LTCS–Report

Aachen University of Technology
Research group for
Theoretical Computer Science

Structural Subsumption Considered from an Automata-Theoretic Point of View

F. Baader, R. Küsters, and R. Molitor

LTCS-Report 98-04

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

Ahornstr. 55
52074 Aachen
Germany

Structural Subsumption Considered from an Automata-Theoretic Point of View

F. Baader, R. Küsters, and R. Molitor
LuFg Theoretische Informatik, RWTH Aachen, Germany.
E-mail: {baader, kuesters, molitor}@informatik.rwth-aachen.de

Abstract

This paper compares two approaches for deriving subsumption algorithms for the description logic $\mathcal{ALN}$: structural subsumption and an automata-theoretic characterization of subsumption. It turns out that structural subsumption algorithms can be seen as special implementations of the automata-theoretic characterization.

1 Introduction

Description logics (DLs) and corresponding DL systems can be used to represent the terminological knowledge of a problem domain in a structured and well-defined way. Relevant concepts of the domain are described by concept descriptions, which are formed from atomic concepts (unary predicates) and roles (binary predicates) using concept forming operators provided by the DL. One of the most important inference services of a DL system is to arrange the represented concepts of the domain in a superconcept/subconcept hierarchy. This reasoning task is based on the subsumption relation between concept descriptions. Intuitively, a concept D subsumes a concept C if the set of individuals represented by D is a superset of the one represented by C.

In the literature several approaches to subsumption have been investigated (see [8] for an overview). In this work, we are interested in the relation between two of these approaches for languages of small expressive power, namely, structural subsumption and an automata theoretic approach.

Structural subsumption algorithms are efficient methods for deciding subsumption in description logics without full negation, disjunction, and existential restrictions. The structural subsumption algorithm employed by the system CLASSIC [5, 6] is based on a specific data structure for representing concept descriptions, called *description graphs*. In this context, subsumption is reduced to a structural comparison of description graphs.

Another approach for deciding subsumption in sub-languages of CLASSIC can be obtained from the automata-theoretic characterizations of subsumption w.r.t. greatest fixed-point (gfp) semantics in cyclic terminologies [1, 12], which reduce the subsumption problem to an inclusion problem for certain *regular languages*. In the case of acyclic terminologies (and thus in particular for concept descriptions), these languages turn out to be finite.

At first sight, there is no connection between these two approaches since they are based on rather different normal forms for concept descriptions. Intuitively speaking, structural subsumption is based on a normal form that applies the equivalence $\forall R.(A \sqcap B) \equiv \forall R.A \sqcap \forall R.B$ as a rewrite rule from right to left, i.e., the descriptions are grouped w.r.t. role names, whereas the finite languages considered in the automata-theoretic approach correspond to a normal form obtained by applying the above equivalence from left to right.

Another difference between the two approaches is that they describe decision procedures for subsumption on two different levels of abstraction. The structural subsumption algorithm for CLASSIC is presented in [5, 6] on the level of the data structure (namely, description graphs) used in the implementation. This provides a description of the algorithm that is very close to its actual implementation. Consequently, both the formal description of the algorithm and the proof of its correctness are quite complex [5, 6, 13]. In contrast, the automata-theoretic approach reduces the subsumption problem to a formal language problem (namely, inclusion of finite or regular languages), which means that the description of the subsumption algorithm (and thus also the proof of its correctness) can be split into two independent parts: (i) the characterization of subsumption on the abstract formal language level, and (ii) an algorithm that decides the formal language problem.

The goal of this report is to show that there is in fact a tight relation between the approaches. In order to illustrate this relation, we will first construct an isomorphism between the data-structures both approaches are working on. Then we point out, that structural subsumption algorithms based on description graphs can be seen as "parallel" implementations of the language inclusion tests required by the automata-theoretic characterizations of subsumption.

The report is structured as follows. We first introduce the description logics of interest, namely, the small language $\mathcal{FL}_0$, which allows for value restrictions and conjunction, and the more expressive language $\mathcal{ALN}$, which additionally provides us with atomic negation and number restrictions. In Section 3 we first describe the automata theoretic approach to subsumption in $\mathcal{FL}_0$ as well as structural subsumption for $\mathcal{FL}_0$-concept descriptions. Thereafter, we discuss the tight relation between both approaches in detail. An extension of the comparison to $\mathcal{ALN}$ is given in Section 6. Both approaches to subsumption of concept descriptions have to be extended in order to cope with inconsistencies that are expressible in $\mathcal{ALN}$. We will show, that there is a 1-1-correspondence between the extensions done in the automata theoretic approach and the way inconsistency is handled in structural subsumption. In the last section we summarize our analysis and give an overview over future work concerned with different approaches to subsumption in description logics.

2 Preliminary

We first introduce syntax and semantics of the description logics $\mathcal{FL}_0$ and $\mathcal{ALN}$ as well as the inference problem subsumption.

Definition 1 (Syntax and semantics) Let $\mathcal{C}$ be a set of *concept names* and Σ a set of *role names*. $\mathcal{ALN}$*-concept descriptions* are inductively defined as follows:

- P and $\neg P$ are concept descriptions for each concept name $P \in \mathcal{C}$.
- Let C, D be concept descriptions, $R \in \Sigma$ a role name and $n \in \mathbb{N}$. Then
  - $C \sqcap D$ (conjunction),
  - $\forall R.C$ (value restriction),
  - $(\leq n\ R)$ and $(\geq n\ R)$ (number restrictions)

  are concept descriptions as well.

An interpretation $\mathcal{I} = (dom(I), \cdot^I)$ exists of the *domain* $dom(I)$, i.e., a set of individuals, and an *interpretation function* $\cdot^I$ that maps each concept name $P \in \mathcal{C}$ to a subset P^I of $dom(I)$ and each role name $R \in \Sigma$ to a binary relation $R^I \subseteq dom(I) \times dom(I)$. The extension of $\cdot^I$ to arbitrary concept descriptions is inductively defined as shown in Table 1. The concept descriptions $\top$ and $\bot$ denote the entire domain and the empty set, respectively.

Syntax	Semantics
$\top$	$dom(I)$
$\bot$	$\emptyset$
$\neg P$	$dom(I) \setminus P^I$
$C \sqcap D$	$C^I \cap D^I$
$\forall R.C$	$\{x \in dom(I) \mid \forall y : (x,y) \in R \longrightarrow y \in C^I\}$
$(\leq n\ R)$	$\{x \in dom(I) \mid \lvert\{y \mid (x,y) \in R^I\}\rvert \leq n\}$
$(\geq n\ R)$	$\{x \in dom(I) \mid \lvert\{y \mid (x,y) \in R^I\}\rvert \geq n\}$

Table 1: Semantics of concept descriptions

The description logic $\mathcal{FL}_0$ only allows for the constructors conjunction $(C \sqcap D)$ and value restriction $(\forall R.C)$, whereas $\mathcal{ALN}$ additionally allows for primitive negation and number restrictions. Notice that both constructors $\top$ and $\bot$ are expressible in $\mathcal{ALN}$ because of $\top \equiv (\geq 0\ R)$ and $\bot \equiv (P \sqcap \neg P)$. W.l.o.g. we will use them only as abbreviations rather than allowing for these constructors in $\mathcal{ALN}$-concepts explicitly.

For some induction we will need the *role depth* of a concept description C:

$$
\begin{array}{lcl}
\mathsf{depth}(P) & := & \mathsf{depth}(\neg P) \;:=\; 0, \\
\mathsf{depth}(\forall R.C) & := & \mathsf{depth}(C) + 1, \\
\mathsf{depth}((\leq n\; R)) & := & \mathsf{depth}((\geq n\; R)) \;:=\; 1, \\
\mathsf{depth}(C_1 \sqcap C_2) & := & \mathsf{max}\{\mathsf{depth}(C_1), \mathsf{depth}(C_2)\}.
\end{array}
$$

Definition 2 (Subsumption) Let C, D be $\mathcal{ALN}$-concept descriptions. D *subsumes* C (for short $C \sqsubseteq D$) iff $C^I \subseteq D^I$ for all interpretations $\mathcal{I}$.
C is *equivalent* to D (for short $C \equiv D$) iff $C \sqsubseteq D$ and $D \sqsubseteq C$, i.e., $C^I = D^I$ for all interpretations $\mathcal{I}$.

Definition 3 (Terminology) Let A be a concept name and C an $\mathcal{ALN}$-concept description. Then $A = C$ is a *concept definition.* A finite set T of concept definitions is an $\mathcal{ALN}$*-terminology* if each concept name in T occurs at most once as left-hand side of a concept definition. We call concept names appearing on the left-hand side of some concept definition in T *defined concepts* and *primitive* otherwise. The set of all defined concepts in T is denoted by $\mathcal{D}_T$. T is called $\mathcal{FL}_0$*-terminology* iff each right hand side of a concept definition in T is an $\mathcal{FL}_0$-concept description.

An interpretation $\mathcal{I} = (dom(I), \cdot^I)$ is called *model of* T iff $A^I = D^I$ for each concept definition in T. The *descriptive semantics* of a terminology T is defined by the set of all models of T.

Notice that Definition 3 allows for *cyclic* terminologies. Formally, cycles are defined as follows: Let A be a defined concept and B an atomic concept in T. The concept A *directly uses* B if B occurs on the right-hand side of the concept definition $A = C$ in T. Let *uses* be the transitive closure of 'directly uses'. Then T is *cyclic* iff there exists a defined concept name A in T that uses itself. Otherwise T is called *acyclic.*

Definition 4 (Primitive interpretation and its extension) Let T be a terminology, $P_1, \ldots, P_m$ the primitive concepts, $R_1, \ldots, R_k$ the role names, and $A_1, \ldots, A_n$ the defined concepts in T. A *primitive interpretation* $\mathcal{J} = (dom(J), \cdot^J)$ consists of the domain $dom(J)$ as well as the interpretation of the primitive concepts $({P_1}^J, \ldots, P_m^J)$ and the interpretation of the role names $(R_1^J, \ldots, R_k^J)$.
An interpretation $\mathcal{I} = (\Delta, \cdot^I)$ of T is an *extension* of J iff $P_1^I = P_1^J, \ldots, P_m^I = P_m^J$ and ${R_1}^I = {R_1}^J, \ldots, {R_k}^I = {R_k}^J$.

In [2] it has been pointed out that for cyclic terminologies one can not uniquely extend an primitive interpretation $\mathcal{J}$ to a model $\mathcal{I}$ of T in general. Therefore, beside the descriptive semantics also fixed-point semantics are considered in case of cyclic terminologies. For acyclic terminologies, descriptive and fixed-point semantics coincide since there is always a unique extension $\mathcal{I}$ of a primitive interpretation $\mathcal{J}$ to a model of T. Therefore, in case of acyclic terminologies we do not distinguish between the primitive interpretation $\mathcal{J}$ and its extension. In particular, A^J denotes the extension of A defined by the model of T which is uniquely determined by $\mathcal{J}$ and T. Furthermore, one can define subsumption of atomic concepts which are defined in different terminologies over the same set of primitive concepts and roles.

Definition 5 (Subsumption w.r.t. terminologies) Let T_A and T_B be acyclic $\mathcal{ALN}$-terminologies where A is a defined concept in T_A and B is defined in T_B. Furthermore, we assume that T_A and T_B are defined over the same set of primitive concepts and roles. We say that A in T_A is *subsumed* by B in T_B ($A \sqsubseteq_{T_A,T_B} B$ for short) iff for all primitive interpretations $\mathcal{J}$ it holds $A^J \subseteq B^J$. A is *equivalent* to B (for short $A \equiv_{T_A,T_B} B$) iff $A^J = B^J$ for all primitive interpretations $\mathcal{J}$.

In the sequel, C, D denote concept descriptions, A, B refer to defined concepts, P, Q are used for primitive concepts, and R, S for roles.

3 The automata theoretic approach to subsumption for $\mathcal{FL}_0$

The automata theoretic approach has been proposed in order to gain a more profound understanding of cyclic terminologies. The idea of this approach is to assign a semi-automaton $\mathcal{A}_T$ to a terminology T, and to characterize different semantics as well as important inference problems, e.g. subsumption, using this semi-automaton. More precisely, Baader [2] has given an automata theoretic characterization of the semantics and subsumption of cyclic $\mathcal{FL}_0$-terminologies. These results are extended in [12] to the language $\mathcal{ALN}$.

3.1 Concept descriptions and automata

As mentioned in the introduction, we are interested in comparing the automata theoretic approach to subsumption of concept descriptions and structural subsumption for concept descriptions. Therefore we represent concept descriptions by defined concepts in (acyclic) $\mathcal{FL}_0$-*terminologies*. The semi-automata corresponding to these terminologies are specified recursively in order to simplify the comparison presented in Section 5.

Definition 6 (Terminology of C) Let $C = P_1 \sqcap \ldots \sqcap P_n \sqcap \forall R_1.C_1 \sqcap \ldots \sqcap \forall R_m.C_m$ be an $\mathcal{FL}_0$-concept description. We represent C by the defined concept A in the *terminology* T_C *of* C. We refer to A as the *defined concept* of C in T_C. The terminology T_C is recursively defined by

- If $C = P_1 \sqcap \ldots \sqcap P_n$, then $T_C := \{A = P_1 \sqcap \ldots \sqcap P_n\}$.
- If $C = P_1 \sqcap \ldots \sqcap P_n \sqcap \forall R_1.C_1 \sqcap \ldots \sqcap \forall R_m.C_m$, then let T_{C_i} be the recursively defined terminologies of C_i, $1 \leq i \leq m$, and A_i the defined concept of C_i in T_{C_i}, respectively. W.l.o.g. the sets $\mathcal{D}_{T_{C_i}}$ are pairwise disjoint. $T_C := \{A = P_1 \sqcap \ldots \sqcap P_n \sqcap \forall R_1.A_1 \sqcap \ldots \sqcap \forall R_m.A_m\} \cup \bigcup\limits_{1 \leq i \leq m} T_{C_i}$

Note that the set of primitive concepts and roles in T_C coincides with the concept names occuring in C. Thus a primitive interpretation $\mathcal{J}$ of T_C is also an interpretation for C. Furthermore, it is easy to see that T_C is acyclic. By induction on the role depth of C it is not hard to show that $A^J = C^J$ holds for all primitive interpretations $\mathcal{J}$.

After introducing the terminologies representing concept descriptions, we now assign a finite semi-automaton $\mathcal{A}_T = (\Sigma, Q, \Delta)$ to an $\mathcal{FL}_0$-terminology T [2]. The alphabet Σ denotes the set of role names in T and the concept names in T yield the set of states Q of $\mathcal{A}_T$. The transitions $\Delta \subseteq Q \times (\Sigma \cup \{\varepsilon\}) \times Q$ of $\mathcal{A}_T$ are induced by the value restrictions in T, i.e., every concept definition $A = P_1 \sqcap \ldots \sqcap P_n \sqcap \forall R_1.A_1 \sqcap \ldots \sqcap \forall R_m.A_m$ in T gives rise to the transitions (A, ε, P_i), $1 \leq i \leq n$ and (A, R_i, A_i), $1 \leq i \leq m$. Notice that for acyclic terminologies the corresponding semi-automata are acyclic as well. In the general case of a cyclic terminology T the corresponding automaton $\mathcal{A}_T$ is cyclic (see [2] for an example).

The semi-automaton $\mathcal{A}_{T_C}$ corresponding to the concept description C can be constructed recursively.

Definition 7 (Semi-Automaton of C) Let C be an $\mathcal{FL}_0$-concept description. The *semi-automaton* $\mathcal{A}_{T_C}$ *of* C is recursively defined by

$\mathsf{depth}(C) = 0$**:** $C = P_1 \sqcap \ldots \sqcap P_n$:
Let $T_C = \{A = P_1 \sqcap \ldots \sqcap P_n\}$ be the terminology of C.
Then $\mathcal{A}_{T_C} := (\emptyset, \{A, P_1, \ldots, P_n\}, \Delta)$ where $\Delta = \{(A, \varepsilon, P_i) \mid 1 \leq i \leq n\}$.

