubseteq \sigma'$ and ii) there exists a modification such that $\sigma_t(X_k) \sqsubset \sigma'_t(E_k)$.

i) $\bot$-modification: For every choice of $\hat{u}$ and for every j it holds for σ'_t that

$$U_\bot'^{t,X_j} \cdot N_R^* = U_\bot^{t,X_j} \cdot N_R^* \setminus W_\bot^j,$$

implying $U_\bot^{t,X_j} \cdot N_R^* \supset U_\bot'^{t,X_j} \cdot N_R^*$. For every $A \in \mathcal{C}$ the inclusion $\hat{W}_A^j \subseteq U_\bot^{t,X_j}$ furthermore implies

$$U_A^{t,X_j} \cup U_\bot^{t,X_j} \cdot N_R^* \supseteq U_A'^{t,X_j} \cup U_\bot'^{t,X_j} \cdot N_R^*$$

because every word possibly gained by $U_A'^{t,X_j}$ is contained in $U_\bot^{t,X_j} \cdot N_R^*$. Consequently, we obtain $\sigma_t \sqsubseteq \sigma'_t$. The second part of the claim, which is addressed below in (ii), is sufficient for strictness.

$\mathcal{C}$-modification: The only difference between σ_t and σ'_t is the deletion of words in role languages referring to an atomic concept $\hat{A} \in \mathcal{C}$. It is therefore not difficult to see that $\sigma_t \sqsubseteq \sigma'_t$ holds.

ii) We now present a guessing strategy to find a modification σ' with $\sigma'_t(X_k) \sqsubset \sigma'_t(E_k)$. To this end, two cases are distinguished.

(Case 1): $\sigma_t(E_k)$ and $\sigma_\sqsubset(E_k)$ disagree on the $\bot$-languages, i.e.

$$U_\bot^{t,X_k} \cdot N_R^* \;=\; \sigma_t(E_k)|_\bot \cdot N_R^* \;\supset\; \sigma_\sqsubset(E_k)|_\bot \cdot N_R^*.$$

Thus, there are $\hat{u} \in U_\bot^{t,X_k}$ and $x \in N_R^*$ such that $\hat{u}x$ does not occur on the right-hand side of the inclusion. Consequently, $\hat{u} \notin \sigma_\sqsubset(E_k)|_\bot \cdot N_R^*$. Construct σ' by a $\bot$-modification, picking one word $\hat{u}$ as introduced above. By definition, we then have

$$U_\bot'^{t,X_j} = (U_\bot^{t,X_j} \setminus W_\bot^j) \cup (U_\bot^{t,X_j} \cap W_\bot^j) \cdot N_R,$$

where $W_\perp^j = \bigcup_{w \in V_{X_j}^{X_k}} w^{-1} \cdot \{\hat{u}\}$.

(Case 2): $\sigma_t(E_k)$ and $\sigma_\sqsubset(E_k)$ agree on the $\perp$-languages in the sense that

$$U_\perp^{t,X_k} \cdot N_R^* = \sigma_t(E_k)|_\perp \cdot N_R^* = \sigma_\sqsubset(E_k)|_\perp \cdot N_R^*.$$

As $\sigma_t(E_k) \sqsubset \sigma_\sqsubset(E_k)$, this implies that there is an $A \in \mathcal{C}$ and a word $\hat{u} \in U_A^{t,X_k}$ such that $\hat{u} \notin \sigma_\sqsubset(E_k)|_A$. For the modification, choose $\hat{A} := A$ and use one word $\hat{u}$ as introduced above.

It is to show now that both in both cases we have $\sigma_t'(X_k) \sqsubset \sigma_t'(E_k)$.

In Case 1, the definition of $W_\perp^j$ implies for the $\perp$-part of $\sigma'(E_k)$ that

$$\sigma_t'(E_k)|_\perp \cdot N_R^* = \sigma_t(E_k)|_\perp \cdot N_R^* \setminus \bigcup_{j=1}^{k-1} V_{X_j}^{X_k} \cdot W_\perp^j$$

The word $\hat{u}$ occurs both in $\sigma_t(E_k)|_\perp \cdot N_R^*$ and in at least one product $V_{X_j}^{X_k} \cdot W_\perp^j$. Since $U_\perp'^{t,X_k} = U_\perp^{t,X_k}$, and since $U_\perp^{t,X_k} \cdot N_R^* = \sigma_t(E_k)|_\perp \cdot N_R^*$, we obtain

$$U_\perp'^{t,X_k} \cdot N_R^* \supset \sigma_t'(E_k)|_\perp \cdot N_R^*. \qquad (*)$$

This is sufficient for our claim, since we have shown in (1) that $\sigma_t \sqsubseteq \sigma_t'$.

