Dov M. Gabbay
Andrzej Szałas

Annotation Theories over Finite Graphs

Abstract. In the current paper we consider theories with vocabulary containing a number of binary and unary relation symbols. Binary relation symbols represent labeled edges of a graph and unary relations represent unique annotations of the graph's nodes. Such theories, which we call *annotation theories*, can be used in many applications, including the formalization of argumentation, approximate reasoning, semantics of logic programs, graph coloring, etc. We address a number of problems related to annotation theories over finite models, including satisfiability, querying problem, specification of preferred models and model checking problem.

We show that most of considered problems are NPTime- or co-NPTime-complete. In order to reduce the complexity for particular theories, we use second-order quantifier elimination. To our best knowledge none of existing methods works in the case of annotation theories. We then provide a new second-order quantifier elimination method for stratified theories, which is successful in the considered cases. The new result subsumes many other results, including those of [2, 28, 21].

Keywords: argumentation theory, labeled graphs, annotations, semantics of logic programs, second-order quantifier elimination.

1. Introduction to Annotation Theories and Related Problems

In the current paper we consider theories with vocabulary containing a number of binary and unary relation symbols. Binary relation symbols represent labeled edges of a graph and unary relations represent unique annotations of the graph's nodes. We call such theories *annotation theories*. They can be used in many applications, including the formalization of argumentation, approximate reasoning, semantics of logic programs, graph coloring, etc. Given an annotation theory, we address the following problems, where only finite models are considered:

- *satisfiability*: given a finite set D, is there a model for the theory with D as the underlying domain?

Special Issue: New Ideas in Argumentation Theory
Edited by **Dov M. Gabbay and Leendert van der Torre**

Studia Logica (2009) 93: 147–180
DOI: 10.1007/s11225-009-9220-3
© Springer 2009

- *querying problem*: given a graph with edges represented by binary relations, are there annotations satisfying the theory?
- *specification of preferred models*: how to use circumscription on chosen annotations to specify preferred models and how to reduce second-order circumscription formulas to first-order or fixpoint logic?
- *model checking problem*: given a relational structure with edges and annotations, does it satisfy the circumscribed annotation theory?

These problems are of second-order nature. The methodology we use depends then on specifying them in the second-order logic and then on eliminating second-order quantifiers.[1] An application of quantifier elimination methods applied in this paper, if successful, can result in:

- a formula of the first-order logic, validity of which (over finite models) is in LOGSPACE and therefore also in PTIME. Here we apply the DLS algorithm of [11], based on the Ackermann lemma (see Lemma 5.3); also the SCAN algorithm of [14] can be used here;
- a formula of the fixpoint logic, validity of which (over finite models) is in PTIME. Here one can apply our Theorem 5.10, which substantially extends existing direct methods, including the theorem of Nonnengart and Szałas [28], theorem of Kachniarz and Szałas [21] and Ackermann's lemma [2] (quoted as Theorems 5.2, Theorem 5.1 and Lemma 5.3 in Section 5).

Therefore second-order quantifier elimination, when successful, gives us also tractable algorithms for problems we address.

The main contribution of this paper depends on providing a general second-order quantifier elimination result (Theorem 5.10), results for reducing circumscribed theories (Lemmas 6.4, 6.5), including annotation theories as well as results related to specific theories: Phan's argumentation theory[2] [30] (Section 6.3), a theory related to approximate reasoning (Section 6.4) and a theory formalizing a semantics of logic programs with negation, derived from considerations of [36], closely related to stable models [17, 1] (Section 6.5).

Complexity results are valid in the case of finite models only. On the other hand, the provided quantifier elimination techniques are not restricted

---

[1]For an overview of known second-order quantifier elimination techniques see [15].

[2]Phan's argumentation theory is perhaps better known as "Dung's argumentation theory" and frequently cited using "Dung" as the author's family name. In fact, "Dung" is the first name and the family name is "Phan".

to finite models nor to annotation theories. To make quantifier elimination possible, we define the notion of stratification of the considered theories, generalizing the corresponding idea known from logic programming. Theorem 5.10 works for any stratified theories. It is worth emphasizing that no existing up to now direct second-order quantifier elimination methods is successful for annotation theories (see the discussion provided in Section 6.1). Also resolution-based methods fail when axioms are recursive, which takes place in theories considered in this paper.