$\mathsf{depth}(C) > 0$**:** $C = P_1 \sqcap \ldots \sqcap P_n \sqcap \forall R_1.C_1 \sqcap \ldots \sqcap \forall R_m.C_m$:
Let T_C be the terminology of C with $A = P_1 \sqcap \ldots \sqcap P_n \sqcap \forall R_1.A_1 \sqcap \ldots \sqcap \forall R_m.A_m \in T_C$ and A_i the defined concept name of C_i, $1 \leq i \leq m$. W.l.o.g. the sets $\mathcal{D}_{T_{C_i}}$, $1 \leq i \leq m$, and $\mathcal{D}_{T_C}$ are pairwise disjoint. Further, let $\mathcal{A}_{T_{C_i}} = (\Sigma_i, Q_i, \Delta_i)$ be the recursively defined semi-automaton of C_i, $1 \leq i \leq m$.
Then $\mathcal{A}_{T_C} := (\Sigma, Q, \Delta)$ where

- $\Sigma := \{R_1, \ldots, R_m\} \cup \bigcup_{1 \leq i \leq m} \Sigma_i$,
- $Q := \{A, P_1 \ldots, P_n\} \cup \bigcup_{1 \leq i \leq m} Q_i$ and
- $\Delta := \{(A, \varepsilon, P_i) \mid 1 \leq i \leq n\} \cup \{(A, R_i, A_i) \mid 1 \leq i \leq m\} \cup \bigcup_{1 \leq i \leq m} \Delta_i$.

In order to define the power set automaton in Section 3 and to compare automata and description graphs in Section 5, we need the following definitions. Let $\mathcal{A} = (\Sigma, Q, \Delta)$ be a semi-automaton, $p, q \in Q$, and $I \subseteq Q$. There exists a *path* from p to q with label $W \in \Sigma^*$ in $\mathcal{A}$ iff there are states $p_0, \ldots, p_n \in Q$ and $R_1, \ldots, R_n \in \Sigma \cup \{\varepsilon\}$ such that $p_0 = p$, $p_n = q$, $(p_{i-1} R_i p_i) \in \Delta$ for $1 \leq i \leq n$, and $R_1 \ldots R_n = W$. The *ε-closure* of a set $I \subseteq Q$ of states is defined by

$$\varepsilon\text{-}\mathsf{closure}(I) \;:=\; \{q' \in Q \mid \text{there exists } q \in I \text{ and a (possibly empty) path from } q \text{ to } q' \text{ with label } \varepsilon \text{ in } \mathcal{A}\}.$$

The *W-successor set* of $I \subseteq Q$ in $\mathcal{A}$ w.r.t. $W \in \Sigma^*$ is defined by

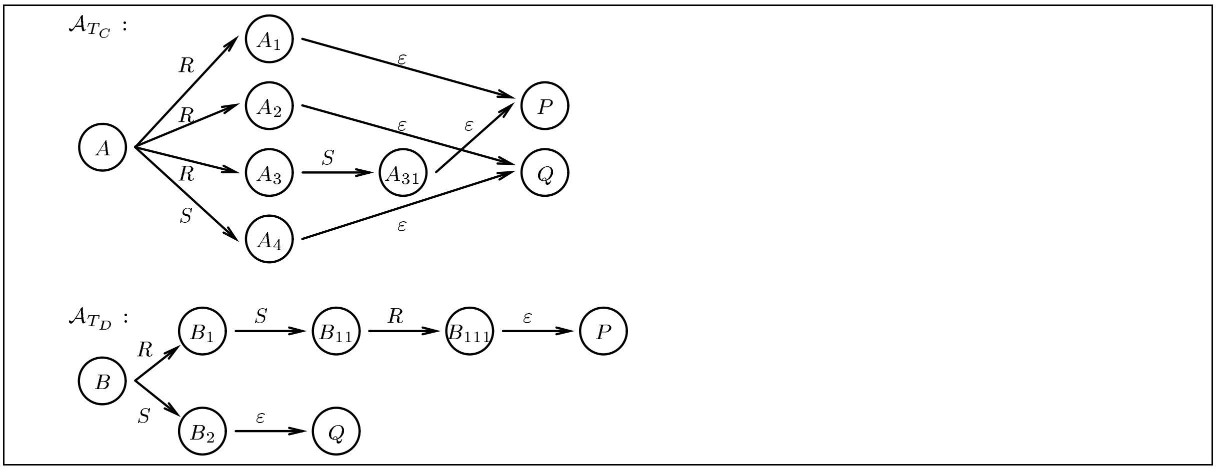

Figure 1: The semi-automata corresponding to T_C and T_D.

$$next_{\mathcal{A}}(I, W) := \{q' \in Q \mid \text{there exists } q \in \varepsilon\text{-}\mathsf{closure}(I) \text{ and a path from } q \text{ to } q' \text{ with label } W \text{ in } \mathcal{A}\}.$$

Example 8 Consider the concept descriptions $C := \forall R.P \sqcap \forall R.Q \sqcap \forall R.\forall S.P \sqcap \forall S.Q$ and $D := \forall R.\forall S.\forall R.P \sqcap \forall S.Q$. These descriptions can be represented by the defined concepts A and B in the following acyclic $\mathcal{FL}_0$-terminologies:

$$\begin{array}{llll} T_C\colon & A = \forall R.A_1 \sqcap \forall R.A_2 \sqcap \forall R.A_3 \sqcap \forall S.A_4, & T_D\colon & B = \forall R.B_1 \sqcap \forall S.B_2, \\ & A_1 = P, & & B_1 = \forall S.B_{11}, \\ & A_2 = Q, & & B_{11} = \forall R.B_{111}, \\ & A_3 = \forall S.A_{31}, & & B_{111} = P, \\ & A_{31} = P, & & B_2 = Q. \\ & A_4 = Q. & & \end{array}$$

The terminologies T_C and T_D yield the semi-automata $\mathcal{A}_{T_C}$ and $\mathcal{A}_{T_D}$ of Figure 1.

For a defined concept A and a primitive concept P in T, the language $L_{\mathcal{A}_T}(A, P)$ is the set of all words labeling paths in $\mathcal{A}_T$ from A to P. Since for acyclic terminologies T the corresponding semi-automata $\mathcal{A}_T$ are also acyclic, the languages $L_{\mathcal{A}_T}(A, P)$ are finite. In the case of cyclic terminologies, these languages can be infinite.

Example 9 (Example 8 continued) Consider the semi-automata $\mathcal{A}_{T_C}$ and $\mathcal{A}_{T_D}$ in Figure 1 of the concept descriptions C and D from Example 8. In this example, we have $L_{\mathcal{A}_{T_C}}(A, P) = \{R, RS\}$, $L_{\mathcal{A}_{T_C}}(A, Q) = \{R, S\}$, $L_{\mathcal{A}_{T_D}}(B, P) = \{RSR\}$, and $L_{\mathcal{A}_{T_D}}(B, Q) = \{S\}$.

3.2 The automata theoretic characterization of subsumption

In [2], an automata theoretic characterization of subsumption has been proved for cyclic $\mathcal{FL}_0$-terminologies w.r.t. to both descriptive semantics and fixed-point semantics. We restrict our attention to acyclic terminologies. Thus, the subsumption relation coincides for these semantics.

Let T denote an ayclic $\mathcal{FL}_0$-terminology, A a defined concept in T, P a primitive concept, and W a finite word over Σ. It can be shown [2] that $A \sqsubseteq_T \forall W.P$ iff $W \in L_{\mathcal{A}_T}(A, P)$. Thus, roughly speaking, the language $L_{\mathcal{A}_T}(A, P)$ represents exactly those value restrictions that are satisfied by A. Consequently, an atomic concept A is subsumed by the atomic concept B iff the set of value restrictions which have to be satisfied by A is a superset of the value restrictions which have to be satisfied by B. Formally, this fact is stated in

Theorem 10 (Characterizing subsumption for $\mathcal{FL}_0$) *Let T_A and T_B be acyclic $\mathcal{FL}_0$-terminologies defined over the same set of primitive concepts and roles. Furthermore, let A be defined in T_A and B in T_B. Then*

$$A \sqsubseteq_{T_A,T_B} B \quad \textit{iff} \quad \textit{for all primitive concepts } P \textit{ it holds that } L_{\mathcal{A}_{T_B}}(B, P) \subseteq L_{\mathcal{A}_{T_A}}(A, P).$$

Proof. See [2].

In the example we have $L_{\mathcal{A}_{T_D}}(B, P) = \{RSR\} \not\subseteq L_{\mathcal{A}_{T_C}}(A, P)$. By Theorem 10 this implies $A \not\sqsubseteq_{T_C,T_D} B$.

In order to decide subsumption based on Theorem 10, one must decide the inclusion problem for regular languages. Note, however, that in case of acyclic terminologies the considered languages are merely finite.

In the next section we recall the automata theoretic approach of deciding inclusion of regular languages given by finite automata (see [10] for details).

3.3 Deciding inclusion of regular languages

A finite automaton $\mathcal{A}$ is a semi-automaton with initial and finite states, i.e., $\mathcal{A} = (\Sigma, Q, I, \Delta, F)$ where Σ denotes the finite alphabet, Q the finite set of states, $I \subseteq Q$ the set of initial states, $\Delta \subseteq Q \times (\Sigma \cup \{\varepsilon\}) \times Q$ the transition set, and $F \subseteq Q$ the set of final states. Note that w.l.o.g. we can assume I to be a singleton.

In automata theory, the inclusion problem for regular languages L_1 and L_2, defined by $\mathcal{A}_1$ and $\mathcal{A}_2$, respectively, is reduced to the emptiness problem: $L_2 \subseteq L_1$ iff $L_2 \cap \overline{L}_1 = \emptyset$. In order to decide this problem one first computes a deterministic automaton $\mathcal{B}_1$ for L_1, i.e., one constructs the powerset automaton of $\mathcal{A}_1$. This automaton is defined as follows:

Definition 11 (Powerset automaton) Let $\mathcal{A} = (\Sigma, Q, q_0, \Delta, F)$ be a finite automaton. Then the *powerset automaton* $\mathcal{P}(\mathcal{A})$ *of* $\mathcal{A}$ is defined by $\mathcal{P}(\mathcal{A}) := (\Sigma, \widehat{Q}, \widehat{q}_0, \widehat{\delta}, \widehat{F})$ where

- $\widehat{Q} := \{G \subseteq Q \mid next_{\mathcal{A}}(\varepsilon\text{-}\mathsf{closure}(q_0), W) = G \text{ for an } W \in \Sigma^*\}$,
- $\widehat{q}_0 := \varepsilon\text{-}\mathsf{closure}(q_0)$,
- $\widehat{\delta}(I, R) := next_{\mathcal{A}}(I, R) \in \widehat{Q}$ for $I \in \widehat{Q}$, and $R \in \Sigma$,

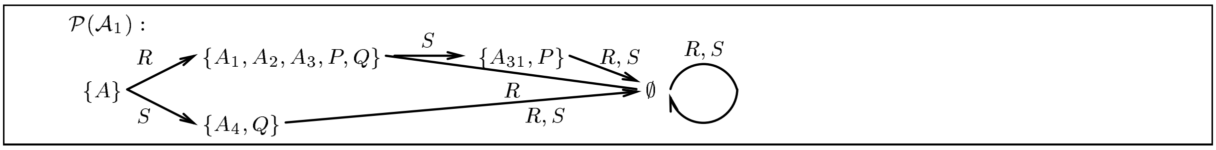

Figure 2: The powerset automaton of $\mathcal{A}_{T_C}$

- $\widehat{F} := \{G \in \widehat{Q} \mid G \cap F \neq \emptyset\}$.

From the powerset automaton $\mathcal{B}_1$ of $\mathcal{A}_1$ we obtain a finite automaton $\overline{\mathcal{B}}_1$ for $\overline{L}_1$ by permuting the set of final and non-final states in $\mathcal{B}_1$. Note that this permutation only leads to an automaton accepting the complement of L_1 since $\mathcal{B}_1$ is deterministic. Now, one can construct the product automaton of $\overline{\mathcal{B}}_1$ and $\mathcal{A}_2$ (without ε-transitions), which accepts the language $L_2 \cap \overline{L}_1$. Emptiness of this language can be tested by deciding if the language accepted by the product automaton is empty. An algorithm deciding this problem can be described as follows: Search for a word W that is accepted by $\mathcal{A}_2$ and $\overline{\mathcal{B}}_1$. If such a word exists, then the language accepted by the product automaton is not empty. Using the 'Pumping-Lemma' [10] it can be shown that it is sufficient to consider words W up to a certain length, namely, the product of the size of $\mathcal{A}_2$ and $\overline{\mathcal{B}}_1$, which corresponds to the size of the product automaton, i.e., the number of states of the product automaton.

In the general case of arbitrary finite automata, the powerset automaton $\mathcal{B}_1$ is of exponential size in the worst case, and inclusion of regular languages is PSPACE-complete [9]. In case of acyclic finite automata, the powerset automaton still may be exponential in the size of $\mathcal{A}_1$, whereby inclusion is coNP-complete (see [14]). However, for terminologies constructed from concept descriptions it can be shown that this exponential blow-up cannot occur (see Section 3.3.1).

Example 12 (Example 8 continued) Consider the language $L_{\mathcal{A}_{T_C}}(A, P)$ (see Example 8). $L_{\mathcal{A}_{T_C}}(A, P)$ is accepted by the semi-automaton $\mathcal{A}_{T_C}$ if A is the initial state and P the final state. The corresponding powerset automaton is shown in Figure 2 where $\{A\}$ is the initial state and the states containing P are final states. This automaton accepts the complement $\overline{L_{\mathcal{A}_{T_C}}(A, P)}$ of the language $L_{\mathcal{A}_{T_C}}(A, P)$ if we specify $\{A\}$ as initial state and all states *not* containing P as final states, i.e., if we permute final and non-final states. Hence, we have $L_{\mathcal{A}_{T_D}}(B, P) \cap \overline{L_{\mathcal{A}_{T_C}}(A, P)} \neq \emptyset$ iff there exists a word W such that (1) there is a path with label W in $\mathcal{P}(\mathcal{A}_{T_C})$ leading from $\{A\}$ to a state not containing P, and (2) there is a path with label W in $\mathcal{A}_{T_D}$ leading from B to P. In the example, RSR is such a word. Consequently, the inclusion $L_{\mathcal{A}_{T_D}}(B, P) \subseteq L_{\mathcal{A}_{T_C}}(A, P)$ does not hold.

3.3.1 The complexity of the powerset construction

In this section we are concerned with the complexity of the powerset construction for finite automata. As already mentioned, even for an acyclic finite automaton $\mathcal{A}$ the corresponding powerset automaton can be exponential in the size of $\mathcal{A}$. The automata obtained from concept descriptions are acyclic but have a certain structure, namely a *weak tree* *structure*. We will show, that for this class of automata the powerset construction yields automata of linear size.

Intuitively speaking, a finite automaton $\mathcal{A} = (\Sigma, Q, q_0, \Delta, F)$ has a *tree structure* if

- there are no ε-transitions in Δ,
- q_0 has no predecessor, i.e., there is no transition of the form $(q, R, q_0) \in \Delta$, and
- each state $q \in Q \setminus \{q_0\}$ is reachable from q_0 and has exactly one predecessor, i.e., for each $q \in Q \setminus \{q_0\}$ there exists a unique state $q' \in Q$ and a unique symbol $R \in \Sigma$ such that $(q', R, q) \in \Delta$.

Thus, the graphical notation of $\mathcal{A}$ yields a tree with root q_0. Furthermore, for each $q \in Q$ there is exactly one path from q_0 to q in $\mathcal{A}$ and each q has a unique level, namely the length of the label of this path from q_0 to q in $\mathcal{A}$.