In Case 2, the definition of σ_t' ensures that for $\hat{A}$ we have

$$\sigma_t'(E_k)|_{\hat{A}} = \sigma_t(E_k)|_{\hat{A}} \setminus \bigcup_{j=1}^{k-1} V_{X_j}^{X_k} \cdot W_\perp^j.$$

The word $\hat{u}$ occurs in $\sigma_t(E_k)|_{\hat{A}}$, since $U_{\hat{A}}^{t,X_k} = pf(\sigma_t(E_k)|_{\hat{A}}) \subseteq \sigma_t(E_k)|_{\hat{A}}$, [3] and occurs in at least one product $V_{X_j}^{X_k} \cdot W_\perp^j$, because otherwise $\hat{u} \notin \sigma_t(E_k)|_{\hat{A}}$. Thus, $\hat{u} \notin \sigma_t'(E_k)|_{\hat{A}}$. We therefore obtain

$$\sigma_t(E_k)|_{\hat{A}} \cup \sigma_t(E_k)|_\perp \cdot N_R^* \supset \sigma_t'(E_k)|_{\hat{A}} \cup \sigma_t'(E_k)|_\perp \cdot N_R^*$$

The inclusion is strict, because otherwise $\hat{u} \in \sigma_t(E_k)|_\perp \cdot N_R^*$, implying $\hat{u} \in U_\perp^{t,X_k} \cdot N_R^*$ in contradiction to the reducedness of σ_t. Together with (1), this concludes our proof. ∎

[3] Recall that *pf* makes a formal language prefix free, as defined in Definition 18.

It has to be shown next that a modification yielding σ'_t is minimal in the sense that no other modification τ at the same time i) lies between σ_t and σ'_t in respect to the strict ordering $\sqsubset$ on substitutions, i.e. if $\sigma_t \sqsubset \tau \sqsubset \sigma'_t$, and ii) also yields strictness for the respective side condition, i.e. $\sigma_t(E_k) \sqsubset \tau(E_k)$. This property justifies that in the algorithm $\text{MATCH}^{\sqsubseteq}_{\mathcal{FL}_0}$ no modification tries to make $\sigma(X_k)$ more specific when modifying a side condition $X_k \sqsubset^? E_k$ with $\sigma(X_k) \equiv \sigma(E_k)$. The following lemma provides the necessary result, again recurring to the matching problem as introduced at the beginning of this section.

Lemma 55 *Minimality of modifications*
Consider a substitution τ such that $\sigma_t \sqsubset \tau \sqsubseteq \sigma'_t$ and $\sigma_t(E_k) \sqsubset \tau(E_k)$.
Then, $\tau \equiv \sigma'_t$.

Proof. Without loss of generality, we may assume σ_t, τ, and σ'_t to be reduced. Two cases are distinguished depending on whether σ'_t was obtained by a $\bot$- or a $\mathcal{C}$-modification.

$\bot$-modification: Then there exists a word $\hat{u} \in \sigma_t(E_k)|_\bot$ such that $\hat{u} \notin \sigma'_t(E_k)|_\bot$. There are two possible reasons for $\sigma_t(E_k) \sqsubset \tau(E_k)$ to hold:

Case 1: $\sigma_t(E_k)|_\bot \cdot N_R^* \supset \tau(E_k)|_\bot \cdot N_R^*$. Since $\tau \sqsubseteq \sigma'_t$, this implies that the difference $\sigma_t(E_k)|_\bot \cdot N_R^* \setminus \tau(E_k)|_\bot \cdot N_R^*$ must be missing in $\sigma'_t(E_k)|_\bot \cdot N_R^*$ as well. But $\sigma'_t(E_k)|_\bot \cdot N_R^*$ was obtained from $\sigma_t(E_k)|_\bot \cdot N_R^*$ by removing as little as possible to remove the word $\hat{u}$. Hence, in $\tau(E_k)|_\bot \cdot N_R^*$ no other words could have been removed, because otherwise either $\sigma'_t(E_k)|_\bot \cdot N_R^*$ would be too small or $\sigma_t(E_k) \sqsubset \tau(E_k)$ could not hold. This implies $\tau(E_k)|_\bot \cdot N_R^* = \sigma'_t(E_k)|_\bot \cdot N_R^*$ For every $A \in \mathcal{C}$, we therefore have $\tau(E_k)|_A \supseteq \sigma'_t(E_k)|_A$. On the other hand, we know that the $\bot$-modification has increased $\sigma'_t(E_k)|_A$ by $W_\bot^j \cap U_\bot^{t,X_j}$, yielding $\sigma'_t(E_k)|_A \supset \sigma'_t(E_k)|_A$. This implies $\tau(E_k)|_A = \sigma'_t(E_k)|_A$, because otherwise the $\bot$-modification would not have added all of $W_\bot^j \cap U_\bot^{t,X_j}$ to the role languages referring to A in all concept descriptions $\sigma'_t(X_j)$. Together with $\tau \sqsubseteq \sigma'_t$ and the reducedness of the substitutions this is sufficient for $\tau \equiv \sigma'_t$.

Case 2: $\sigma_t(E_k)|_\bot \cdot N_R^* = \tau(E_k)|_\bot \cdot N_R^*$. Thus, i) there is some $A \in \mathcal{C}$ such that $\sigma_t(E_k)|_A \supset \tau(E_k)|_A$ a and ii) we have $\tau(E_k)|_\bot \cdot N_R^* \supset \sigma'_t(E_k)|_\bot \cdot N_R^*$ as a consequence of the $\bot$-modification for σ'_t. It can be shown that this implies a contradiction with $\tau \sqsubseteq \sigma'_t$, because $\tau(E_k)|_A$ contains not enough words for $\tau(E_k)|_A \cup \tau(E_k)|_\bot \cdot N_R^* \supseteq \tau(E_k)|_A \cup \tau(E_k)|_\bot \cdot N_R^*$ to hold, which is a necessary condition, as seen in the characterization of subsumption.

$\mathcal{C}$-modification: Then there exists an $\hat{A} \in \mathcal{C}$ and a word $\hat{u} \in \sigma_t(E_k)|_{\hat{A}}$ such that $\hat{u} \notin \sigma'_t(E_k)|_\perp$. For τ, we again have two cases to distinguish:

Case 1: $\sigma_t(E_k)|_\perp \cdot N_R^* \supset \tau(E_k)|_\perp \cdot N_R^*$. This implies a contradiction with the fact that the $\mathcal{C}$-modification did not alter role languages in σ'_t referring to the $\perp$-concept, which implies $\sigma_t(E_k)|_\perp \cdot N_R^* = \sigma'_(E_k)|_\perp \cdot N_R^*$. Together with $\tau(E_k)|_\perp \cdot N_R^* \supseteq \sigma'_t(E_k)|_\perp \cdot N_R^*$, this forbids the assumption of Case 1.

Case 2: $\sigma_t(E_k)|_\perp \cdot N_R^* = \tau(E_k)|_\perp \cdot N_R^*$. Again, this implies some $A \in \mathcal{C}$ such that $\sigma_t(E_k)|_A \supset \tau(E_k)|_A$. Since we know that $\tau(E_k) \sqsubset \sigma'_t(E_k)$ and since the $\mathcal{C}$-modification is the only difference between $\sigma_t(E_k)$ and $\sigma'_t(E_k)$, we can conclude that $A = \hat{A}$. On the one hand one can see that as few as possible words are removed in $\sigma'_t(E_k)|_A$ to gain strictness while on the other hand $\tau(E_k)|_A \supseteq \sigma'_t(E_k)|_A$. Together with $\tau \sqsubseteq \sigma'_t$ and the reducedness of the substitutions we get $\tau \equiv \sigma'_t$. ∎

We are now prepared to prove soundness of the algorithm $\text{MATCH}^{\sqsubseteq}_{\mathcal{FL}_\perp}$. To this end, we need to make sure that side conditions remain valid once they are modified appropriately.