The paper is structured as follows. In Section 2 we recall some well-known notions used in the paper. Section 3 introduces the concept of annotation theories and illustrates them by Phan's argumentation theory, a theory related to approximate reasoning as well as by a formalization of semantics of logic programs. In Section 4 we show that the satisfiability problem and the querying problem are NPTime-complete by noticing that graph colorability can be formulated as an annotation theory. We also show that model checking problem for circumscribed annotation theories is co-NPTime complete. Then, in Section 5, we provide second-order quantifier elimination results. Section 6 is devoted to elimination of second-order quantifiers in the context of circumscribed annotation theories. Finally, Section 7 concludes the paper.

2. Preliminaries

2.1. Basics

Through the paper we use the language of classical first- and second-order logic without function symbols.[3] We assume the standard first- and second-order semantics.

By a *literal* we understand a first-order formula of the form $R(\ldots)$ or $\neg R(\ldots)$, where R is a relation symbol. A formula A is in the *negation normal form* if it uses no propositional connectives other than $\neg, \vee, \wedge$ and the negation sign $\neg$ does not occur in A outside of literals. It is well-known that every classical first- and second- order formula can equivalently be transformed into a formula in negation normal form.

Let $A(R)$ be a formula and $B(R)$ be a formula in the negation normal form equivalent to $A(R)$. An *occurrence of* R *is positive in* $A(R)$, if the corresponding occurrence of R in $B(R)$ is not preceded by $\neg$. An *occurrence of* R *is negative in* $A(R)$, if the corresponding occurrence of R in $B(R)$ is of the form $\neg R$. Formula $A(R)$ is *positive w.r.t.* R if all occurrences of R in A

[3] We do not use function symbols as they do not appear in theories we deal with. Also, this allows us to use deductive databases [1, 13, 20] as a computational machinery.

are positive. It is *negative w.r.t.* R if all occurrences of R in A are negative.[4]

Writing $A(\bar{X}, \bar{y})$, we mean that A contains variables $\bar{X}$ and $\bar{y}$, but we do not exclude other arguments.

If $M = \langle D, \bar{R} \rangle$ is a relational structure and v is a valuation of variables in D then we write $M, v \models A$ to denote that A is true in M under the valuation v. We write $M \models A$ to denote that A is true in M under all valuations of free variables occurring in A. If $\bar{R}$ is empty then we write $D \models A$ rather than $\langle D \rangle \models A$.

By $A(a, \ldots)^{\lambda\bar{x}_1.e_1,\ldots,\lambda\bar{x}_k.e_k}_{\lambda\bar{y}_1.f_1,\ldots,\lambda\bar{y}_k.f_k}$ we understand the expression obtained from A in such a way that for any $1 \leq i \leq k$, all occurrences of e_i of the form $e_i(\bar{a})$ are replaced by $f_i(\bar{y}_i)$, where $\bar{y}_i$ itself is replaced by $\bar{a}$. When λ-expression is a relation symbol applied to some arguments, say $P(\bar{z})$, then we write $P(\bar{z})$ rather than $\lambda\bar{z}.P(\bar{z})$. For example,

$$\big(P(s) \vee R(t)\big)^{P(x),\qquad\qquad\qquad\quad R(y)}_{\lambda u.(Q(a,u) \wedge Q(u,b)),\ S(z,z)} = \big(Q(a,s) \wedge Q(s,b)\big) \vee S(t,t).$$

2.2. Circumscription

In what follows we also use circumscription [27, 26, 11], which is basically a technique for minimizing chosen predicates with some other allowed to vary and all other fixed. Let us now formally define this concept.

Definition 2.1. Let $\bar{P} = \langle P_1, \ldots, P_k \rangle, \bar{S} = \langle S_1, \ldots, S_m \rangle$ be disjoint tuples of relation symbols, and let $T(\bar{P}, \bar{S})$ be a first-order formula.