The automata corresponding to concept descriptions do not have a tree structure, because they contain ε-transitions (see Figure 1). But in these automata, we have only ε-transitions of a special kind, i.e., each state that is reached via ε-transitions has no outgoing transitions. In other words, these states, namely the primitive concepts occuring in the concept descriptions, can be seen as special leaves in these automata.

Definition 13 (weak tree structure) A finite automaton $\mathcal{A} = (\Sigma, Q, q_0, \Delta, F)$ has a *weak tree structure* iff

- q_0 has no predecessor, i.e., there is no transition of the form $(q, R, q_0) \in \Delta$, $R \in \Sigma \cup \{\varepsilon\}$,
- for each $q \in Q \setminus \{q_0\}$, there exists at least one path from q_0 to q in $\mathcal{A}$,
- $Q = Q_\sigma + Q_\varepsilon$ such that
  - each $q \in Q_\varepsilon$ is only reached via ε-transitions and has no outgoing transitions, i.e., for all $q \in Q_\varepsilon$: $(q', R, q) \in \Delta \Longrightarrow q' \in Q_\sigma$ and $R = \varepsilon$, and there exists no transition of the form $(q, R, q') \in \Delta$, $R \in \Sigma \cup \{\varepsilon\}$,
  - for each $q \in Q_\sigma \setminus \{q_0\}$ there exists a unique state $q' \in Q$ and a unique symbol $R \in \Sigma$ such that $(q', R, q) \in \Delta$.

We define the *level* of each state $q \in Q_\sigma$, $\mathit{level}(q) \in \mathbb{N}$, as the length of the unique path from q_0 to q in $\mathcal{A}$.

Notice that if $\mathcal{A}$ has a weak tree structure then $\mathcal{A}$ is acyclic. Consequently, $\mathcal{A}$ is not complete, i.e., it exists $q \in Q, R \in \Sigma$ such that there exists no transition of the form $(q, R, q') \in \Delta$. So one must introduce a sink state $\emptyset$ within the powerset construction. But due to the weak tree structure of $\mathcal{A}$, each state in the powerset automaton has a special form.

To make this more precise, let $\mathcal{P}(\mathcal{A}) = (\Sigma, \widehat{Q}, \widehat{q}_0, \widehat{\delta}, \widehat{F})$ be the powerset automaton obtained from $\mathcal{A}$ by Definition 11. We refer to

- the set of all states $q \in Q_\sigma$ with $level(q) = l$ by
  $Q_l := \{q \in Q_\sigma \mid level(q) = l\}$, and
- the set of all states in $\mathcal{P}(\mathcal{A})$ that are reached by words of length l by
  $\widehat{Q}_l := \{next_{\mathcal{A}}(\{q_0\}, W) \mid W \in \Sigma^l\} \setminus \{\emptyset\}$.

Intuitively speaking, each state in the powerset automaton beside the sink has a certain level. Further, the states on one level yield a partition of the set Q_l w.r.t. Q_σ. Therefore, the number of states on level l in $\mathcal{P}(\mathcal{A})$ can be bounded by $|Q_l|$ and we get

$$|\widehat{Q}| = 1 + \sum_{l \leq 0} |\widehat{Q}_l| \leq 1 + \sum_{l \leq 0} |Q_l| = 1 + |Q_\sigma| \leq 1 + |Q|.$$[1]

Consequently, we obtain the desired result, i.e., the size of the powerset automaton of an automaton $\mathcal{A}$ with weak tree structure is linear in the size of $\mathcal{A}$.

Formally, we show by induction on the level $l \in \mathbb{N}$ that for $\widehat{Q}_l$ it is

1. $I_1 \cap I_2 \setminus Q_\varepsilon = \emptyset$ for $I_1, I_2 \in \widehat{Q}_l$, $I_1 \neq I_2$ and
2. $\bigcup_{I \in \widehat{Q}_l} I \setminus Q_\varepsilon = Q_l$.

$l = 0:$ It is $\widehat{Q}_0 = \{next_{\mathcal{A}}(\{q_0\}, \varepsilon)\} = \{\varepsilon\text{-}\mathsf{closure}(q_0)\}$. So, the first condition is satisfied trivially. The second condition is satisfied because it is $\varepsilon\text{-}\mathsf{closure}(q_0) \setminus Q_\varepsilon = \{q_0\} = Q_0$.

$l \longrightarrow l + 1:$ It is $\widehat{Q}_{l+1} = \{next_{\mathcal{A}}(I, R) \mid I \in \widehat{Q}_l, R \in \Sigma\} \setminus \{\emptyset\}$.

Let $q \in Q_{l+1}$. There exists a word W of length $l+1$ such that $W \in L_{\mathcal{A}}(q_0, q)$. Thus, $q \in next_{\mathcal{A}}(\{q_0\}, W)$. By definition of $\widehat{Q}_{l+1}$ we get $q \in \bigcup_{I \in \widehat{Q}_{l+1}} I \setminus Q_\varepsilon$ and therefore $Q_{l+1} \subseteq \bigcup_{I \in \widehat{Q}_{l+1}} I \setminus Q_\varepsilon$.

Conversely, let $I \in \widehat{Q}_{l+1}$. There exists $I' \in \widehat{Q}_l$, $R \in \Sigma$ such that $I = next_{\mathcal{A}}(I', R)$. Let $q \in I$. For $q \in Q_\varepsilon$ nothing has to be shown. Assume $q \in Q_\sigma$. We want to show $q \in Q_{l+1}$. Since $\mathcal{A}$ has a weak tree structure q has a unique R-predecessor $q' \in I'$. By induction we get $q' \in Q_l$ and hence $q \in Q_{l+1}$. Thus we have $\bigcup_{I \in \widehat{Q}_{l+1}} I \setminus Q_\varepsilon \subseteq Q_{l+1}$.

So the second condition is satisfied. In order to prove claim (1) for $l + 1$ we show that for $I_1, I_2 \in \widehat{Q}_{l+1}$ $I_1 \cap I_2 \setminus Q_\varepsilon \neq \emptyset$ implies $I_1 = I_2$. So let $q \in I_1 \cap I_2 \setminus Q_\varepsilon$. There exist $I'_1, I'_2 \in \widehat{Q}_l$ and $R_1, R_2 \in \Sigma$ such that $I_i = next_{\mathcal{A}}(I'_i, R_i)$ for $i = 1, 2$. As already shown it is $q \in Q_{l+1}$. This implies that there exists a unique $q' \in Q_l$ and a unique $R \in \Sigma$ with $(q', R, q) \in \Delta$. It follows $R = R_1 = R_2$ and $q' \in I'_1$, $q' \in I'_2$. Hence $q' \in I'_1 \cap I'_2 \setminus Q_\varepsilon$. By induction we get $I'_1 = I'_2$. Since $R_1 = R_2$ this implies $I_1 = I_2$.

We sum up the complexity result for the powerset construction for finite automata with weak tree structure in

[1]Notice that the upper bound $1 + |Q|$ is reached, if $\mathcal{A}$ is deterministic and has a weak tree structure without ε-transitions. In this case, the powerset construction would add the sink state and transitions leading to the sink in order to obtain a deterministic and complete automaton.

Lemma 14 *Let $\mathcal{A} = (\Sigma, Q, q_0, \Delta, F)$ be a finite automaton with weak tree structure and $\mathcal{P}(\mathcal{A}) = (\Sigma, \widehat{Q}, \widehat{q}_0, \widehat{\delta}, \widehat{F})$ the powerset automaton of $\mathcal{A}$. Then the size of $\mathcal{P}(\mathcal{A})$ is linear in the size of $\mathcal{A}$.*

As an easy consequence of Definition 6 and Definition 7 we get that the automata corresponding to $\mathcal{FL}_0$-concept descriptions have a weak tree structure[2]. By Lemma 14 this implies that the size of the corresponding powerset automata is linear in the size of the concept descriptions. So the language inclusion tests required by the automata theoretic characterization of subsumption can be decided in time polynomial in the size of the concept descriptions.

4 Structural subsumption algorithms based on description graphs

In this section we present a characterization of subsumption of $\mathcal{FL}_0$-concept descriptions based on structural subsumption in CLASSIC [5, 6]. The idea behind is as follows: given two $\mathcal{FL}_0$-concept descriptions C and D, we translate the concept descriptions into equivalent *description graphs* $\mathcal{G}_C$ and $\mathcal{G}_D$. A normalization of $\mathcal{G}_C$ yields the *canonical* description graph $\widehat{\mathcal{G}}_C$ of C. Thereafter, we can decide $C \sqsubseteq D$ by some kind of structural comparison of $\widehat{\mathcal{G}}_C$ and $\mathcal{G}_D$.

4.1 Description Graphs

Description graphs were introduced in [5, 6] for deciding subsumption of concept descriptions in CLASSIC. Since $\mathcal{FL}_0$ is a sublanguage of CLASSIC, we first confine the notion of description graphs given in [5]. Description graphs are rooted directed acyclic graphs whose nodes are labeled by sets of primitive concepts and whose edges are labeled by roles. Concept descriptions can be turned into description graphs by a straightforward translation of the syntactic structure of the descriptions. It will turn out, that the description graphs corresponding to $\mathcal{FL}_0$-concept descriptions are trees.

Definition 15 ($\mathcal{FL}_0$-description graphs) Let $\mathcal{C}$ be a set of primitive atomic concepts and Σ a set of role names. An *$\mathcal{FL}_0$-description graph over $\mathcal{C}$ and Σ* is a tuple $\mathcal{G} = (V, E, v_0, l)$ where $V = \{v_0, \ldots, v_n\}$ is a set of nodes, $E \subseteq V \times \Sigma \times V$ a set of edges and $v_0 \in V$ the root of $\mathcal{G}$ such that

- there exists no edge vRv_0 in E,
- for each $v \in V \setminus \{v_0\}$ there exists exactly one $v' \in V$ and exactly one $R \in \Sigma$ with $v'Rv \in E$,

[2]The automata corresponding to $\mathcal{ALN}$-concept descriptions also have a weak tree strucuture, so the result can be extended to $\mathcal{ALN}$.

- each $v \in V$ is reachable from v_0, i.e., there is a path $v_0 R_1 v_1 \dots v_{n-1} R_n v$ in E, and
- the label $l(v)$ of a node $v \in V$ is a finite subset of $\mathcal{C}$.

In the sequel, we will use the following notions referring to paths and subgraphs.
$p = w_0 R_1 w_1 \dots w_{n-1} R_n w_n$ is called *path* from w_0 to w_n with label $W = R_1 \dots R_n$ in $\mathcal{G}$ iff $w_{i-1} R_i w_i \in E$ for all $1 \leq i \leq n$. The path p is called *rooted path* if $w_0 = v_0$, i.e., p starts at the root of $\mathcal{G}$. $\mathcal{G} \mid_v$ denotes the subgraph of $\mathcal{G}$ with root $v \in V$, i.e., $\mathcal{G} \mid_v = (V', E', v, l')$ with $V' := \{w \in V \mid \text{ exists path from } v \text{ to } w \text{ in } \mathcal{G}\}$, $E' := E \cap V' \times \Sigma \times V'$, and $l'(w) := l(w)$ for $w \in V'$.
The size of a description graph $\mathcal{G} = (V, E, v_0, l)$ is defined as the sum of the number of nodes and edges and the sum of the size of all labels, i.e.,

$$|\mathcal{G}| := |V| + |E| + \sum_{v \in V} |l(v)|.$$

After introducing the syntax of description graphs and thus the data structure our structural subsumption test is working on, we now have to define the semantics: which set of individuals $\mathcal{G}^I$ is determined by a description graph $\mathcal{G}$ under an interpretation $\mathcal{I}$.

Definition 16 (Extension of Description Graphs) Let $\mathcal{I}$ be an interpretation of $\mathcal{C}$ and Σ, $\mathcal{G} = (V, E, v_0, l)$ a description graph over $\mathcal{C}$ and Σ.
The *extension* of a node $v \in V$ is recursively defined by $x \in v^I$ iff

- $x \in P^I$ for all $P \in l(v)$ and
- for all $vRv' \in E$ and $y \in dom(I)$ with $(x, y) \in R^I$ it holds that $y \in v'^I$.

The *extension* of $\mathcal{G}$ is defined as $\mathcal{G}^I := v_0^I$.

We have introduced syntax and semantics of description graphs. Our aim is to use this representation formalism to characterize subsumption of $\mathcal{FL}_0$-concept descriptions. Therefore, we first have to translate $\mathcal{FL}_0$-concept descriptions into (equivalent) description graphs.

4.2 Translating concept descriptions into description graphs

The translation of $\mathcal{FL}_0$-concept descriptions into description graphs is formalized by the algorithm in Figure 3. Obviously, the size of $\mathcal{G}_C$ is linear in the size of C. In the sequel, $\mathcal{G}_C$ denotes the description graph of C where C is an $\mathcal{FL}_0$-concept description and $\mathcal{G}_C$ is obtained from C by the algorithm in Figure 3.

The translation is sound in the following sense:

Lemma 17 (Equivalence of concepts and description graphs) *Let C be an arbitrary $\mathcal{FL}_0$-concept description and $\mathcal{G}_C$ the description graph of C. Then for all interpretations $\mathcal{I}$ it holds that $C^I = \mathcal{G}_C^I$.*

Input: An $\mathcal{FL}_0$-concept $C = \big(P_1 \sqcap \ldots \sqcap P_n \sqcap \forall R_1.C_1 \sqcap \ldots \sqcap \forall R_m.C_m\big)$
Output: The corresponding description graph $\mathcal{G}_C = (V, E, v_0, l)$

$m = 0$: $\mathcal{G}_C := (\{v_0\}, \emptyset, v_0, l)$ where
$l(v_0) := \{P_1, \ldots, P_n\}$.

$m > 0$: Let $\mathcal{G}_{C_i} = (V_i, E_i, v_{0i}, l_i)$ be the recursively defined description graph of C_i, $1 \leq i \leq m$, where w.l.o.g. the V_i are pairwise disjoint and $v_0 \notin \bigcup_{1\leq i \leq m} V_i$. $\mathcal{G}_C := (V, E, v_0, l)$ is defined by

- $V := \{v_0\} \cup \bigcup_{1\leq i\leq m} V_i$,
- $E := \{v_0 R_i v_{0i} \mid 1 \leq i \leq m\} \cup \bigcup_{1\leq i\leq m} E_i$,
- $l(v) := \begin{cases} \{P_1, \ldots, P_n\} & \text{, if } v = v_0 \\ l_i(v) & \text{, if } v \in V_i \end{cases}$

Figure 3: Translating concept descriptions into description graphs

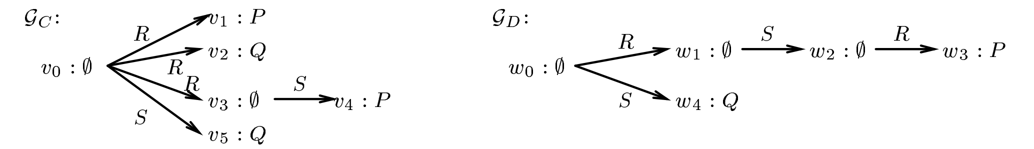

Figure 4: The description graphs corresponding to C and D.

Proof. By induction on the number of all-quantifiers $|C|_\forall$ in C (see [13] for details).

Example 18 (Example 8 continued) Figure 4 shows the description graphs $\mathcal{G}_C$ and $\mathcal{G}_D$ corresponding to the concept descriptions C, D given in Example 8. The label $\emptyset$ at the root of $\mathcal{G}_C$ expresses that no primitive concept occurs in the top-level conjunction of C. The edge labeled S from the root to a node labeled Q says that there is a value restriction $\forall S.C'$ in the top-level conjunction of C such that Q is the only primitive concept occurring in the top-level conjunction of C', etc.