Lemma 56 *Soundness*

1. *For every t and for every modification of σ_t yielding σ'_t it holds that if $\text{MATCH}^{\sqsubseteq}_{\mathcal{FL}_\perp}(M')$ succeeds in Step 4 of the algorithm, then*
   - *i) $\sigma_t(X_j) = \sigma_{t+1}(X_j)$ for every $j \in \{k, \ldots, \ell\}$*
   - *ii) $\sigma_t \sqsubset \sigma'_t \sqsubseteq \sigma_{t+1}$.*
2. *If $\text{MATCH}^{\sqsubseteq}_{\mathcal{FL}_\perp}(M)$ returns the substitution σ, then σ solves M (soundness).*

PROOF. 1. i) According to Definition 49, for $j \in \{k, \ldots, \ell\}$ the substitution σ'_t assigns the same values to every variable X_j as σ_t does. Due to Lemma 54, the right-hand side of every side condition can only become more general. Consequently, every value assigned to variables X_k to X_ℓ by σ'_t is also a solution for the matching problem defined for σ_{t+1}. Exploiting the minimality of $\text{MATCH}^{\sqsubseteq}_\perp$ and the assumption of reduced normal forms concludes the argument.

ii) It was shown in Lemma 54 that $\sigma_t \sqsubset \sigma'_t$ holds for every modification σ'_t. Subsumption $\sigma'_t \sqsubseteq \sigma_{t+1}$ obviously holds because of the matching problems modulo subsumption $\{\sigma'_t(X_j) \sqsubseteq^? X_j | 1 \leq j \leq \ell\}$ which are included in the matching problem M' for σ_{t+1}.

2. Assume that $\text{MATCH}^{\sqsubseteq}_{\perp}(M) = \sigma$. Hence, σ is the solution of a matching problem solved in Step 2 or Step 4 of the algorithm. In both cases, obviously $C \equiv^? D$ holds. Furthermore, the initial solution computed in Step 2 also respects $S_{\sqsubseteq}$, where every side condition from M is non-strict. If the execution of the algorithm has succeeded, then in every iteration from ℓ to 1 either the k-th side condition was found valid in Step 3 or guessing a modification in Step 4 has succeeded. It is obvious that in both cases every strict subsumption condition under consideration has been met *in the respective iteration.*

   As $S_{\sqsubseteq}$ is acyclic, we find as a consequence of Part (1) that once a side condition is met it remains valid in subsequent iterations of the algorithm. This holds for two reasons: i) the variables not modified by a modification σ'_t remain unchanged in σ'_{t+1} and ii) the variables which are modified are assigned more general concept descriptions. Consequently, if every iteration is successful, then finally every side condition is met by the resulting substitution σ. ∎

In order to prove completeness, it is sufficient to show that the algorithm $\text{MATCH}^{\sqsubseteq}_{\mathcal{FL}_\perp}(M)$ successfully returns a solution if the input matching problem M is solvable.

Lemma 57 *Completeness*
Let $\sigma_{\sqsubseteq}$ be a reduced solution to M.

1. *Then for every t there exists a modification for σ_t yielding σ'_t, such that:*
   i) If $\sigma'_t \sqsubseteq \sigma_{\sqsubseteq}$ then $\sigma_{t+1} \sqsubseteq \sigma_{\sqsubseteq}$.
   ii) $\sigma'_t \sqsubseteq \sigma_{\sqsubseteq}$
2. *$\text{MATCH}^{\sqsubseteq}_{\mathcal{FL}_\perp}(M)$ returns a substitution σ which solves M (completeness).*

PROOF. 1. i) Presupposing $\sigma'_t \sqsubseteq \sigma_{\sqsubseteq}$ it is not difficult to see that $\sigma_{\sqsubseteq}$ is also a valid solution to the matching problem defined in the algorithm for σ_{t+1}. The additional requirements for σ_{t+1} are $\{\sigma'_t(X_j) \sqsubseteq^? X_j | 1 \leq j \leq \ell\}$ which are met by $\sigma_{\sqsubseteq}$ due to $\sigma'_t \sqsubseteq \sigma_{\sqsubseteq}$. The minimality of the matching algorithm $\text{MATCH}^{\sqsubseteq}_{\mathcal{FL}_\perp}$ then guarantees that $\sigma_{t+1} \sqsubseteq \sigma_{\sqsubseteq}$.

ii) For every $j \in \{1, \ldots, \ell\}$, denote $\sigma_{\sqsubseteq}(X_j)$ in $U^{\sqsubseteq, X_j}$-labeled normal form. Proof by induction over t.