The *second-order circumscription* of $\bar{P}$ in $T(\bar{P}, \bar{S})$ with variable $\bar{S}$, written $Circ_{\downarrow}(T; \bar{P}; \bar{S})$, is the second-order formula

$$\begin{aligned} & T(\bar{P}, \bar{S}) \wedge \\ & \forall \bar{X} \forall \bar{Y} \Big\{ \big[T(\bar{P}, \bar{S})^{\bar{P},\bar{S}}_{\bar{X},\bar{Y}} \wedge \bigwedge_{i=1}^{k} \forall \bar{x}_i [X_i(\bar{x}_i) \to P_i(\bar{x}_i)] \, \big] \to \\ & \qquad\qquad \bigwedge_{i=1}^{k} \forall \bar{x}_i [P_i(\bar{x}_i) \to X_i(\bar{x}_i)] \Big\}, \end{aligned} \tag{1}$$

where $\bar{X}$ and $\bar{Y}$ are tuples of relational variables of the same arities as those in $\bar{P}$ and $\bar{S}$, respectively. ■

We will also need a dual form of circumscription, where some predicates are maximized rather than minimized.

[4] Observe that formula in which R does not occur is both positive and negative w.r.t. R.

Definition 2.2. Let $\bar{P}, \bar{S}$ and $T(\bar{P}, \bar{S})$ be as in Definition 2.1. The *dual second-order circumscription* of $\bar{P}$ in $T(\bar{P}, \bar{S})$ with variable $\bar{S}$, written $Circ^{\uparrow}(T; \bar{P}; \bar{S})$, is the second-order formula

$$T(\bar{P}, \bar{S}) \wedge$$
$$\forall \bar{X} \forall \bar{Y} \Big\{ \big[T(\bar{P}, \bar{S})^{\bar{P}, \bar{S}}_{\bar{X}, \bar{Y}} \wedge \bigwedge_{i=1}^{k} \forall \bar{x}_i [P_i(\bar{x}_i) \rightarrow X_i(\bar{x}_i)] \, \big] \rightarrow \tag{2}$$
$$\bigwedge_{i=1}^{k} \forall \bar{x}_i [X_i(\bar{x}_i) \rightarrow P_i(\bar{x}_i)] \Big\}. \qquad \blacksquare$$

The class of all models of a theory T will be denoted by $mod(T)$. We assume that the class consists of relational structures of the form $M = \langle D^M, \langle R_i^M \rangle_{i \in I} \rangle$.

The semantics of circumscription is based on the concept of sub-models (see [25, 26]) defined as follows.

Definition 2.3. Let $\bar{P}$, $\bar{S}$ and $T(\bar{P}, \bar{S})$ be as in Definitions 2.1 and 2.2. Let M and N be models of T. We say that M is a $(\bar{P}, \bar{S})$-*submodel* of N, written $M \leq^{(\bar{P}, \bar{S})} N$, iff

1. $D^M = D^N$
2. $R^M = R^N$, for any relation symbol R not in $\bar{P} \cup \bar{S}$
3. $R^M \subseteq R^N$, for any relation symbol R in $\bar{P}$. ■

We write $M <^{(\bar{P}, \bar{S})} N$ when $M \leq^{(\bar{P}, \bar{S})} N$, but not $N \leq^{(\bar{P}, \bar{S})} M$. A model M of T is $(\bar{P}, \bar{S})$-*minimal* iff T has no model N such that $N <^{(\bar{P}, \bar{S})} M$. It is $(\bar{P}, \bar{S})$-*maximal* iff T has no model N such that $M <^{(\bar{P}, \bar{S})} N$. We also write $mod_{\downarrow}^{(\bar{P}, \bar{S})}(T)$ to denote the class of all $(\bar{P}, \bar{S})$-minimal models of T and $mod^{\uparrow (\bar{P}, \bar{S})}(T)$ to denote the class of all $(\bar{P}, \bar{S})$-maximal models of T. The semantics of circumscription is now given by

$$mod\big(Circ_{\downarrow}(T; \bar{P}; \bar{S})\big) = mod_{\downarrow}^{(\bar{P}, \bar{S})}(T) \tag{3}$$
$$mod\big(Circ^{\uparrow}(T; \bar{P}; \bar{S})\big) = mod^{\uparrow (\bar{P}, \bar{S})}(T). \tag{4}$$

2.3. Simultaneous Fixpoints

We will also use the notion of simultaneous fixpoints (see, e.g., [1, 13, 20] for a detailed exposition of the theory of fixpoints and their applications as database queries).