4.3 Structural subsumption for $\mathcal{FL}_0$

Before we can decide whether C is subsumed by D based on a structural comparison of the description graphs, the graph for the subsumee C must be normalized by merging successor nodes reached by edges labeled by the same role name. This corresponds to applying the rewrite rule $\forall R.A \sqcap \forall R.B \to \forall R.(A \sqcap B)$ to the descriptions. Formally, we apply the normalization rule shown in Figure 5 as long as possible to an $\mathcal{FL}_0$-description graph $\mathcal{G}_C$. Notice that each iterated application of the rule in Figure 5 terminates since $|\mathcal{G}| > |\mathcal{G}'|$ if $\mathcal{G}'$ is obtained from $\mathcal{G}$ by one application of the rule. Furthermore, it is not

Let $\mathcal{G} = (V, E, v_0, l)$ be an $\mathcal{FL}_0$-description graph. $\mathcal{G}' = (V', E', v_0, l')$ is obtained from $\mathcal{G}$ by:
Let $v \in V$ with $n > 1$ R-successors $v_1, \ldots, v_n$ and v' a new node not occuring in V. Then $\mathcal{G}'$ is defined by merging $v_1, \ldots, v_n$ to one R-successor v' of v:

- $V' := V \setminus \{v_1, \ldots, v_n\} \cup \{v'\}$,
- $E' := E[v_i/v' \mid i = 1 \ldots n]$ (each v_i is replaced by v'),
- $l'(v') := \bigcup_{i=1 \ldots n} l(v_i)$ and $l'(w) := l(w)$, $w \in V' \setminus \{v'\}$.

Figure 5: The normalization rule for $\mathcal{FL}_0$-description graphs

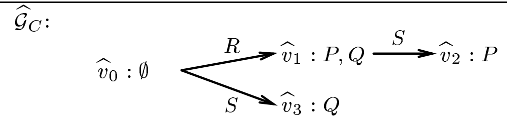

Figure 6: The canonical description graph corresponding to C.

hard to see that the normalization rule is sound, i.e., if $\mathcal{G}'$ is obtained from $\mathcal{G}$, then it is $\mathcal{G}^I = \mathcal{G}'^I$ for all interpretations $\mathcal{I}$.

Definition 19 (Canonical $\mathcal{FL}_0$-description graphs) Let C be an $\mathcal{FL}_0$-concept description and $\mathcal{G}_C$ the description graph of C. The *canonical description graph of* $\mathcal{G}_C$ is defined as the description graph $\widehat{\mathcal{G}}_C$ that is obtained from $\mathcal{G}_C$ by applying the normalization rule in Figure 5 to $\mathcal{G}_C$ as long as possible.

Notice that the canonical description graph $\widehat{\mathcal{G}}_C = (\widehat{V}, \widehat{E}, \widehat{v}_0, \widehat{l})$ of $\mathcal{G}_C$ is a deterministic tree, i.e., each node $v \in \widehat{V}$ has at most one R-successor in $\widehat{\mathcal{G}}_C$ for each $R \in \Sigma$. Since $|\mathcal{G}_C|$ is linear in the size of C and $|\widehat{\mathcal{G}}_C| \leq |\mathcal{G}_C|$ the size of the canonical description graph corresponding to an $\mathcal{FL}_0$-concept description C is linear in the size of C. As an example consider the canonical description graph $\widehat{\mathcal{G}}_C$ of $\mathcal{G}_C$ from Example 18 depicted in Figure 6.

In order to formalize the structural comparison of description graphs we need the notion of more specific paths [6, 13].

Definition 20 (More specific paths) Let $\mathcal{G} = (E, V, v_0, l)$ and $\mathcal{G}' = (V', E', v_0', l')$ be $\mathcal{FL}_0$-description graphs.
A node $v \in V$ is *more specific* than a node $v' \in V'$ iff $l'(v') \subseteq l(v)$. A rooted path $p = v_0 R_1 v_1 \ldots v_{n-1} R_n v_n$ in $\mathcal{G}$ is *more specific* than the rooted path $p' = v_0' R_1' v_1' \ldots v_{m-1}' R_m' v_m'$ in $\mathcal{G}'$ iff

1. $m \leq n$,
2. $R_i = R_i'$ for $1 \leq i \leq m$, and

3. for all $0 \leq i \leq m$ it is v_i more specific than v'_i.

Now we are equipped to characterize subsumption of $\mathcal{FL}_0$-concept descriptions by a structural comparison of description graphs. C is subsumed by D iff the conditions to W-successors of instances of D are subsets of the conditions to W-successors of instances of C for each $W \in \Sigma^*$. Intuitively speaking, these conditions to W-successors are represented by the labels of W-successor nodes in the corresponding description graphs $\widehat{\mathcal{G}}_C$ and $\mathcal{G}_D$, respectively. If the label of a W-successor node of the root contains the primitive concept P, then for each instance x of C all W-successors of x must be in the extension of P. Therefore, we can decide $C \sqsubseteq D$ by testing wether the label of each W-successor node in $\mathcal{G}_D$ is a subset of the label of the W-successor node in $\widehat{\mathcal{G}}_C$[3] for each $W \in \Sigma^*$. Formally, we can prove

Theorem 21 (Structural subsumption for $\mathcal{FL}_0$) *Let C, D be $\mathcal{FL}_0$-concept descriptions, $\widehat{\mathcal{G}}_C$ the canonical description graph of C and $\mathcal{G}_D$ the description graph of D. Then $C \sqsubseteq D$ iff for each rooted path p in $\mathcal{G}_D$ there exists a more specific rooted path $\widehat{p}$ in $\widehat{\mathcal{G}}_C$.*

Proof. See [13].

Example 22 (Example 18 continued) Consider the $\mathcal{FL}_0$-concept descriptions C and D from Example 18, $\mathcal{G}_D$ in Figure 4 and $\widehat{\mathcal{G}}_C$ in Figure 6. The path with label RS in $\widehat{\mathcal{G}}_C$ is more specific than the path with label RS in $\mathcal{G}_D$. However, for the path with label RSR in $\mathcal{G}_D$ there does not exist a more specific path in $\widehat{\mathcal{G}}_C$. Consequently, the structural subsumption test recognizes that C is not subsumed by D.

An algorithm deciding $C \sqsubseteq D$ by Theorem 21 considers the (canonical) description graphs $\widehat{\mathcal{G}}_C$ and $\mathcal{G}_D$ and tests wether there exists a more specific rooted path $\widehat{p}$ in $\widehat{\mathcal{G}}_C$ for each rooted path p in $\mathcal{G}_D$. In other words, for each W-successor node v of v_0 in $\mathcal{G}_D = (V, E, v_0, l)$ we test (1) wether there exists a W-successor node $\widehat{v}$ of $\widehat{v}_0$ in $\widehat{\mathcal{G}}_C = (\widehat{V}, \widehat{E}, \widehat{v}_0, \widehat{l})$ or not and (2) if $\widehat{v}$ is the W-successor of $\widehat{v}_0$ in $\widehat{\mathcal{G}}_C$, wether $l(v) \subseteq \widehat{l}(\widehat{v})$. If (1) or (2) does not hold, then $C \not\sqsubseteq D$; otherwise $C \sqsubseteq D$. A more formal algorithm can be found in [13].

5 Comparing the approaches for $\mathcal{FL}_0$

We first illustrate the tight relation between both approaches by Example 8. If we compare Figure 1 with Figure 4, then we see that the description graphs $\mathcal{G}_C$ and $\mathcal{G}_D$ essentially agree with the semi-automata $\mathcal{A}_{T_C}$ and $\mathcal{A}_{T_D}$. The only difference is that in the semi-automata there is only one state for every atomic concept and the primitive concepts P and Q are only reached from defined concepts, e.g. A_1, by ε-transitions. In general, there exists an "isomorphism" between the semi-automaton $\mathcal{A}_{T_C}$ and the description graph $\mathcal{G}_C$ corresponding to C. Roughly speaking, we can define a bijective mapping φ from the set

[3]Notice that since $\widehat{\mathcal{G}}_C$ is canonical, there is at most one W-successor node of the root in $\widehat{\mathcal{G}}_C$.

of nodes in $\mathcal{G}_C$ to the set of defined concepts in $\mathcal{A}_{T_C}$ such that the label of a node v is the same as the set of all primitive concepts reached from $\varphi(v)$ by ε-transitions.

Another obvious similarity between the automata-theoretic and the structural approach is that in both cases the automaton/graph for the subsumee C must be modified. A closer look at Figure 2 and Figure 6 reveals, that the powerset automaton $\mathcal{P}(\mathcal{A}_{T_C})$ and the canonical description graph $\widehat{\mathcal{G}}_C$ are also essentially identical. To be more precise, $\widehat{\mathcal{G}}_C$ can be obtained from $\mathcal{P}(\mathcal{A}_{T_C})$ by (1) removing the names of defined concepts from the states, and (2) by removing the sink state $\emptyset$ and the edges leading to this sink.

First, we consider the relationship between $\mathcal{G}_C$ and $\mathcal{A}_{T_C}$.

Lemma 23 *Let C be an $\mathcal{FL}_0$-concept description, $\mathcal{G}_C = (V, E, v_0, l)$ the description graph of C, $\mathcal{A}_{T_C} = (\Sigma, Q, \Delta)$ the semi-automaton of C and $\mathcal{D}_{T_C}$ the set of defined atomic concepts in $\mathcal{A}_{T_C}$.*
Then there exists a bijective mapping $\varphi : V \longrightarrow \mathcal{D}_{T_C}$ such that

1. *$l(v) = \varepsilon\text{-}\mathsf{closure}(\varphi(v)) \setminus \{\varphi(v)\}$ for all $v \in V$ and*

2. *$(\varphi(v), R, \varphi(w)) \in \Delta$ iff $vRw \in E$ for all $R \in \Sigma$.*

Proof. By induction on the role depth of C.

$\mathsf{depth}(C) = 0$: Then we have $C = P_1 \sqcap \ldots \sqcap P_n$. Furthermore, it is $\mathcal{G}_C = (\{v_0\}, \emptyset, v_0, l)$ with $l(v_0) = \{P_1, \ldots, P_n\}$ and $\mathcal{A}_{T_C} = (\emptyset, \{A, P_1, \ldots, P_n\}, \Delta)$ with $\Delta = \{(A, \varepsilon, P_i) \mid 1 \leq i \leq n\}$ and $\mathcal{D}_{T_C} = \{A\}$. We define $\varphi : \{v_0\} \longrightarrow \{A\}$ with $\varphi(v_0) := A$. By construction it is $\varepsilon\text{-}\mathsf{closure}(A) = \{A, P_1, \ldots, P_n\}$ and hence $l(v_0) = \varepsilon\text{-}\mathsf{closure}(\varphi(v_0)) \setminus \{\varphi(v_0)\}$. So, φ satisfies the first condition of Lemma 23. The second condition is satisfied trivially because there is no R-successors of v_0 in $\mathcal{G}_C$ or of A in $\mathcal{A}_{T_C}$.
$\mathsf{depth}(C) > 0$: Then we have $C = P_1 \sqcap \ldots \sqcap P_n \sqcap \forall R_1.C_1 \sqcap \ldots \sqcap \forall R_m.C_m$. By the algorithm in Figure 3 we have

- the description graph of C, $\mathcal{G}_C = (V, E, v_0, l)$,
- the recursively defined description graphs of C_i, $\mathcal{G}_{C_i} = (V_i, E_i, v_{0i}, l_i)$, $1 \leq i \leq m$.

For $\mathcal{G}_C$ and $\mathcal{G}_{C_i}$ it holds that

- V_i, V_j are pairwise disjoint,
- $V = \{v_0\} \cup V_1 \cup \ldots \cup V_m$, and
- $E = \{v_0 R_i v_{0i} \mid 1 \leq i \leq m\} \cup E_1 \cup \ldots \cup E_m$.

By Definition 6 and Definition 7 we have

- the recursively defined terminologies of C and C_i, $1 \leq i \leq m$, T_C and T_{C_i} with $\mathcal{D}_{T_C} = \{A\} \cup \mathcal{D}_{T_{C_1}} \cup \ldots \cup \mathcal{D}_{T_{C_m}}$,

- the defined concept names A and A_i of C and C_i, $1 \leq i \leq m$, and
- the recursively defined semi-automata of C and C_i, $\mathcal{A}_{T_C} = (\Sigma, Q, \Delta)$ and $\mathcal{A}_{T_{C_i}} = (\Sigma_i, Q_i, \Delta_i)$.

It is $\mathsf{depth}(C_i) < \mathsf{depth}(C)$ for $1 \leq i \leq m$. By induction there exist bijective mappings $\varphi_i : V_i \longrightarrow \mathcal{D}_{T_{C_i}}$ such that φ_i satisfies 1 and 2 of Lemma 23 for $\mathcal{G}_{C_i}$ and $\mathcal{A}_{T_{C_i}}$.
We define $\varphi : V \longrightarrow \mathcal{D}_{T_C}$ by

$$\varphi(v_0) := A \text{ and } \varphi(v) := \varphi_i(v) \text{ for } v \in V_i.$$

Since by construction the V_i as well as the $\mathcal{D}_{T_{C_i}}$ are pairwise disjoint and each φ_i is bijective, φ is a bijective mapping from $V = \{v_0\} \cup V_1 \cup \ldots \cup V_m$ to $\mathcal{D}_{T_C} = \{A\} \cup \mathcal{D}_{T_{C_1}} \cup \ldots \cup \mathcal{D}_{T_{C_m}}$. Because of $E = \{v_0 R_i v_{0i} \mid 1 \leq i \leq m\} \cup E_1 \cup \ldots \cup E_m$ and $\Delta := \{(A, \varepsilon, P_i) \mid 1 \leq i \leq n\} \cup \{(A, R_i, A_i) \mid 1 \leq i \leq m\} \cup \Delta_1 \cup \ldots \cup \Delta_m$ and the induction hypothesis it is $(\varphi(v), R, \varphi(w)) \in \Delta$ iff $vRw \in E$ for all $R \in \Sigma$. As an easy consequence of the construction we have $l(v_0) = \varepsilon\text{-}\mathsf{closure}(A) \setminus \{A\}$ and by the induction hypothesis it is $l(v) = \varepsilon\text{-}\mathsf{closure}(\varphi(v)) \setminus \{\varphi(v)\}$ for all $v \in V_1 \cup \ldots \cup V_m$. So, φ satisfies 1 and 2 of Lemma 23. □

Example 24 (Example 8 and Example 18 continued) Consider the description graph $\mathcal{G}_C = (V, E, v_0, l)$ in Figure 4 and the semi-automaton $\mathcal{A}_{T_C}$ in Figure 1 of the concept description C from Example 8. The mapping $\varphi : V \longrightarrow \mathcal{D}_{T_C}$ is given by

$v \in V$	v_0	v_1	v_2	v_3	v_4	v_5
$\varphi(v) \in \mathcal{D}_{T_C}$	A	A_1	A_2	A_3	A_{31}	A_4

In the next step, we formalize the relation between the canonical description graph $\widehat{\mathcal{G}}_C$ of C and the power set automaton $\mathcal{P}(\mathcal{A}_{T_C})$ of C. As already mentioned, there is no node in the canonical description graph of a concept description C corresponding to the sink state $\emptyset$ in the powerset automaton of C. The automaton directly corresponding to the canonical description graph of C can be obtained from the powerset automaton by eliminating the sink and all edges leading to the sink. In the sequel, this *deterministic automaton of* C is denoted by $\mathcal{B}_{T_C}$. Analogous to Definition 11, $\mathcal{B}_{T_C}$ can be defined by $\mathcal{B}_{T_C} := (\Sigma, Q', \delta')$ with

- $Q' := \{F \subseteq Q \mid F \neq \emptyset, next_{\mathcal{A}_{T_C}}(\varepsilon\text{-}\mathsf{closure}(A), W) = F \text{ for a word } W \in \Sigma^*\}$ and
- for $I \in Q'$, $R \in \Sigma$

$$\delta'(I, R) := \begin{cases} next_{\mathcal{A}_{T_C}}(I, R) \in Q' \,, next_{\mathcal{A}_{T_C}}(I, R) \neq \emptyset \\ \text{undefined} \,, \text{otherwise.} \end{cases}$$

The relation between the canonical description graph of C and the deterministic automaton becomes obvious, if we compare the normalization rule in Figure 5 and the definition of the transition function δ' of $\mathcal{B}_{T_C}$.