($t = 0$): Again, we begin by considering a $\perp$-modification as introduced in the second part of Lemma 54. Due to the minimality of $\text{MATCH}^{\sqsubseteq}_{\mathcal{FL}_\perp}$

it holds that $\sigma_0 \sqsubseteq \sigma_\sqsubset$. This implies $U_\perp^{0,X_j} \cdot N_R^* \supseteq U_\perp^{\sqsubset,X_j} \cdot N_R^*$. As $\hat{u} \notin \sigma_\sqsubset(E_k)|_\perp \cdot N_R^*$, no product $U_\perp^{\sqsubset,X_j} \cdot N_R^*$ contains words from $W_\perp^j$. Since $U_\perp'^{0,X_j} \cdot N_R^* = U_\perp^{0,X_j} \cdot N_R^* \setminus W_\perp^j$, we obtain $U_\perp'^{0,X_j} \cdot N_R^* \supseteq U_\perp^{\sqsubset,X_j} \cdot N_R^*$, which is the first condition for subsumption.

For $A \in \mathcal{C}$, a $\perp$-modification obviously guarantees that $U_A^{0,X_j} \subseteq U_A'^{0,X_j}$. We also know from (i) that $U_\perp^{0,X_j} \cdot N_R^* \supseteq U_\perp'^{0,X_j} \cdot N_R^*$. As $\sigma_0 \sqsubseteq \sigma_\sqsubset$ implies

$$U_A^{0,X_j} \cup U_\perp^{0,X_j} \cdot N_R^* \supseteq U_A^{\sqsubset,X_j} \cup U_\perp^{\sqsubset,X_j} \cdot N_R^*, \qquad (*)$$

we may replace U_A^{0,X_j} by $U_A'^{0,X_j}$. Now, why may we also replace the product $U_\perp^{0,X_j} \cdot N_R^*$ by the—smaller—language $U_\perp'^{0,X_j} \cdot N_R^*$? The language $U_\perp'^{0,X_j} \cdot N_R^*$ does not contain a word from $W_\perp^j$. We already know that $U_\perp^{\sqsubset,X_j} \cdot N_R^*$ does not either, so the only problem could be $U_A^{\sqsubset,X_j}$ containing words from $W_\perp^j$. But since $U_A'^{0,X_j}$ is defined as $U_A^{t,X_j} \cup (U_A^{\sqsubset,X_j} \cap W_\perp^j)$, this case is covered. This completes the proof for $\perp$-modifications.

For $\mathcal{C}$-modifications, we only have to consider the second condition for subsumption, because role languages referring to the bottom concept remain unchanged. For all $A \neq \hat{A}$, nothing changes as well. For $\hat{A}$, only those words of U_A^{0,X_j} are missing in $U_A'^{0,X_j}$ which do not occur in $U_A^{\sqsubset,X_j}$ also. Consequently, starting from equation $(*)$ again we obtain the result sought.

$(t+1)$: The induction hypothesis states that $\sigma_t' \sqsubseteq \sigma_\sqsubset$. Due to (i), this implies $\sigma_{t+1} \sqsubseteq \sigma_\sqsubset$. With these findings the remaining proof is analogous to the previous case $t = 0$.

2. If M is solvable, the matching problem $\langle C \equiv^? D, S_\sqsubseteq \rangle$ computed in Step 2 of the algorithm is solvable as well. Furthermore, we know from Theorem 43 that the solutions computed by $\text{MATCH}_{\mathcal{FL}_\perp}^{\sqsubseteq}$ are least matchers with respect to the ordering $\sqsubseteq$ on substitutions. Hence, for the initial solution σ it holds that $\sigma \sqsubset \sigma_\sqsubset$. Inductively, we can now exploit the results of Lemma 54 and Part (1): Lemma 54 guarantees that the modification probably necessary in the first iteration of the algorithm succeeds. According to (ii), for the first modification σ_1' we also have $\sigma_1' \sqsubset \sigma_\sqsubset$, which by (i) implies $\sigma_2 \sqsubseteq \sigma_\sqsubset$. If σ_2 does not solve M, then obviously we have $\sigma_2 \sqsubset \sigma_\sqsubset$. Hence, in the next iteration we can inductively apply the same argument.