Let $\bar{Q} = \langle Q_1, \ldots, Q_k \rangle$ be a tuple of relation symbols and $A_i(\bar{Q}, \bar{x}_i, \bar{y}_i)$, for $i = 1, \ldots, k$, be classical first-order formulas, where

- $\bar{x}_i$ and $\bar{y}_i$ are all free first-order variables of A_i
- the number of variables in $\bar{x}$ is k_i
- none of the x's is among the y's
- for $i = 1, \ldots, k$, Q_i is a k_i-argument relation symbol, whose all occurrences in $A_1, \ldots, A_k$ are positive.

DEFINITION 2.4. Under the above assumptions, the expression

$$\text{SLFP}\,[Q_1(\bar{x}_1) \equiv A_1(\bar{Q},\, \bar{x}_1, \bar{y}_1), \ldots, Q_k(\bar{x}_k) \equiv A_k(\bar{Q},\, \bar{x}_k, \bar{y}_k)] \tag{5}$$

is called the *simultaneous least fixpoint of* $A_1, \ldots, A_k$, and the expression

$$\text{SGFP}\,[Q_1(\bar{x}_1) \equiv A_1(\bar{Q},\, \bar{x}_1, \bar{y}_1), \ldots, Q_k(\bar{x}_k) \equiv A_k(\bar{Q},\, \bar{x}_k, \bar{y}_k)] \tag{6}$$

is called the *simultaneous greatest fixpoint of* $A_1, \ldots, A_k$. ■

Note that both SLFP and SGFP represent k-tuples of relations.

In the rest of the paper we often abbreviate the formula in the scope of SLFP in (5) (and of SGFP in (6)) by $\bar{Q} \equiv \bar{A}$, formula (5) by $\text{SLFP}\,[\bar{Q} \equiv \bar{A}]$, and formula (6) by $\text{SGFP}\,[\bar{Q} \equiv \bar{A}]$.

DEFINITION 2.5. The semantics of $\text{SLFP}\,[\bar{Q} \equiv \bar{A}]$ is given by (the unique) tuple of relations $\bar{Q}$ satisfying $Circ_{\downarrow}(\bar{Q} \equiv \bar{A}; \bar{Q}; \emptyset)$, and the semantics of $\text{SGFP}\,[\bar{Q} \equiv \bar{A}]$ is given by (the unique) tuple of relations $\bar{Q}$ satisfying $Circ^{\uparrow}(\bar{Q} \equiv \bar{A}; \bar{Q}; \emptyset)$. ■

If $k = 1$ in formulas (5) and (6), then the simultaneous fixpoints reduce to the standard fixpoints. In such cases we write $\text{LFP}\big[Q(\bar{x}) \equiv A(Q, \bar{x}, \bar{y})\big]$ to stand for $\text{SLFP}\,[Q(\bar{x}) \equiv A(Q, \bar{x}, \bar{y})]$ and $\text{GFP}\big[Q(\bar{x}) \equiv A(Q, \bar{x}, \bar{y})\big]$ to stand for $\text{SGFP}\,[Q(\bar{x} \equiv A(Q, \bar{x}, \bar{y})]$.

3. Annotation Theories

3.1. Definition

We consider a directed graph or network, seen as a model of binary relations $\bar{R} = \langle R_i \rangle_{i=1,\ldots,m}$ $(m \geq 1)$ on a finite set D. Relations in $\bar{R}$ represent *edges* of various kinds. D is called the *set of nodes*. If $m = 1$ then we omit the subscript and write R rather than R_1. We allow annotations on the nodes. The annotation is represented by unary predicates $\bar{Q} = \langle Q_j \rangle_{j=1,\ldots,n}$ $(n > 1)$, where $Q_i(x)$ holds if node x is annotated by Q_i. Δ^{mn}-theories allow us to set requirements on annotations.