Let $\mathcal{G}_C = (V, E, v_0, l)$ be the description graph of C, $\mathcal{A}_{T_C} = (\Sigma, Q, \Delta)$ the semi-automaton of C, and A the defined concept of C. Let further $\{v_1, \ldots, v_n\} \subseteq V$ be the non-empty set of all R-successors of v_0 in $\mathcal{G}_C$. The application of the normalization rule to v_0 and R in $\mathcal{G}_C$ yields a description graph $\mathcal{G}' = (V', E', v_0, l')$ where $v_1, \ldots, v_n$ are merged to one new R-successor v_{new} of v_0.

On the other hand, defining the R-successor $\delta'(\varepsilon\text{-}\mathsf{closure}(A), R)$ of $\varepsilon\text{-}\mathsf{closure}(A)$ in $\mathcal{B}_{T_C}$ means merging all states that can be reached from a state in $\varepsilon\text{-}\mathsf{closure}(A)$ by a path labeled with R to one new state in $\mathcal{B}_{T_C}$. By Lemma 23 we know that the set $\{\varphi(v_1), \ldots, \varphi(v_n)\}$ is the set of all R-successors of A in $\mathcal{A}_{T_C}$. By definition of $\mathcal{A}_{T_C}$, primitive concepts are only reached from defined concepts by ε-transitions and there are no edges leaving from primitive concepts in $\mathcal{A}_{T_C}$. It follows $next_{\mathcal{A}_{T_C}}(\varepsilon\text{-}\mathsf{closure}(A), R) = next_{\mathcal{A}_{T_C}}(\{A\}, R) = \bigcup_{1 \le i \le n} \varepsilon\text{-}\mathsf{closure}(\varphi(v_i))$. This shows that merging all R-successors of v_0 in $\mathcal{G}_C$ to one new R-successor is the same as defining $\delta'(\varepsilon\text{-}\mathsf{closure}(A), R)$. As a consequence of Lemma 23, property 1 we get

$$
\begin{aligned}
l'(v_{new}) &= \bigcup_{1 \le i \le n} l(v_i) \\
&= \bigcup_{1 \le i \le n} \varepsilon\text{-}\mathsf{closure}(\varphi(v_i)) \setminus \{\varphi(v_i)\} \\
&= \bigcup_{1 \le i \le n} \varepsilon\text{-}\mathsf{closure}(\varphi(v_i)) \setminus \{\varphi(v_1), \ldots, \varphi(v_n)\} \\
&= \bigcup_{1 \le i \le n} \varepsilon\text{-}\mathsf{closure}(\varphi(v_i)) \setminus \mathcal{D}_{T_C} \\
&= \delta'(\varepsilon\text{-}\mathsf{closure}(A), R) \setminus \mathcal{D}_{T_C}.
\end{aligned}
$$

More general, let $\mathcal{G}'_C$ be the description graph obtained from $\mathcal{G}_C$ by applying the normalization rule to each non-empty set of R-successors of v_0, $R \in \Sigma$, and let v be an R-successor of v_0 in $\mathcal{G}'_C$. Generating $\mathcal{G}'_C$ is the same as defining the transition function δ' for $\varepsilon\text{-}\mathsf{closure}(A)$ and each $R \in \Sigma$ such that there is at least one R-successor of A in $\mathcal{A}_{T_C}$. Furthermore, applying the normalization rule recursively to the subgraph $\mathcal{G}'_C \mid_v$ is the same as defining δ' for each RW-successor set of $\varepsilon\text{-}\mathsf{closure}(A)$, $W \in \Sigma^*$. Thus, generating the canonical description graph $\widehat{\mathcal{G}}_C$ of C recursively is the same as defining the deterministic automaton $\mathcal{B}_{T_C}$ of C recursively.

As a consequence of the above observations we can recursively define a bijective mapping $\widehat{\varphi}$ from the set of nodes $\widehat{V}$ of the canonical description graph $\widehat{\mathcal{G}}_C = (\widehat{V}, \widehat{E}, v_0, \widehat{l})$ of C to the set of states Q' of the deterministic automaton $\mathcal{B}_{T_C}$ of C, such that

- $\widehat{\varphi}(v_0) = \varepsilon\text{-}\mathsf{closure}(A)$,
- $\widehat{\varphi}(v) = \delta'(\varepsilon\text{-}\mathsf{closure}(A), W)$ if v is a W-successor of v_0 in $\widehat{\mathcal{G}}_C$,
- $\widehat{l}(v) = \widehat{\varphi}(v) \setminus \mathcal{D}_{T_C}$ for all $v \in \widehat{V}$, and
- $vRw \in \widehat{E}$ iff $\delta'(\widehat{\varphi}(v), R) = \widehat{\varphi}(w)$ for all $R \in \Sigma$.

So far, we have pointed out the tight relation between both approaches on the level of the data structures they are working on. In the sequel, we will argue that due to the similarity between the automata and description graphs structural subsumption algorithms can be seen as a special implementation of the language inclusion tests required by the automata theoretic characterization of subsumption of concept descriptions.

To make this more precise, let C, D be concept descriptions, $\widehat{\mathcal{G}}_C$ and $\mathcal{G}_D$ the corresponding (canonical) description graphs, $\mathcal{A}_{T_D}$ the automaton of D with defined concept B of D, and $\mathcal{P}(\mathcal{A}_{T_C})$ the powerset automaton of C with defined concept A of C. Assume that $C \not\sqsubseteq D$. By Theorem 21, the structural subsumption algorithm detects non-subsumption by finding a rooted path p in $\mathcal{G}_D$ with label W such that there does not exist a more specific rooted path $\widehat{p}$ in $\widehat{\mathcal{G}}_C$. There are two possible reasons why this more specific path does not exist in $\widehat{\mathcal{G}}_C$:

(1) There is no path with label W in $\widehat{\mathcal{G}}_C$. Without loss of generality we may assume that the path p in $\mathcal{G}_D$ ends in a node with non-empty label set. Otherwise, the path could be extended appropriately because each leaf node in a description graph corresponding to an $\mathcal{FL}_0$-concept description has a non-empty label set. Obviously, such an extended path has still no more specific path in $\widehat{\mathcal{G}}_C$. Now assume that the primitive concept P is contained in the label set of the last node in p. Then we have $W \in L_{\mathcal{A}_{T_D}}(B, P) \cap \overline{L_{\mathcal{A}_{T_C}}(A, P)}$. In fact, $W \in L_{\mathcal{A}_{T_D}}(B, P)$ because the path p in $\mathcal{G}_D$ to a node containing P yields a path in $\mathcal{A}_{T_D}$ from B to P. The fact that there is no (rooted) path with label W in $\widehat{\mathcal{G}}_C$ implies that the path with label W in $\mathcal{P}(\mathcal{A}_{T_C})$ leads from the initial state to the sink state $\emptyset$. Since $\emptyset$ does not contain P, it is a final state for the automaton accepting $\overline{L_{\mathcal{A}_{T_C}}(A, P)}$.

(2) For p and the (unique) path $\widehat{p}$ with label W in $\widehat{\mathcal{G}}_C$, the inclusion condition between the labels is violated by some primitive concept P, i.e., P belongs to the label of a node in p, but not to the label of the corresponding node in $\widehat{p}$. Again, we may assume that p ends in the node v for which the inclusion condition is violated. (Obviously, the prefix of p that ends in v has also no more specific path in $\widehat{\mathcal{G}}_C$.) An argument similar to the one employed in the first case can be used to show that $W \in L_{\mathcal{A}_{T_D}}(B, P) \cap \overline{L_{\mathcal{A}_{T_C}}(A, P)}$.

To sum up, we have shown that the existence of a rooted path in $\mathcal{G}_D$ without a more specific path in $\widehat{\mathcal{G}}_C$ implies that there is a primitive concept P such that $L_{\mathcal{A}_{T_D}}(B, P) \not\subseteq L_{\mathcal{A}_{T_C}}(A, P)$. The converse of this implication can be shown analogously.

6 Subsumption for $\mathcal{ALN}$

$\mathcal{ALN}$ is an extension of $\mathcal{FL}_0$ which additionally provides us with primitive negation and number restrictions, i.e., atomic concepts of the form $\neg P$, $(\geq n\ R)$, and $(\leq n\ R)$ where P is a primitive concept, n a nonnegative integer, and R a role name.

In both approaches, number restrictions and negated primitive concepts are treated like new primitive concepts. In the description graphs, they may also occur in node labels. In the automata-theoretic approach, they give rise to new states in the semi-automaton of C, and to additional inclusion conditions. Notice that the automaton $\mathcal{A}_{T_C}$ corresponding to an $\mathcal{ALN}$-concept description C also has a weak tree structure (see Definition 13). Thus, even for $\mathcal{ALN}$-concept descriptions C there is no exponential blow up in constructing the powerset automaton corresponding to the acyclic $\mathcal{ALN}$-terminology T_C of C.

However, since both primitive negation and number restrictions may cause inconsistencies, this straightforward extension is not sufficient to obtain a complete subsumption algorithm.

Example 25 Consider the $\mathcal{ALN}$-concept description $C' := \forall S.Q \sqcap \forall R.(P \sqcap Q \sqcap \forall S.\forall S.(Q \sqcap \neg Q) \sqcap \forall S.(\geq 1\ S))$. This description can be represented by the defined concept A' in the $\mathcal{ALN}$-terminology

$$
\begin{array}{lll}
T_{C'}\colon & A' & = \forall S.A'_1 \sqcap \forall R.A'_2 \\
& A'_1 & = Q \\
& A'_2 & = P \sqcap Q \sqcap \forall S.A'_{21} \sqcap \forall S.A'_{22} \\
& A'_{21} & = \forall S.A'_{211} \\
& A'_{211} & = Q \sqcap \neg Q \\
& A'_{22} & = (\geq 1\ S).
\end{array}
$$

The semi-automaton of C' is depicted in Figure 7.

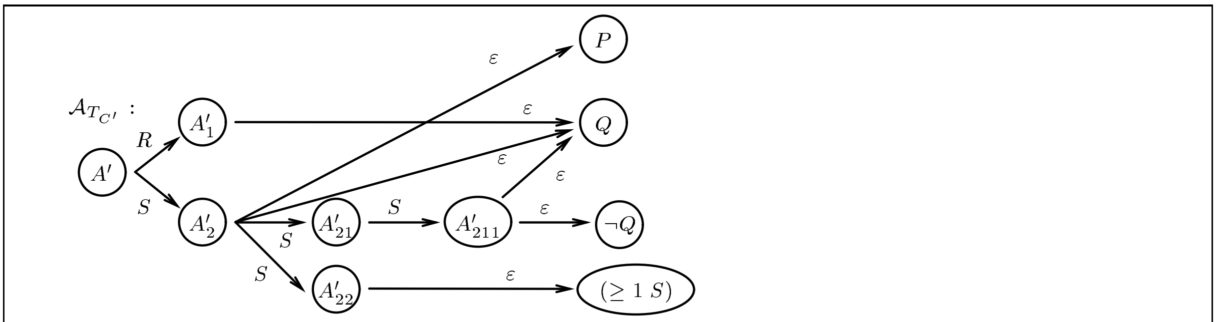

Figure 7: The semi-automaton of C'.

The powerset automaton of C' and the description graph of C' are shown in Figure 8.

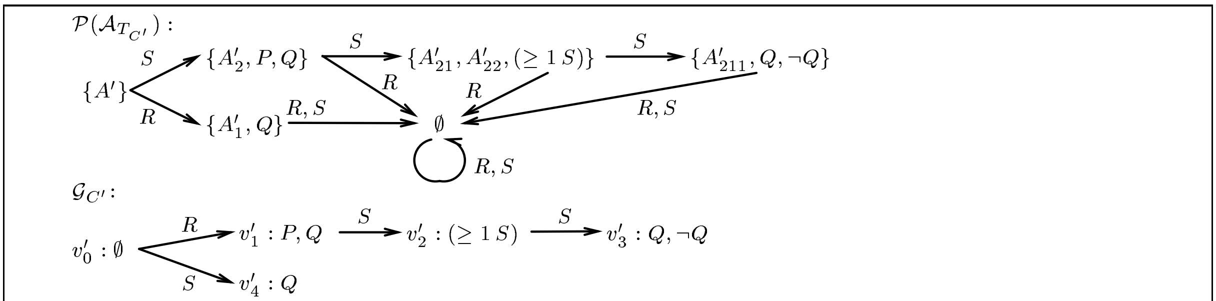

Figure 8: The powerset automaton and $\mathcal{FL}_0$-canonical description graph of C'.

Now consider the concept description C from Example 8. We have $C' \sqsubseteq C$, even though $RS \in L_{\mathcal{A}_{T_C}}(A, P) \cap \overline{L_{\mathcal{P}(\mathcal{A}_{T_{C'}})}(A', P)}$ and the path with label RS in $\mathcal{G}_C$ does not have a corresponding more specific path in $\mathcal{G}_{C'}$.

In the next two sections, we describe in detail the extensions that are made in both approaches to obtain a sound and complete subsumption algorithm. Thereafter, we extend our comparison from Section 5 to $\mathcal{ALN}$ and illustrate the 1-1-correspondence between these extensions.

6.1 The automata theoretic approach for $\mathcal{ALN}$

In [12], subsumption has been characterized for cyclic $\mathcal{ALN}$-terminologies both for descriptive and fixed-point semantics. As for $\mathcal{FL}_0$, we restrict our attention to *acyclic* $\mathcal{ALN}$-terminologies. Thus, we do not have to distinguish different semantics.

In Section 3.2, we have seen that for an $\mathcal{FL}_0$-terminology T the languages $L_{\mathcal{A}_T}(A,P)$ represent exactly those value restrictions for P subsuming A, i.e., $A \sqsubseteq_T \forall W.P$ iff $W \in L_{\mathcal{A}_T}(A,P)$. Since inconsistencies are expressible in $\mathcal{ALN}$, this equivalence does not hold for $\mathcal{ALN}$-terminologies. There may be words such that $A \sqsubseteq_T \forall W.\bot$ (RS in Example 25). These words are called A-*excluding* words; let $E(A)$ denote the set of these words. A-excluding words imply value restrictions that are not explicitly represented in $L_{\mathcal{A}_T}(A,P)$, i.e., even if the A-excluding word W is not contained in $L_{\mathcal{A}_T}(A,P)$ it obviously follows $A \sqsubseteq_T \forall W.P$. Thus, in order to represent the value restrictions that are satisfied by A, beside $L_{\mathcal{A}_T}(A,P)$, additionally, A-excluding words have to be taken into account. Since we are interested in an automata theoretic characterization of subsumption, it is necessary to characterize A-excluding words based on $\mathcal{A}_T$.

Proposition 26 (Exclusion) *Let T denote an acyclic $\mathcal{ALN}$-terminology, and let S denote the minimal subset of Σ^* such that the following conditions hold:*

1. *If there is a primitive concept P or conflicting number restriction $(\geq l\ R)$ and $(\leq r\ R)$, $l > r$, such that $W \in L_{\mathcal{A}_T}(A,P) \cap L_{\mathcal{A}_T}(A,\neg P)$ or $W \in L_{\mathcal{A}_T}(A,(\geq l\ R)) \cap L_{\mathcal{A}_T}(A,(\leq r\ R))$, then $W \in S$.*
2. *If $WR \in \Sigma^*$, $W \in \Sigma^*$, $R \in \Sigma$, and $W \in L_{\mathcal{A}_T}(A,(\leq 0\ R))$, then $WR \in S$.*
3. *If $WR \in S$, $W \in \Sigma^*$, $R \in \Sigma$, and $W \in L_{\mathcal{A}_T}(A,(\geq l\ R))$, $l \geq 1$, then $W \in S$.*
4. *If $W \in S$ and $V \in \Sigma^*$, then $WV \in S$.*

Then $S = E(A)$.