Consequently, we end up with a successful computation yielding a solution σ with $\sigma \sqsubseteq \sigma_{\sqsubset}$. ∎

As a consequence of the previous lemma, all minimal matchers (w.r.t. subsumption of substitutions) are computed. This can be readily seen when using a minimal solution $\sigma_{\sqsubset}$ in the previous lemma, which then implies that the solution computed by $\text{MATCH}^{\sqsubseteq}_{\mathcal{FL}_\bot}$ is equivalent to $\sigma_{\sqsubset}$.

Soundness and completeness in $\mathcal{FL}_\neg$

In $\mathcal{FL}_\neg$-concept descriptions, inconsistencies can additionally be introduced by words occurring in role languages referring to an atomic concept and to its negation. We need to alter the notation $\sigma(E)|_H$ introduced in the previous part in order to respect this effect for $H = \bot$.

Definition 58 *Notation*
Let E be an $\mathcal{FL}_\neg$-concept pattern in V-labeled normal form over the role alphabet N_R and the set $\mathcal{X}$ of variables. For a substitution σ and for all $X \in \mathcal{X}$, denote $\sigma(X)$ in U^X-labeled normal form. Then, define

$$\sigma(E)|_\bot := U_\bot \cup \bigcup_{X\in\mathcal{X}} V_X{\cdot}U^X_\bot \cup \bigcup_{A\in\mathcal{C}}\Big(\bigcup_{X\in\mathcal{X}}(V_X{\cdot}U^X_A) \cap \bigcup_{X\in\mathcal{X}}(V_X{\cdot}U^X_{\neg A})\Big)$$

In Lemma 59 we could prove for every possible modification σ'_t that $\sigma_t \sqsubseteq \sigma'_t$. In case of $\mathcal{FL}_\neg$, this is no longer possible, because we depend stronger on the properties of a strict solution $\sigma_{\sqsubseteq}$. In the followng lemma we therefore begin by specifying a guessing strategy relative to $\sigma_{\sqsubset}$.

Lemma 59 *Strictness of Modifications in $\mathcal{FL}_\neg$*
Let $\sigma_{\sqsubset}$ be a reduced solution to M, let σ_t be a reduced substitution with $\sigma_t(X_k) \equiv \sigma_t(E_k)$ for some $k \in \{2,\ldots,\ell\}$ with $\rho_k = \sqsubset$. Let $\sigma_t \sqsubset \sigma_{\sqsubset}$. Then (non-deterministically) modifying σ_t to σ'_t yields $\sigma \sqsubset \sigma'$ and $\sigma'_t(X_k) \sqsubset \sigma'_t(E_k)$.

Proof. We show that there exists a modification in accordance with Definition 52 such that $\sigma \sqsubseteq \sigma'$ and $\sigma'_t(X_k) \sqsubset \sigma'_t(E_k)$. To this end, we present a guessing strategy to find an appropriate modification σ', distinguishing two cases.

(Case 1): $\sigma_t(E_k)$ and $\sigma_{\sqsubset}(E_k)$ disagree on the $\bot$-languages, i.e.

$$U^{t,X_k}_\bot{\cdot}N^*_R \;=\; \sigma_t(E_k)|_\bot{\cdot}N^*_R \;\supset\; \sigma_{\sqsubset}(E_k)|_\bot{\cdot}N^*_R.$$

This situation is analogous to the case $\mathcal{FL}_\bot$, because the left-hand side of the equation is reduced, which forbids inconsistencies being introduced by interactions of atomic concepts and their negations. It is therefore sufficient to restrict the choice of $\hat{u}$ to the role language $U_\bot^{t,X_k}$. Hence, we again construct σ' by a $\bot$-modification, picking one word $\hat{u}$ as introduced above. By definition, we then have

$$U_\bot'^{t,X_j} = (U_\bot^{t,X_j} \setminus W_\bot^j) \cup (U_\bot^{t,X_j} \cap W_\bot^j) \cdot N_R,$$

where $W_\bot^j = \bigcup_{w \in V_{X_j}^{X_k}} w^{-1} \cdot \{\hat{u}\}$. Nevertheless, an additional guess is necessary for the second part of the $\bot$-modification. For every $j \in \{1, \ldots, k-1\}$ and for every $H \in \mathcal{H} \setminus \{\bot\}$, choose as $\hat{W}^j$ the intersection $U_H^{\sqsubset,X_j} \cap W_\bot^j$. Hence,

$$U_H'^{t,X_j} = ((U_H^{t,X_j} \setminus (U_H^{t,X_j} \cap U_{\neg H}^{t,X_j} \cap W_\bot^j)) \cup (U_H^{\sqsubset,X_j} \cup W_\bot^j)$$