DEFINITION 3.1. Let $m \geq 1$ and $n > 1$. By an (m, n)-*annotation theory*, referred to as Δ^{mn}-theory, we understand any finite first-order theory $\Delta^{mn}(\bar{R}, \bar{Q})$ over the signature containing binary relation symbols $\bar{R} = \langle R_i \rangle_{i=1,\ldots,m}$ and unary relation symbols $\bar{Q} = \langle Q_j \rangle_{j=1,\ldots,n}$, where we assume that *annotations* in $\bar{Q}$ are unique, i.e., each Δ^{mn}-theory, in addition to specific axioms, contains also axioms:

$$\sigma(n) \stackrel{\text{def}}{\equiv} \forall x \Big[\bigvee_{1 \leq i \leq n} Q_i(x) \Big] \text{ and } \pi(n) \stackrel{\text{def}}{\equiv} \bigwedge_{1 \leq i \neq j \leq n} \forall x \, [\neg Q_i(x) \vee \neg Q_j(x)] \,. \quad \blacksquare$$

Observe that for each $1 \leq i \leq n$,

$$\sigma(n) \equiv \forall x \Big[\Big(\bigwedge_{1 \leq k \neq i \leq n} \neg Q_k(x) \Big) \rightarrow Q_i(x) \Big]. \tag{7}$$

The above simple observation is useful in specifying preferred models. In particular, it shows that annotations are strongly related to each other and minimizing/maximizing some of them usually requires varying all others. In is also useful in eliminating second-order quantifiers from circumscription axioms.

In the rest of the paper we only allow a finite number of axioms. Any finite set of axioms can be represented by a single formula being the conjunction of its members. This restriction allows us to encode the considered problems by second-order formulas.

3.2. Example: Phan's Argumentation Theory

In argumentation theory the following Δ^{13}-theory, $\mathcal{A}(R, Q_1, Q_2, Q_3)$, is often considered (see Phan [30]):

$$\sigma(3) \wedge \pi(3) \tag{8}$$

$$\forall x[\forall y[R(y, x) \rightarrow Q_1(y)] \rightarrow Q_2(x)] \tag{9}$$

$$\forall x[\exists y[R(y, x) \wedge Q_2(y)] \rightarrow Q_1(x)] \tag{10}$$

$$\forall x[(\forall y[R(y, x) \rightarrow (Q_1(y) \vee Q_3(y))] \wedge \exists y[R(y, x) \wedge Q_3(y)]) \rightarrow Q_3(x)]. \tag{11}$$

The intended meaning of R, Q_1, Q_2, Q_3 is:

- elements of underlying models are arguments
- $R(x, y)$ means that argument x attacks argument y
- $Q_1(x)$ means that argument x is not-active/refuted, $Q_2(x)$ means that argument x is active and $Q_3(x)$ means that argument x is undecided.

Here one looks for minimal Q_2, maximal Q_2 or minimal Q_3, where in each case all relations, other than the minimized/maximized one, are allowed to vary (see [7]). That is, we respectively consider $Circ_{\downarrow}(\mathcal{A}; Q_2; Q_1, Q_3)$ (see Section 6.3.1), $Circ^{\uparrow}(\mathcal{A}; Q_2; Q_1, Q_3)$ (see Section 6.3.2) and $Circ_{\downarrow}(\mathcal{A}; Q_3; Q_1, Q_2)$ (see Section 6.3.3).

3.3. Example: Theory Related to Approximate Reasoning

In approximate reasoning one often uses a generalization of rough sets and relations [29], which depends on allowing arbitrary similarity relations, while in the rough set theory only equivalence relations are considered. Such generalized approximate reasoning has been shown useful in many application areas requiring the use of approximate knowledge structures (see, e.g., [9, 12]).