Proof. Consequence of Lemma 35 in [11]. Since the proof is lengthy and technical it is omitted here. □

In order to simplify the comparison of structural subsumption and the automata theoretic approach, the characterization stated in Proposition 26 differs from the definition of $E(A)$ in [12, 11]. However, it can easily be verified that these descriptions of $E(A)$ are equivalent.

To illustrate Proposition 26 we consider the concept C' from Example 25. Obviously, $RSS \in E(A')$ since $RSS \in L_{\mathcal{A}_{T_{C'}}}(A',Q) \cap L_{\mathcal{A}_{T_{C'}}}(A',\neg Q)$, i.e., any RSS-successor of an individual in A' must belong both to Q and $\neg Q$, which is impossible. Furthermore, by Proposition 26, 3. $RS \in L_{\mathcal{A}_{T_{C'}}}(A', (\geq 1\ S))$ implies $RS \in E(A')$. This is motivated by the following fact: Every RS-successor of an individual in A' also has an RSS-successor. Since $RSS \in E(A')$ means that individuals in A' cannot have RSS-successors, this implies that they cannot have RS-successors. Finally, since $RS \in E(A')$ we know by Proposition 26, 4 that $RSW \in E(A')$ for all $W \in \Sigma^*$. These words must be contained in $E(A')$ since $A' \sqsubseteq_{T_{C'}} \forall RS.\bot$ implies $A' \sqsubseteq_{T_{C'}} \forall RSW.\bot$. Now, it is not hard to see that $E(A') = RS\Sigma^*$.

It can be shown that $L_{\mathcal{A}_T}(A, P) \cup E(A)$ contains exactly those words W such that $A \sqsubseteq_T \forall W.P$. Intuitively, $L_{\mathcal{A}_T}(A, P) \cup E(A)$ represents all value restrictions for P that are satisfied by A. For primitive negation $\neg P$ there is an analogous set.

For number restrictions, beside excluding words, another phenomenon comes into the picture. If $W \in L_{\mathcal{A}_T}(A, (\geq l\ R))$, then not only $A \sqsubseteq_T \forall W.(\geq r\ R)$ but also $A \sqsubseteq_T \forall W.(\geq l\ R)$ holds for all $l \leq r$. Therefore, the value restrictions for $(\geq l\ R)$ that are satisfied for A include all words in $\bigcup_{r \geq l} L_{\mathcal{A}_T}(A, (\geq r\ R))$. In fact, it can be shown that $\bigcup_{r \geq l} L_{\mathcal{A}_T}(A, (\geq r\ R)) \cup E(A)$ contains all value restrictions for $(\geq l\ R)$, $l > 0$, that are satisfied by A.

For $\leq$-restrictions $\bigcup_{r \leq l} L_{\mathcal{A}_T}(A, (\leq r\ R)) \cup E(A)$ is not sufficient for $(\leq l\ R)$. If $WR \in E(A)$, then it follows $A \sqsubseteq_T \forall WR.\bot$, Consequently, it holds $A \sqsubseteq_T \forall W.(\leq 0\ R)$. More generally, it can be observed that $W \in E(A)R^{-1}$ implies $A \sqsubseteq_T \forall W.(\leq l\ R)$ for all $l \geq 0$.[4] Now, it can be shown that $\bigcup_{r \leq l} L_{\mathcal{A}_T}(A, (\leq r\ R)) \cup E(A)R^{-1}$ represents exactly those value restrictions for $(\leq l\ R)$ that are satisfied by A.

Intuitively, A is subsumed by B if the set of value restrictions that has to be satisfied by B are contained in those which are satisfied by A, i.e., A satisfies at least those value restrictions that must hold for individuals in B.

Theorem 27 (Characterizing subsumption for $\mathcal{ALN}$) *Let T_A and T_B be acyclic $\mathcal{ALN}$-terminologies defined over the same set of primitive concepts and roles. Furthermore, let A be defined in T_A and B in T_B. Then $A \sqsubseteq_{T_A, T_B} B$ iff*

1. *for all primitive concepts P it holds that*
   $L_{\mathcal{A}_{T_B}}(B, P) \subseteq L_{\mathcal{A}_{T_A}}(A, P) \cup E(A)$,
2. *for all primitive negation $\neg P$ it holds that*
   $L_{\mathcal{A}_{T_B}}(B, \neg P) \subseteq L_{\mathcal{A}_{T_A}}(A, \neg P) \cup E(A)$,
3. *for all $\geq$-restrictions $(\geq l\ R)$, $l > 0$, it holds that*
   $L_{\mathcal{A}_{T_B}}(B, (\geq l\ R)) \subseteq \bigcup_{r \geq l} L_{\mathcal{A}_{T_A}}(A, (\geq r\ R)) \cup E(A)$, *and*
4. *for all $\leq$-restrictions $(\leq l\ R)$ it holds that*
   $L_{\mathcal{A}_{T_B}}(B, (\leq l\ R)) \subseteq \bigcup_{r \leq l} L_{\mathcal{A}_{T_A}}(A, (\leq r\ R)) \cup E(A)R^{-1}$.

Proof. See [11]. □

In our example we have $L_{\mathcal{A}_{T_C}}(A, P) = \{R, RS\} \subseteq L_{\mathcal{A}_{T_{C'}}}(A', P) \cup E(A') = \{R\} \cup RS\Sigma^*$ as well as $L_{\mathcal{A}_{T_C}}(A, Q) = \{R, S\} \subseteq L_{\mathcal{A}_{T_{C'}}}(A', Q) \cup E(A') = \{S, R\} \cup RS\Sigma^*$. The other languages for A are empty. Thus by Theorem 27, C' is subsumed by C.

[4] For $L \subseteq \Sigma^*$ and $R \in \Sigma$ we define $LR^{-1} := \{W \mid WR \in L\}$.

Note that although we only consider acyclic terminologies the set of excluding words are either empty or infinite. Thus, in order to use Theorem 27 for deciding subsumption inclusions of infinite languages have to be tested. This is achieved by applying automata theoretic techniques. First we have to show that the set of excluding words is regular. In fact, it turns out [12] that for an $\mathcal{ALN}$-terminology T the set $E(A)$ of A-excluding words is accepted by a certain extension of the powerset automaton of $\mathcal{A}_T$. To see this, we need the following

Definition 28 (Exclusion set) Let T be an $\mathcal{ALN}$-terminology and $\mathcal{A}_T = (\Sigma, Q, \Delta)$ the corresponding semi-automaton. The set $F_0 \subseteq Q$ is called *exclusion set* w.r.t. $\mathcal{A}_T$ if there is a non-negative integer n, a word $R_1 \cdots R_n \in \Sigma^*$, conflicting number-restrictions $(\geq l\ R)$ and $(\leq r\ R)$, $l > r$, or a primitive concept P, and for all $1 \leq i \leq n$ there are integers $m_i \geq 1$ such that for $F_i := next_\varepsilon(F_{i-1}, R_i)$, $1 \leq i \leq n$, it holds that $(\geq m_i\ R_i) \in F_{i-1}$ for all $1 \leq i \leq n$ and $(\geq l\ R), (\leq r\ R) \in F_n$ or $P, \neg P \in F_n$.

Now, an automata theoretic characterization of exclusion can be shown [12]:

Lemma 29 *Let T be an $\mathcal{ALN}$-terminology, $\mathcal{A}_T$ the corresponding semi-automaton, and A an atomic concept in T. Then*

$$E(A) = \{W \in \Sigma^* \mid \text{there is a prefix } V \text{ of } W \text{ such that } next_{\mathcal{A}_T}(\{A\}, V) \text{ is an exclusion set or there is a prefix } VR \text{ of } W,\ V \in \Sigma^*,\ R \in \Sigma \text{ such that } (\leq 0\ R) \in next_{\mathcal{A}_T}(\{A\}, V)\}.$$

Using this Lemma, we can construct a finite automaton accepting $E(A)$:

Definition 30 Let $\mathcal{P}(\mathcal{A}_T) = (\Sigma, \widehat{Q}, \widehat{\delta})$ denote the powerset automaton of $\mathcal{A}_T = (\Sigma, Q, \Delta)$ with initial state $\varepsilon\text{-}\mathsf{closure}(A)$. We extend $\mathcal{P}(\mathcal{A}_T)$ to $\mathcal{B}_T^* = (\Sigma, Q^*, \varepsilon\text{-}\mathsf{closure}(A), \Delta^*, \{q\})$ by a new state q which is the final state of $\mathcal{B}_T^*$. For every exclusion set $F \subseteq Q$ we add a transition (F, ε, q) in $\mathcal{B}_T^*$ and for every state $F \subseteq Q$ in $\mathcal{B}_T^*$ and every $\leq$-restriction $(\leq 0\ R) \in F$ we add the transition (F, R, q). Finally, we add (q, R, q) for every $R \in \Sigma$.

If, furthermore, every $F \subseteq Q$ that contains P is a final state in $\mathcal{B}_T^*$ then this automaton accepts the language $L_{\mathcal{A}_T}(A, P) \cup E(A)$. Analogously, one can define final states in $\mathcal{B}_{T_C}^*$ such that the languages on the right-hand side of the inclusions in Theorem 27 for primitive negation, $\geq$-restrictions, and $\leq$-restrictions are accepted. Note, that for $\leq$-restrictions some transitions must be added as well, since we are faced with $E(A)R^{-1}$ instead of $E(A)$. This extension can easily be achieved. Now, one can test the inclusions in Theorem 27 as pointed out in Section 3.3 using $\mathcal{A}_{T_D}$ and $\mathcal{B}_{T_C}^*$.

6.2 Structural subsumption for $\mathcal{ALN}$

In this section we extend our notion of description graphs to $\mathcal{ALN}$ by allowing for negated primitive concepts and number restrictions in the labels of nodes, i.e., an $\mathcal{ALN}$-description graph $\mathcal{G} = (V, E, v_0, l)$ over a set $\mathcal{C}$ of primitive concepts and a set Σ of role names is a finite tree with root v_0 such that

- $E \subseteq V \times \Sigma \times V$ and
- for all $v \in V$ $l(v)$ is a finite subset of
$\mathcal{C} \cup \{\neg P \mid P \in \mathcal{C}\} \cup \{(\geq n\ R) \mid n \in \mathbb{N},\ R \in \Sigma\} \cup \{(\leq n\ R) \mid n \in \mathbb{N},\ R \in \Sigma\}$.

The semantics of $\mathcal{ALN}$-description graphs is defined in

Definition 31 (Extension of $\mathcal{ALN}$-description graphs) Let $\mathcal{I}$ be an interpretation of $\mathcal{C}$ and Σ, $\mathcal{G} = (V, E, v_0, l)$ an $\mathcal{ALN}$-description graph over $\mathcal{C}$ and Σ.
The *extension* of a node $v \in V$ is recursively defined by $x \in v^I$ iff

- $x \in P^I$ for all $P \in l(v)$ and
- $x \notin P^I$ for all $\neg P \in l(v)$ and
- $|\{y \mid (x, y) \in R^I\}| \leq n$ for all $(\leq n\ R) \in l(v)$ and
- $|\{y \mid (x, y) \in R^I\}| \geq n$ for all $(\geq n\ R) \in l(v)$ and
- for all $vRv' \in E$ and $y \in dom(I)$ with $(x, y) \in R^I$ it holds that $y \in v'^I$.

The *extension* of $\mathcal{G}$ is defined by $\mathcal{G}^I := v_0^I$.

The recursive algorithm in Figure 3 for translating $\mathcal{FL}_0$-concept descriptions into $\mathcal{FL}_0$-description graphs can be easily extended to $\mathcal{ALN}$. We are concerned with $\mathcal{ALN}$-concept descriptions of the form

$$\begin{array}{rcl} C & = & P_1 \sqcap \ldots \sqcap P_n \sqcap \neg Q_1 \sqcap \ldots \sqcap \neg Q_k \sqcap \\ & & (\leq \nu_1\ S_1) \sqcap \ldots \sqcap (\leq \nu_l\ S_l) \sqcap (\geq \mu_1\ T_1) \sqcap \ldots \sqcap (\geq \mu_r\ T_r) \sqcap \\ & & \forall R_1.C_1 \sqcap \ldots \sqcap \forall R_m.C_m. \end{array}$$

We replace the definition of the label of v_0 in Figure 3 by

$$l(v_0) := \{P_1, \ldots, P_n, \neg Q_1 \ldots, \neg Q_k, (\leq \nu_1\ S_1), \ldots, (\leq \nu_l\ S_l), (\geq \mu_1\ T_1), \ldots, (\geq \mu_r\ T_r)\}.$$

The algorithm obtained by this modification yields the $\mathcal{ALN}$-description graph $\mathcal{G}_C$ of an $\mathcal{ALN}$-concept description C. Analogously to $\mathcal{FL}_0$, the translation is sound.

Lemma 32 (Equivalence of concepts and description graphs for $\mathcal{ALN}$) *Let C be an arbitrary $\mathcal{ALN}$-concept description and $\mathcal{G}_C$ the description graph of C. Then for all interpretations $\mathcal{I}$ it holds that $C^I = \mathcal{G}_C^I$.*

Proof. See [13]. □

Example 25 shows that Theorem 21 does not hold for $\mathcal{ALN}$-concept descriptions C and D. Notice that the description graph $\mathcal{G}_{C'}$ of C' depicted in Figure 8 is already deterministic, i.e., the normalization rule in Figure 5 is not applicable to $\mathcal{G}_{C'}$. We will

Let $\mathcal{G} = (V, E, v_0, l)$ be a description graph. $\mathcal{G}' = (V', E', v_0, l')$ denotes the description graph that is obtained from $\mathcal{G}$ by applying one of the following rules.

1. Let $v \in V$ with $n > 1$ R-successors $v_1, \dots, v_n$ in E and v' a new node not occuring in V. Then $\mathcal{G}'$ is defined by
   - $V' := V \setminus \{v_1, \dots, v_n\} \cup \{v'\}$ and
     $E' := E[v_i/v' \mid i = 1 \dots n]$ (each v_i is replaced by v' in E),
   - $l(v') := \bigcup_{i=1\dots n} l(v_i)$ and $l(w) := l(w)$, $w \in V' \setminus \{v'\}$.
2. Let $v, w \in V$ with $vRw \in E$ and $\{P, \neg P\} \subseteq l(w)$. Then $\mathcal{G}'$ is defined by
   - $V' := V \setminus (\{w\} \cup \{w' \mid \text{exists a path from } v \text{ to } w'\})$ and
     $E' := E \cap V' \times \Sigma \times V'$,
   - $l'(v') := l(v')$, $v' \in V' \setminus \{v\}$ and $l'(v) := l(v) \cup \{(\leq 0\ R)\}$.
3. Let $v, w \in V$ with $vRw \in E$, $\{(\geq n\ S), (\leq m\ S)\} \subseteq l(w)$, and $m > n$. Then $\mathcal{G}'$ is defined by
   - $V' := V \setminus (\{w\} \cup \{w' \mid \text{exists a path from } w \text{ to } w'\})$ and
     $E' := E \cap V' \times \Sigma \times V'$,
   - $l'(v') := l(v')$, $v' \in V' \setminus \{v\}$ and $l'(v) := l(v) \cup \{(\leq 0\ R)\}$.
4. Let $v \in V$ with $(\leq 0\ R) \in l(v)$ and $vRw \in E$. Then $\mathcal{G}'$ is defined by
   - $V' := V \setminus (\{w\} \cup \{w' \mid \text{exists a path from } w \text{ to } w'\})$ and
     $E' := E \cap V' \times \Sigma \times V'$,
   - $l'(v') := l(v')$, $v' \in V'$.
5. Let $v \in V$ with $k > 1$ number restrictions $\{(\leq n_1\ R), \dots, (\leq n_k\ R)\} \subseteq l(v)$. Then $\mathcal{G}'$ is defined by
   - $V' := V$ and $E' := E$,
   - $$\begin{aligned} l(v) :=\ & l(v) \setminus \{(\leq n_1\ R), \dots, (\leq n_k\ R)\} \cup \\ & \{(\leq \mathsf{min}\{n_1, \dots, n_k\}\ R)\}, \\ l(w) :=\ & l(w),\ w \in V' \setminus \{v\}. \end{aligned}$$
6. Let $v \in V$ with $k > 1$ number restrictions $\{(\geq n_1\ R), \dots, (\geq n_k\ R)\} \subseteq l(v)$. Then $\mathcal{G}'$ is defined by
   - $V' := V$ and $E' := E$,
   - $$\begin{aligned} l(v) :=\ & l(v) \setminus \{(\geq n_1\ R), \dots, (\geq n_k\ R)\} \cup \\ & \{(\geq \mathsf{max}\{n_1, \dots, n_k\}\ R)\}, \\ l(w) :=\ & l(w),\ w \in V' \setminus \{v\}. \end{aligned}$$
7. If $\{P, \neg P\} \subseteq l(v_0)$ or $\{(\geq n\ R), (\leq m\ R)\} \subseteq l(v_0)$, $m > n$, then $\mathcal{G}'$ is defined by
   - $V' := \{v_0\}$ and $E' := \emptyset$ and
   - $l'(v_0) := \bot$.