(Case 2): This case is analogous to the guessing strategy for modifications in $\mathcal{FL}_\bot$. If $\sigma_t(E_k)$ and $\sigma_\sqsubset(E_k)$ agree on the $\bot$-languages, we again have

$$U_\bot^{t,X_k} \cdot N_R^* = \sigma_t(E_k)|_\bot \cdot N_R^* = \sigma_\sqsubset(E_k)|_\bot \cdot N_R^*.$$

As $\sigma_t(E_k) \sqsubset \sigma_\sqsubset(E_k)$, this implies that there is an $A \in \mathcal{C}$ and a word $\hat{u} \in U_A^{t,X_k}$ such that $\hat{u} \notin \sigma_\sqsubset(E_k)|_A$. For the modification, choose $\hat{A} := A$ and use one word $\hat{u}$ as introduced above.

In Case 1, we again find that

$$\sigma_t'(E_k)|_\bot \cdot N_R^* = \sigma_t(E_k)|_\bot \cdot N_R^* \setminus \bigcup_{j=1}^{k-1} V_{X_j}^{X_k} \cdot W_\bot^j.$$

Following the same argument as employed for $\mathcal{FL}_\bot$, we furthermore obtain

$$U_\bot'^{t,X_k} \cdot N_R^* \supset \sigma_t'(E_k)|_\bot \cdot N_R^*,$$

which is a necessary condition for the strict subsumption $\sigma_t(E_k) \sqsubset \sigma'(E_k)$. The second second condition for subsumption remains to be shown, i.e.:

$$U_H'^{t,X_k} \cup U_\bot'^{t,X_k} \cdot \Sigma^* \supseteq \sigma'(E_k)|_H \cup \sigma_t'(E_k)|_\bot \cdot \Sigma^*$$

for all $H \in \mathcal{H} \setminus \{\bot\}$. Note that the substitution σ is assumed reduced, which makes it possible to use the role language $U_\bot'^{t,X_k}$ instead of $\hat{U}_\bot'^{t,X_k}$, as seen in

the characterization of the subsumption. We can show the above inclusion by exploiting the fact that $\sigma_t \sqsubset \sigma_{\sqsubset}$. Since after removing inconsistencies from the relevant role languages only the intersection $U_H^{\sqsubset,X_j} \cup W_\perp^j$ was added to the role languages $U_H^{\prime t,X_j}$, it is not difficult to see that $\sigma \sqsubset \sigma'$ holds as well.

In Case 2, the proof is analogous to that for $\mathcal{FL}_\perp$, because the $\mathcal{C}$-modification for $\mathcal{FL}_\neg$ also only removes words from role languages U_H^{t,X_j}, where $H \in \mathcal{H} \setminus \{\perp\}$. ∎

Minimality of the modifications can be shown similar to the proof for $\mathcal{FL}_\perp$. The possibility of inconsistencies introduced by role words referring to negated atomic concepts thereby is hidden by the notation defined in Definition 58. With this prerequisite, the proof works analogous. Furthermore, Lemma 56 only depends on the facts i) that in the modification of $\sigma_t(E_k)$ the variables in $\{X_k, \ldots, X_\ell\}$ remain unchanged, ii) that $\text{MATCH}^{\sqsubseteq}_{\mathcal{FL}_\perp}$ computes least matchers w.r.t. the ordering $\sqsubseteq$ on substitutions and iii) that modifications are successful for a solvable matching problem. These facts also hold for $\mathcal{FL}_\neg$, as we have already seen. Consequently, the proof of soundness of the algorithm $\text{MATCH}^{\sqsubseteq}_{\mathcal{FL}_\neg}$ is identical to Lemma 56.

Part (i) in the proof of completeness for $\mathcal{FL}_\perp$ (Lemma 57) again only relies on the minimality of the algorithm $\text{MATCH}^{\sqsubseteq}_{\mathcal{FL}_\perp}$, so that the same argument can be used for $\mathcal{FL}_\neg$. It can also be shown that (ii) is valid for modifications in $\mathcal{FL}_\neg$, i.e. we always have $\sigma'_t \sqsubseteq \sigma_{\sqsubset}$. As seen in the second part of Lemma 57, these findings—in addition to the minimality of matching under subsumption conditions—are sufficient to show completeness.