In order to formalize the fact that similarities should preserve properties of objects, we use the following Δ^{13}-theory, $\mathcal{R}(R, Q_1, Q_2, Q_3)$:

$$\sigma(3) \wedge \pi(3) \tag{12}$$

$$\forall x \forall y[(R(x, y) \wedge Q_1(x)) \rightarrow Q_1(y)] \tag{13}$$

$$\forall x \forall y[(R(x, y) \wedge Q_2(x)) \rightarrow Q_2(y)] \tag{14}$$

$$\forall x \forall y[(R(x, y) \wedge Q_3(x)) \rightarrow Q_3(y)]. \tag{15}$$

The intended meaning of R, Q_1, Q_2, Q_3 is:

- elements of underlying models are objects
- $R(x, y)$ means that object x is similar to object y
- $Q_1(x)$ means that x satisfies a given property, $Q_2(x)$ means that x does not satisfy the property and $Q_3(x)$ means that it is unknown whether x satisfies the property.

Here one often looks for simultaneous minimization of Q_1 and Q_3 with Q_2 allowed to vary, i.e., we consider $Circ_{\downarrow}(\mathcal{R}; Q_1, Q_3; Q_2)$ (see Section 6.4). This policy corresponds to the closed world assumption, where it is assumed that all positive facts are specified and all other facts should be considered false.

3.4. Example: Formalizing Semantics for Logic Programs with Negation

In order to provide a semantics for logic programs with negation allowed in the bodies of rules, we use a Δ^{23}-theory, which we derive from considerations provided in [36].

Let P be a propositional logic program with negation as failure. Consider clauses of the form:

$$q :- \bigwedge_{i \in I} a_i, \bigwedge_{j \in J} \neg b_j, \tag{16}$$

where I, J are finite sets of indices.

We regard the atoms $q, \{a_i\}_{i \in I}, \{b_j\}_{j \in J}$ as elements in a classical model. The domain od the model, D_P, consists of all the atoms appearing in a given logic program.

We consider two binary relations R_+, R_- such that:

$$\begin{array}{l} \text{– for clauses of the form (16) we require that} \\ \quad R_+(a_i, q) \text{ and } R_-(b_j, q) \text{ hold } (i \in I, j \in J) \\ \text{– for all other cases } R+, R_- \text{ are false.} \end{array} \tag{17}$$

Consider a logic program P satisfying the condition that

for any literal q, P contains at most one clause with q as its head. (18)

With such a program we can associate a model $\langle D_P, R_+, R_- \rangle$, where D_P is the set of atoms in P and R_+, R_- are defined as above. Conversely, with any finite domain model $\langle D_P, R_+, R_- \rangle$ with two binary relations R_+, R_- we can associate a logic program

$$y :- \bigwedge_{\{x | R_+(x,y)\}} x, \quad \bigwedge_{\{x | R_-(x,y)\}} \neg x. \tag{19}$$

Assume now that a program violates (18), i.e., contains several clauses with the same head,

$$q^k :- \bigwedge_{i \in I_k} a^k_{i_k}, \bigwedge_{j \in J_k} \neg b^k_{j_k}, \tag{20}$$

where $k \leq r$, for some $r \geq 1$.

We add r new propositions, $q^k_1, \ldots, q^k_r$ and we replace clauses $C_1, \ldots, C_r$ by clauses:

$$q^k_i :- \bigwedge_{i \in I_k} a^k_{i_k}, \bigwedge_{j \in J_k} \neg b^k_{j_k}, \qquad q^* :- \bigwedge_{1 \leq k \leq r} \neg q^k_i, \qquad q :- \neg q^*. \tag{21}$$

Now,

- q succeeds if q^* fails

- q loops if q^* loops
- q^* fails if at least one of $\neg q_i^k$ fails, i.e., if at least one of q_i^k succeeds, i.e., if at least one of bodies $\left[\bigwedge_{i\in I_k} a_{i_k}^k, \bigwedge_{j\in J_k} \neg b_{j_k}^k\right]$ succeeds.

Therefore, given a program P, one can write a program P' satisfying the assumption (18) as to unique heads.

We now formalize the semantics of logic programs with negation by a Δ^{23}-theory $\mathcal{L}(R_+, R_-, Q_1, Q_2, Q_3)$, where

- elements of underlying models are atoms of a given logic program
- R_+, R_- are explained in (17)
- $Q_1(x)$ means that the computation of x fails
- $Q_2(x)$ means that the computation of x succeeds
- $Q_3(x)$ means that the computation of x loops.