Figure 9: Normalization rules for $\mathcal{ALN}$-description graphs

$\widehat{\mathcal{G}}_{C'}$:

$$\begin{tikzcd}[column sep=small] & \widehat{v}'_1 : P, Q, (\leq 0\ S) \\ \widehat{v}'_0 : \emptyset \arrow[ur, "R"] \arrow[dr, "S"'] & \\ & \widehat{v}'_2 Q \end{tikzcd}$$

Figure 10: The $\mathcal{ALN}$-canonical description graph of C'.

deal with the problems caused by inconsistencies by applying additional normalization rules when computing the canonical description graph [5, 6, 13] (see Figure 9).

As for $\mathcal{FL}_0$-description graphs we first obtain a deterministic graph by merging all R-successor nodes of a node v to one new R-successor of v (Rule 1).

The rules 2, 3, and 4 cope with nodes labeled by inconsistent sets, i.e., nodes v with $\{P, \neg P\} \subseteq l(v)$ or $\{(\geq l\ S), (\leq r\ S)\} \subseteq l(v)$, $l > r$. Nodes labeled by inconsistent sets and the edges leading to these nodes are removed. In addition, if there was an edge labeled R from node v to the inconsistent node, the label of v is extended by $(\leq 0\ R)$ (rules 2, 3). This is due to the equivalence $\forall R.\bot \equiv (\leq 0\ R)$. For the same reason, we have to remove each subgraph with root v if the label of the R-predecessor of v contains $(\leq 0\ R)$ (rule 4).

If the root v_0 is labeled by an inconsistent set, then the whole concept is inconsistent. In this case, we remove all nodes except the root and all edges and label v_0 by $\bot$ (rule 7).

The rules 5 and 6 deal with number restrictions. Using $(\geq n\ R) \sqsubseteq (\geq m\ R)$ iff $n \geq m$ and $(\leq n\ R) \sqsubseteq (\leq m\ R)$ iff $n \leq m$, we can reduce all $\geq$-restrictions and all $\leq$-restrictions for an $R \in \Sigma$ to one $\geq$-restriction and one $\leq$-restriction, respectively, in the label of a node v.

Example 33 (Example 24 continued)
Consider the description graph $\mathcal{G}_{C'} = (V', E', v'_0, l')$ of C' in Figure 8. Because of $\{Q, \neg Q\} \subseteq l'(v'_3)$, the node v'_3 and the edge $v'_2 S v'_3$ are removed, and $(\leq 0\ S)$ is added to the label of v'_2. Now, v'_2 is labeled with $\{(\geq 1\ S), (\leq 0\ S)\}$, which is again inconsistent. Consequently, it is removed, and $(\leq 0\ S)$ is added to the label of the S-predecessor node v'_1. The description graph obtained this way is depicted in Figure 10.

As for $\mathcal{FL}_0$ each iterated application of the normalization rules terminates since $|\mathcal{G}| > |\mathcal{G}'|$ if $\mathcal{G}'$ is obtained from $\mathcal{G}$ by applying one of the rules in Figure 9. As mentioned above, each rule is based on an equivalence between concept descriptions, e.g., $\forall R.C \sqcap \forall R.D \equiv \forall R.(C \sqcap D)$. Thus, it is not hard to see, that the rules are sound, i.e., if $\mathcal{G}'$ is obtained from $\mathcal{G}$, then it is $\mathcal{G}^I = \mathcal{G}'^I$ for all interpretations $\mathcal{I}$.

In order to distinguish the two normal forms used in the structural approach for $\mathcal{FL}_0$ and $\mathcal{ALN}$, respectively, we refer to the description graph $\mathcal{G}'_C$ that is obtained from $\mathcal{G}_C$ by applying only the first normalization rule in Figure 9 as the *$\mathcal{FL}_0$-canonical description graph.* The description graph $\widehat{\mathcal{G}}_C$ that is obtained from $\mathcal{G}_C$ by applying all normalization rules in Figure 9 as long as possible is called *$\mathcal{ALN}$-canonical description graph.*

Definition 34 ($\mathcal{ALN}$-canonical description graphs) Let C be an $\mathcal{ALN}$-concept description and $\mathcal{G}_C$ the description graph of C. The *$\mathcal{ALN}$-canonical description graph of $\mathcal{G}_C$*

is defined as the description graph $\widehat{\mathcal{G}}_C$ that is obtained from $\mathcal{G}_C$ by an iterated application of the normalization rules in Figure 9 such that no more rule is applicable to $\widehat{\mathcal{G}}_C$.

Notice that the size of the description graph $\mathcal{G}_C$ as well as the size of the $\mathcal{ALN}$-canonical description graph $\widehat{\mathcal{G}}_C$ of an $\mathcal{ALN}$-concept description is linear in the size of C.

Before we can formalize structural subsumption for $\mathcal{ALN}$-concept descriptions, we have to generalize the notion of *more specific paths* [6, 13].

Definition 35 (More specific nodes and paths)
Let $\mathcal{G} = (E, V, v_0, l)$ and $\mathcal{G}' = (V', E', v_0', l')$ be $\mathcal{ALN}$-description graphs.
A node $v \in V$ is *more specific* than a node $v' \in V'$ iff

- for each primitive concept $P \in l'(v')$ it is $P \in l(v)$,
- for each negated primitive concept $\neg P \in l'(v')$ it is $\neg P \in l(v)$,
- for each $(\geq \mu' \ R) \in l'(v')$, there exists $(\geq \mu \ R) \in l(v)$ with $\mu \geq \mu'$, and
- for each $(\leq \nu' \ R) \in l'(v')$ there exists $(\leq \nu \ R) \in l(v)$ with $\nu \leq \nu'$.

A rooted path $p = v_0 R_1 v_1 \dots v_{n-1} R_n v_n$ in $\mathcal{G}$ is *more specific* than a rooted path $p' = v_0' R_1' v_1' \dots v_{m-1}' R_m' v_m'$ in $\mathcal{G}'$ iff

- $R_i = R_i'$ for $1 \leq i \leq \mathsf{min}(m, n)$,
- for all $0 \leq i \leq \mathsf{min}(m, n)$ it is v_i more specific than v_i', and
- if $n < m$, then $(\leq 0 \ R_{n+1}') \in l(v_n)$.

The conditions on more specific nodes v and v' ensure that the conditions given by atomic concepts in the label of v' are satisfied by each instance of v. As an example consider the node $\widehat{v}_1'$ in Figure 10 and a node v labeled with $l(v) = \{P, (\leq 1 \ S)\}$. Obviously, $\widehat{v}_1'$ is more specific than v and it holds that

$$\bigsqcap_{C \in \widehat{l'}(\widehat{v}_1')} C = P \sqcap Q \sqcap (\leq 0 \ S) \ \sqsubseteq \ P \sqcap (\leq 1 \ S) = \bigsqcap_{C \in l(v)} C.$$

More generally, the conjunction of all atomic concepts in $l'(v')$ subsumes the conjunction of all atomic concepts in $l(v)$ if v is more specific than v'.

Due to number restrictions of the form $(\leq 0 \ R)$, a path, which is more specific than a path p', can be shorter than p'. To be more precise, let $\mathcal{G} = (V, E, v_0, l)$ and $\mathcal{G}' = (V', E', v_0', l')$ be description graphs. If v is the W-successor node of the root v_0 and $(\leq 0 \ R) \in l(v)$, then each instance x of v_0 has no WR-successor. Thus, all conditions on WR-successors v' of v_0' in $\mathcal{G}'$ are satisfied trivially and the rooted path with label W in $\mathcal{G}$ is more specific than a rooted path p' in $\mathcal{G}'$ with label WRW', $W' \in \Sigma^*$.

In order to obtain a complete structural subsumption test using $\mathcal{ALN}$-canonical description graphs and the extended notion of more specific paths we have to make an assumption on the subsumer D [13]. Since $\top$ is expressible in $\mathcal{ALN}$ by $(\geq 0\ R)$, concepts equivalent to $\top$, e.g., $\forall S.(\geq 0\ R)$, must be taken into account.

It is easy to see, that D is equivalent to $\top$ iff each atomic concept D' in the labels of the nodes in $\mathcal{G}_D$ is of the form $(\geq 0\ R)$. Intuitively, if there is an atomic concept D' in $\mathcal{G}_D$ not of the form $(\geq 0\ R)$, then there is at most one non-trivial restriction to an instance of D and therefore, D is not equivalent to $\top$. It is not hard to see [13], that $D \equiv \top$ can be decided in time polynomial in the size of D. In order to simplify the structural characterization of subsumption as well as the presentation of our structural subsumption algorithm, we assume that no subconcept of the form $(\geq 0\ R)$ occurs in the subsumer D. Thus, w.l.o.g. we reduce our attention to subsumers that are not equivalent to $\top$. Notice that this assumption is stated explicitly in the automata theoretic approach by the condition "$l > 0$" in Theorem 27, 3.

Theorem 36 (Structural subsumption for $\mathcal{ALN}$) *Let C, D be $\mathcal{ALN}$-concept descriptions, $\widehat{\mathcal{G}}_C = (\widehat{V}, \widehat{E}, \widehat{v}_0, \widehat{l})$ the canonical description graph of C and $\mathcal{G}_D = (V, E, v_0, l)$ the description graph of D. Then $C \sqsubseteq D$ iff C is equivalent to $\bot$, i.e., $\widehat{l}(\widehat{v}_0) = \bot$, or if for each rooted path p in $\mathcal{G}_D$ there exists a more specific rooted path $\widehat{p}$ in $\widehat{\mathcal{G}}_C$.*

Proof. See [13]. □

An algorithm deciding subsumption of two $\mathcal{ALN}$-concept descriptions C and D based on Theorem 36 can be described as follows: Consider the (canonical) description graphs $\widehat{\mathcal{G}}_C = (\widehat{V}, \widehat{E}, \widehat{v}_0, \widehat{l})$ and $\mathcal{G}_D = (V, E, v_0, l)$. For each $W \in \Sigma^*$ and W-successor v of v_0 in $\mathcal{G}_D$ we test

1. if there exists a proper prefix W' of W and a W'-successor node $\widehat{v}$ of $\widehat{v}_0$ in $\widehat{\mathcal{G}}_C$ such that
   (a) $W = W'RW''$, $R \in \Sigma$ and $W'' \in \Sigma^*$,
   (b) each node v' on the path labeled with W' in $\widehat{\mathcal{G}}_C$ is more specific than the corresponding node in $\mathcal{G}_D$, and
   (c) $(\leq 0\ R) \in \widehat{l}(\widehat{v})$ or
2. if $\widehat{v}$ is the W-successor node of $\widehat{v}_0$ in $\widehat{\mathcal{G}}_C$, wether $\widehat{v}$ is more specific than v or not.

If (1) and (2) are not satisfied, than $C \not\sqsubseteq D$; otherwise $C \sqsubseteq D$ (see [13] for a formal algorithm).

7 Extending the comparison to $\mathcal{ALN}$

In this section we show that there is a 1-1 correspondence between the extended normalization steps for canonical description graphs and the definition of A-excluding words and exclusion sets in the automata theoretic approach. More precisely, we will point out that adding the set $E(A)$ to the right-hand side of the inclusion statements in the automata-theoretic characterization corresponds to the additional normalization done in the structural approach.

First notice that Lemma 23 still holds for $\mathcal{ALN}$-concept descriptions, since negated primitive concepts and number restrictions are treated like primitive concepts. Furthermore, we obtain an extension to $\mathcal{ALN}$ of the mapping $\widehat{\varphi}$ from $\mathcal{FL}_0$-canonical description graphs to deterministic automata defined on page 19. More precisely, computing the $\mathcal{FL}_0$-canonical description graph directly corresponds to defining the transition function δ' of the deterministic automaton $\mathcal{B}_{T_C} = (\Sigma, Q', \delta')$ and we get a recursively defined bijective mapping $\widehat{\varphi} : V' \longrightarrow Q'$ from the $\mathcal{FL}_0$-canonical description graph $\mathcal{G}'_C = (V', E', v_0, l')$ to the deterministic automaton $\mathcal{B}_{T_C}$ of C such that

- $\widehat{\varphi}(v_0) := \varepsilon\text{-}\mathsf{closure}(A)$,[5]
- $\widehat{\varphi}(v) := \delta'(\varepsilon\text{-}\mathsf{closure}(A), W)$ if v is a W-successor of v_0 in $\mathcal{G}'_C$,
- $l(v) = \widehat{\varphi}(v) \setminus \mathcal{D}_{T_C}$ for all $v \in V$, and
- $vRw \in E$ iff $\delta'(\widehat{\varphi}(v), R) = \widehat{\varphi}(w)$ for all $R \in \Sigma$.

We are now interested in the relation between the additional normalization rules in Figure 9 on the one hand and the notions of A-excluding words and exclusion sets on the other hand. Let $\mathcal{G}'_C = (V', E', v_0, l')$ be the $\mathcal{FL}_0$-canonical description graph of C, $\mathcal{B}_{T_C} = (\Sigma, Q', \delta')$ the deterministic automaton of C with defined concept A of C.

Now assume that one of the normalization rules 2 or 3 is applicable to $\mathcal{G}'_C$, i.e., there exists a node $v \in V' \setminus \{v_0\}$ with $\{P, \neg P\} \subseteq l'(v)$ or $\{(\geq l\ S), (\leq r\ S)\} \subseteq l'(v)$, $l > r$. Then, $\{P, \neg P\} \subseteq \widehat{\varphi}(v) \setminus \mathcal{D}_{T_C}$ or $\{(\geq l\ S), (\leq r\ S)\} \subseteq \widehat{\varphi}(v) \setminus \mathcal{D}_{T_C}$, respectively. Obviously, if v is the W-successor node of v_0 in $\mathcal{G}'_C$, then W is an A-excluding word, i.e., $W \in E(A)$. Furthermore, by Definition 28 $\widehat{\varphi}(v)$ is an exclusion set in $\mathcal{B}_{T_C}$.

Applying rule 2 or 3 means removing the subgraph with root v from $\mathcal{G}'_C$ and adding $(\leq 0\ R)$ to the label of the R-predecessor[6] v' of v. If this new number restriction reveals an inconsistency in v', i.e., there exists $(\geq n\ R) \in l'(v')$, $n > 0$, then rule 3 is applicable again. Since $(\geq n\ R) \in \widehat{\varphi}(v') \setminus \mathcal{D}_{T_C}$, W' is also an A-excluding word (see Proposition 26, 3.), and $\widehat{\varphi}(v')$ is an exclusion set (see Definition 28).