It is easy to see that the length of each computation branch of the nondeterminstic algorithm $\text{MATCH}^{\sqsubseteq}_{\mathcal{L}}$ is polynomially bounded. Because matching under strict subsumption conditions in $\mathcal{FL}_\perp$ and $\mathcal{FL}_\neg$ is known to be NP-hard, we obtain the following theorem.

Theorem 60 *Let $\mathcal{L} \in \{\mathcal{FL}_\perp, \mathcal{FL}_\neg\}$. Deciding the solvability of $\mathcal{L}$-matching problems under general side conditions is an NP-complete problem.*

References

[1] F. Baader and R. Küsters. Matching in description logics with existential restrictions. In A.G. Cohn, F. Giunchiglia, and B. Selman, editors,

Proceedings of the Seventh International Conference on Knowledge Representation and Reasoning (KR2000), pages 261–272, Breckenridge, CO, 2000. Morgan Kaufmann Publishers.

[2] F. Baader, R. Küsters, A. Borgida, and D. McGuinness. Matching in Description Logics. *Journal of Logic and Computation*, 9(3):411–447, 1999.

[3] F. Baader and P. Narendran. Unification of concept terms in description logics. In H. Prade, editor, *Proceedings of the 13th European Conference on Artificial Intelligence (ECAI-98)*, pages 331–335, Brighton, UK, 1998. John Wiley & Sons Ltd.

[4] Franz Baader and Tobias Nipkow. *Term Rewriting and All That.* Cambridge University Press, New York, 1998.

[5] A. Borgida and R. Küsters. What's not in a name: Some Properties of a Purely Structural Approach to Integrating Large DL Knowledge Bases. In F. Baader and U. Sattler, editors, *Proceedings of the 2000 International Workshop on Description Logics (DL2000)*, number 33 in CEUR-WS, Aachen, Germany, 2000. RWTH Aachen. Proceedings online available from http://SunSITE.Informatik.RWTH-Aachen.DE/Publications/CEUR-WS/Vol-33/.

[6] A. Borgida and D. L. McGuinness. Asking Queries about Frames. In L.C. Aiello, J. Doyle, and S.C. Shapiro, editors, *Proceedings of the Fifth International Conference on Principles of Knowledge Representation and Reasoning (KR'96)*, pages 340–349, Cambridge, Massachusetts, USA, 1996. Morgan Kaufmann Publishers.

[7] A. Borgida and P. Patel-Schneider. A Semantics and Complete Algorithm for Subsumption in the CLASSIC Description Logic. *Journal of Artificial Intelligence Research*, 1:277–308, 1994.

[8] R. J. Brachman, D. L. McGuinness, P. F. Patel-Schneider, L. A. Resnick, and A. Borgida. Living with CLASSIC: When and how to use a KL-ONE-like language. In J. Sowa, editor, *Principles of Semantic Networks*, pages 401–456. Morgan Kaufmann Publishers, San Mateo, Calif., 1991.

[9] Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, and Andrea Schaerf. Reasoning in description logics. In Gerhard Brewka, editor,

Principles of Knowledge Representation, Studies in Logic, Language and Information, pages 193–238. CSLI Publications, 1996.

[10] J.E. Hopcroft and J.D. Ullman. *Introduction to Automata Theory.* Addison-Wesley Publ. Co., 1979.

[11] I. Horrocks. Using an expressive description logic: FaCT or fiction? In A.G. Cohn, L. Schubert, and S.C. Shapiro, editors, *Proceedings of the Sixth International Conference on Principles of Knowledge Representation and Reasoning (KR'98)*, pages 636–647, Trento, Italy, 1998. Morgan Kaufmann Publishers.

[12] R. Küsters. Characterizing the Semantics of Terminological Cycles in $\mathcal{ALN}$ using Finite Automata. In *Proceedings of the Sixth International Conference on Principles of Knowledge Representation and Reasoning (KR'98)*, pages 499–510, Trento, Italy, 1998. Morgan Kaufmann Publishers.

[13] D.L. McGuinness. *Explaining Reasoning in Description Logics.* PhD thesis, Department of Computer Science, Rutgers University, October, 1996. Also available as a Rutgers Technical Report LCSR-TR-277.