If the normalization rule 7 is applicable to $\mathcal{G}'_C$, then C is equivalent to the empty concept $\perp$ and $\widehat{\varphi}(v_0) = \varepsilon\text{-}\mathsf{closure}(A)$ is an exclusion set.

By induction on the number of applications of the normalization rules in Figure 9 we can prove

Lemma 37 *Let C be an $\mathcal{ALN}$-concept description, $\mathcal{G}'_C = (V', E', v_0, l')$ the $\mathcal{FL}_0$-canonical description graph of C, $\mathcal{B}_{T_C} = (\Sigma, Q', \delta')$ the deterministic automaton with defined concept A of C, and $\widehat{\varphi} : V' \longrightarrow Q'$ the bijection from $\mathcal{G}'_C$ to $\mathcal{B}_{T_C}$. For each iterated application of the normalization rules in Figure 9 to $\mathcal{G}'_C$, it holds that*

[5] Let A be the defined concept of C in T_C.

[6] R and v' are uniquely determined since $\mathcal{G}'_C$ is a deterministic tree.

- *if* $(\leq 0\ R)$ *is added to the label of* v, *then the* R*-successor* $\delta'(\widehat{\varphi}(v), R)$ *of* $\widehat{\varphi}(v)$ *in* $\mathcal{B}_{T_C}$ *is an exclusion set, and*

- *if the normalization rule 7 becomes applicable, then* C *is inconsistent and* $\widehat{\varphi}(v_0) = \varepsilon\text{-}\mathsf{closure}(A)$ *is an exclusion set.*

In order to complete our comparison we have to investigate the relation between the generalized conditions on more specific paths and adding the set $E(A)$ of A-excluding words to the right hand side of the set inclusion tests required by the automata theoretic characterization.

Therefore, we consider Theorem 36 and the following proposition. Roughly speaking, we characterize $A \sqsubseteq_{t_A,T_B} B$ by means of the extended powerset automaton $\mathcal{B}^*_{T_A}$ (see Definition 30). This description can easily be verified.

Proposition 38 *With the denotations of Theorem 27 let* $\mathcal{B}^*_{T_A}$ *be defined as in Definition 30 for the terminology* T_A. *Then for all* $W \in \Sigma^*$ *it holds:*

1. $W \in L_{\mathcal{A}_{T_B}}(B, P) \cap \overline{L_{\mathcal{A}_{T_A}}(A, P) \cup E(A)}$ *iff*
   (a) *there is a path from* B *to* P *in* $\mathcal{A}_{T_B}$, *and*
   (b) $P \notin next_{\mathcal{A}_{T_A}}(A, W)$, *and*
   (c) *there is no path in* $\mathcal{B}$ *from* $\varepsilon\text{-}\mathsf{closure}(A)$ *to* q *labeled with* W.
2. $W \in L_{\mathcal{A}_{T_B}}(B, \neg P) \cap \overline{L_{\mathcal{A}_{T_A}}(A, \neg P) \cup E(A)}$ *iff*
   (a) *there is a path from* B *to* $\neg P$ *in* $\mathcal{A}_{T_B}$, *and*
   (b) $\neg P \notin next_{\mathcal{A}_{T_A}}(A, W)$, *and*
   (c) *there is no path in* $\mathcal{B}$ *from* $\varepsilon\text{-}\mathsf{closure}(A)$ *to* q *labeled with* W.
3. $W \in L_{\mathcal{A}_{T_B}}(B, (\geq l\ R)) \cap \overline{\bigcup_{r \geq l} L_{\mathcal{A}_{T_A}}(A, (\geq r\ R)) \cup E(A)}$, $l > 0$, *iff*
   (a) *there is a path from* B *to* $(\geq l\ R)$ *in* $\mathcal{A}_{T_B}$, *and*
   (b) $\bigcup_{r \geq l}\{(\geq r\ R)\} \cap next_{\mathcal{A}_{T_A}}(A, W) = \emptyset$, *and*
   (c) *there is no path in* $\mathcal{B}$ *from* $\varepsilon\text{-}\mathsf{closure}(A)$ *to* q *labeled with* W.
4. $W \in L_{\mathcal{A}_{T_B}}(B, (\leq l\ R)) \cap \overline{\bigcup_{r \leq l} L_{\mathcal{A}_{T_A}}(A, (\leq r\ R)) \cup E(A)R^{-1}}$ *iff*
   (a) *there is a path from* B *to* $(\leq l\ R)$ *in* $\mathcal{A}_{T_B}$, *and*
   (b) $\bigcup_{r \leq l}\{(\leq r\ R)\} \cap next_{\mathcal{A}_{T_A}}(A, W) = \emptyset$, *and*
   (c) *there is no path* $\mathcal{B}$ *from* $\varepsilon\text{-}\mathsf{closure}(A)$ *to* q *labeled with* WR.

Now, assume that $C \not\sqsubseteq D$. On the one hand, we know that there exists a $W \in \Sigma^*$ such that W satisfies the conditions (a)–(c) of one of the points 1.–4. in Proposition 38. On the other hand, by Theorem 36 there exists a rooted path p in $\mathcal{G}_D$ with label $W' \in \Sigma^*$ such that there exists no more specific path $\widehat{p}$ in $\widehat{\mathcal{G}}_C$. We point out the correspondence between both characterizations by

Lemma 39 *Let C, D be $\mathcal{ALN}$-concept descriptions such that $C \not\sqsubseteq D$. There exists a rooted path p in $\mathcal{G}_D$ with label W such that*

- *there exists no more specific rooted path $\widehat{p}$ in $\widehat{\mathcal{G}}_C$ and*
- *W satisfies the three conditions $(a), \dots, (c)$ of at least one of the four points 1.–4. in Proposition 38.*

Proof.

By Theorem 36, there exists a rooted path p in $\mathcal{G}_D$ such that there exists no more specific rooted path in $\widehat{\mathcal{G}}_C$. We have to consider several cases.

1. $\widehat{v}_0$ is not more specific than v_0. By Definition 35, one of the following cases holds.
   (a) There exists a primitive concept P such that $P \in l(v_0)$ and $P \notin \widehat{l}(\widehat{v}_0)$. Then it is $\varepsilon \in L_{\mathcal{A}_{T_D}}(B, P)$ and $P \notin \varepsilon\text{-}\mathsf{closure}(A)$ in $\mathcal{B}^*_{T_C}$. Since $C \not\equiv \bot$, $\varepsilon\text{-}\mathsf{closure}(A)$ is not an exclusion set, and hence there exists no transition $(\varepsilon\text{-}\mathsf{closure}(A), \varepsilon, q)$ in $\mathcal{B}^*_{T_C}$. So, W satisfies the conditions (a)–(c) of Proposition 38, 1.
   (b) There exists a primitive concept P such that $\neg P \in l(v_0)$ and $\neg P \notin \widehat{l}(\widehat{v}_0)$. Analogous to (a), the three conditions in Proposition 38, 2. are satisfied by W.
   (c) There exists $(\geq \mu' \ S) \in l(v_0)$, and there exists no $(\geq \mu \ S) \in \widehat{l}(\widehat{v}_0)$ with $\mu \geq \mu'$. By our assumption on D, it is $\mu' > 0$. Then it is $\bigcup_{r \geq l}\{(\geq l \ R)\} \cap \varepsilon\text{-}\mathsf{closure}(A) = \emptyset$, thus the conditions in Proposition 38, 3. are satisfied.
   (d) There exists $(\leq \nu' \ S) \in l(v_0)$ and there exists no $(\leq \nu \ S) \in \widehat{l}(\widehat{v}_0)$ with $\nu \leq \nu'$. Analogous to (c), the conditions in Proposition 38, 4. are satisfied.

2. Let $0 \leq m < n$ be the maximal index such that there exists $\widehat{p} = \widehat{v}_0 R_1 \widehat{v}_1 \dots \widehat{v}_{m-1} R_m \widehat{v}_m$ in $\widehat{\mathcal{G}}_C$ whereby each $\widehat{v}_i$ is more specific than v_i, $0 \leq i \leq m$. The path $\widehat{p}$ is uniquely determined by $R_1 \dots R_m$, because $\widehat{\mathcal{G}}_C$ is deterministic. Since $\widehat{p}$ is not more specific than p, it is $(\leq 0 \ R_{m+1}) \notin \widehat{l}(\widehat{v}_m)$ (see Definition 35).
   (a) There exists an R_{m+1}-successor $\widehat{v}_{m+1}$ of $\widehat{v}_m$ in $\widehat{\mathcal{G}}_C$ such that $\widehat{v}_{m+1}$ is not more specific than v_{m+1}. It follows $l(v_{m+1}) \neq \emptyset$. Similar to case 1. we can show that the conditions (a) and (b) of one of the points 1.–4. in Proposition 38 are satisfied by $W = R_1 \dots R_{m+1}$. Since $\widehat{\mathcal{G}}_C$ is canonical, it is $(\leq 0 \ R_{i+1}) \notin \widehat{l}(\widehat{v}_i)$ for $0 \leq i < m$. $(\leq 0 \ R_{i+1}) \notin \widehat{l}(\widehat{v}_i)$ for $0 \leq i \leq m$ implies that there is no path with label $R_1 \dots R_m$ from $\varepsilon\text{-}\mathsf{closure}(A)$ to the accepting sink state q in $\mathcal{B}^*_{T_C}$. So, W satisfies (a)–(c) for one of the points 1.–4. in Proposition 38.
   (b) There exists no R_{m+1}-successor $\widehat{v}$ of $\widehat{v}_m$ in $\widehat{\mathcal{G}}_C$. Without loss of generality, we may assume that the last node v in p has a non-empty label set (otherwise, p can be extended appropriately; see Section 5). Consequently, $R_1 \dots R_n$ satisfies the conditions (a) and (b) of at least one of the four points 1. –4. of Proposition 38.
   As before, there exists no path with label $R_1 \dots R_m$ form $\varepsilon\text{-}\mathsf{closure}(A)$ to the accepting sink state q in $\mathcal{B}^*_{T_C}$. Because of $\widehat{v}_m R_{m+1} \widehat{v} \in \widehat{E}$ iff $\delta'(\widehat{\varphi}(\widehat{v}_m), R_m) =$

$\widehat{\varphi}(\widehat{v})$, there exists no R_{m+1}-successor of $\widehat{\varphi}(\widehat{v}_m)$ in the deterministic automaton $\mathcal{B}_{T_C}$ of C. By construction, it follows that the rejecting sink state $\emptyset$ is the unique state reached by $R_1 \dots R_n$ in $\mathcal{B}^*_{T_C}$. In particular, there exists no path with label $R_1 \dots R_n$ from ε-$\mathsf{closure}(A)$ to q in $\mathcal{B}^*_{T_C}$. Thus, $R_1 \dots R_n$ satisfies the conditions (a)–(c) for at least one of the points 1. –4. in Proposition 38. $\square$

We summarize the results of the comparison.

- There exists an isomorphism between the automaton $\mathcal{A}_{T_C}$ and the description graph $\mathcal{G}_C$ of an $\mathcal{ALN}$-concept description C.
- We have defined a bijective mapping from the $\mathcal{FL}_0$-canonical description graph $\mathcal{G}'_C$ to the deterministic automaton $\mathcal{B}_{T_C}$.
- There is a 1-1-correspondence between the additional normalization rules for computing $\mathcal{ALN}$-canonical description graphs and the characterization of A-excluding words by terms of the automaton $\mathcal{B}^*_{T_C}$.
- We have shown that looking for a rooted path in $\mathcal{G}_D$ without a more specific rooted path in $\widehat{\mathcal{G}}_C$ corresponds to testing the set inclusions for all atomic concepts occuring in C and D in parallel.

As a consequence of these results structural subsumption algorithms based on description graphs can be seen as a special implementation of the inclusion tests required by the automata theoretic characterization of subsumption for $\mathcal{ALN}$-concept descriptions.

8 Conclusion and future work

We have shown that structural subsumption algorithms are special implementations of the language inclusion tests required by the automata-theoretic characterization of subsumption. This provides a more abstract understanding of how structural subsumption algorithms work.

More precisely, we have pointed out that there exists an isomorphic relation between the description graph $\mathcal{G}_C$ of a concept description C and the automaton $\mathcal{A}_{T_C}$ representing the acyclic terminology T_C of C. Furthermore, we introduced a bijective mapping from the canonical description graph $\widehat{\mathcal{G}}_C$ of C to the deterministic automaton $\mathcal{B}_{T_C}$.

We have seen that the canonical description graph $\widehat{\mathcal{G}}_C$ of C and the description graph of D is linear in the size of C and D, respectively. Hence, wether D subsumes C can be decided in time polynomial in the size of C and D.

The size of the automaton $\mathcal{A}_{T_C}$ corresponding to a concept description C is also linear in the size of C. Due to the weak tree structure of $\mathcal{A}_{T_C}$, the powerset automaton of C is also linear in the size of C. The inclusion of regular languages can be decided in time polynomial in the size of the automata defining these languages. Consequently, one can decide subsumption of concept descriptions within the automata theoretic approach in time polynomial in the size of the concept descriptions.

In future work, we will extend the comparison to cyclic terminologies, by comparing the automata-theoretic characterization of subsumption with the structural subsumption algorithm for cyclic terminologies realized in K-Rep [7].

The comparison between the structural and the automata-theoretic approach can also be extended to other inference tasks such as computing the least common subsumer (lcs) of $\mathcal{ALN}$-concept descriptions. Again, the algorithm for computing the lcs based on description graphs [6] can be seen as a special implementation of the automata theoretic characterization of the lcs [3, 4]. An advantage of the automata-theoretic approach is that it easily carries over to computing the lcs for concepts defined by cyclic terminologies [3, 4].

References

[1] F. Baader. Using automata theory for characterizing the semantics of terminological cycles. *Annals of Mathematics and Artificial Intelligence*, 18(2–4):175–219, 1996.

[2] F. Baader. Using automata theory for characterizing the semantics of terminological cycles. *Annals of Mathematic and Artificial Intelligence*, 18:175–219, 1996.

[3] F. Baader and R. Küsters. Computing the least common subsumer and the most specific concept in the presence of cyclic $\mathcal{ALN}$-concept descriptions. In *22. Deutsche Jahrestagung für Künstliche Intelligenz KI'98*, Bremen, 1998.

[4] F. Baader and R. Küsters. Least common subsumer computation w.r.t. cyclic $\mathcal{ALN}$-terminologies. In *Proceedings of the 1998 International Workshop on Description Logic, DL'98*, Trento, Italy, 1998.

[5] A. Borgida and P. Patel-Schneider. A semantics and complete algorithm for subsumption in the CLASSIC description logic. *J. of Artificial Intelligence Research*, 1:277–308, 1994.

[6] William W. Cohen and Haym Hirsh. Learning the CLASSIC description logic: Theoretical and experimental results. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Fourth International Conference (KR94)*. Morgan Kaufmann, 1994.

[7] R. Dionne, E. Mays, and F.J. Ohles. The equivalence of model-theoretic and structural subsumption in description logics. In *Proceedings of IJCAI'93*, 1993.

[8] Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, and Andrea Schaerf. Reasoning in description logics. In *Foundation of Knowledge Representation*, CSLI-Publications, pages 191–236. Cambrige University Press, 1996.

[9] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* Freeman, San Francisco, 1979.

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

[11] R. Küsters. Characterizing the semantics of terminological cycles in $\mathcal{ALN}$ using finite automata. Technical report, LuFG Theoretical Computer Science, RWTH Aachen, 1997. See http://www-lti.informatik.rwth-aachen.de/Forschung/Papers.html.

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

[13] R. Molitor. Structural Subsumption for $\mathcal{ALN}$. Technical report, LuFG Theoretical Computer Science, RWTH Aachen, Germany, 1998. See http://www-lti.informatik.rwth-aachen.de/Forschung/Papers.html.

[14] B. Nebel. Terminological reasoning is inherently intractable. *Journal on Artificial Intelligence*, 43(2):235–249, 1990.