The theory $\mathcal{L}$ consists of the following axioms:

$$\sigma(3) \wedge \pi(3) \tag{22}$$

$$\forall x\big[\big(\exists y[R_+(y,x) \wedge Q_1(y)] \vee \exists y[R_-(y,x) \wedge Q_2(y)]\big) \rightarrow Q_1(x)\big] \tag{23}$$

$$\forall x\big[\big(\forall y[R_+(y,x) \rightarrow Q_2(y)] \wedge \forall y[R_-(y,x) \rightarrow Q_1(y)]\big) \rightarrow Q_2(x)\big] \tag{24}$$

$$\begin{aligned}\forall x\big[\big(\forall y[R_+(y,x) \rightarrow (Q_2(y) \vee Q_3(y))] \wedge \forall y[R_-(y,x) \rightarrow (Q_1(y) \vee Q_3(y))]\\ \wedge \exists y[(R_-(y,x) \vee R_+(y,x)) \wedge Q_3(y)]\big) \rightarrow Q_3(x)\big].\end{aligned} \tag{25}$$

We look for models with minimal Q_2, where Q_1 and Q_3 are allowed to vary, which is expressed by $Circ_{\downarrow}(\mathcal{L}; Q_2; Q_1, Q_3)$ (see Section 6.5).

3.5. Some other Applications

Observe that roles considered in description logics [3] can be represented as graphs whose edges correspond to roles. Annotations become necessary whenever one needs to uniquely identify nodes. In [19], Horrocks and Sattler discuss the need for annotations:

> "realistic ontologies typically contain references to named individuals within class descriptions. E.g., *Italians* might be described as persons who are citizens of *Italy*, where *Italy* is a named individual."

Yet another motivation for annotation theories is related to *nominals* which are a prominent feature of hybrid logics and their immediate ancestors, called modal logics with names [4, 18]. Consider, for example temporal reasoning. Once we refer to particular time points, e.g., by using dates ("it is going to be a board meeting on November 15th at 13:15"), we deal with unique annotations.

4. Complexity Results

Consider first satisfiability checking and the querying problem.

Let $\mathcal{T}(R_1, \ldots, R_m, Q_1, \ldots, Q_n)$ be an arbitrary Δ^{mn}-theory. The *satisfiability problem* for $\mathcal{T}$ over a set of nodes D is expressed by

$$D \models \exists R_1 \ldots \exists R_m \exists Q_1 \ldots \exists Q_n [\mathcal{T}(R_1, \ldots, R_m, Q_1, \ldots, Q_n)]. \tag{26}$$

The *querying problem* assumes that a structure $\mathcal{M} = \langle D, R_1, \ldots, R_m \rangle$ is given and one asks whether

$$\mathcal{M} \models \exists Q_1 \ldots \exists Q_n [\mathcal{T}(R_1, \ldots, R_m, Q_1, \ldots, Q_n)]. \tag{27}$$

We start with the querying problem.

THEOREM 4.1. *For $m \geq 1$ and $n \geq 3$, the querying problem for annotation theories is* NPTIME*-complete.*

PROOF. Let $\mathcal{T}(R_1, \ldots, R_m, Q_1, \ldots, Q_n)$ be an arbitrary Δ^{mn}-theory, where m, n are fixed natural numbers.

Given a finite model $M = \langle D, R_1, \ldots, R_m, Q_1, \ldots, Q_n \rangle$, one can check whether

$$M \models \mathcal{T}(R_1, \ldots, R_m, Q_1, \ldots, Q_n)$$

deterministically in time polynomial in the size of D (see [1, 13, 20]). So, given $\langle D, R_1, \ldots, R_m \rangle$, a nondeterministic polynomial time algorithm for the querying problem depends on guessing $Q_1, \ldots, Q_n$ and then accepting the result when the obtained model satisfies $\mathcal{T}(R_1, \ldots, R_m, Q_1, \ldots, Q_n)$. Of course, guessing $Q_1, \ldots, Q_n$ can be done in time linear in the size of D (recall that n is fixed). Thus the querying problem is in NPTIME.

To show NPTIME-completeness we consider the following Δ^{1n}-theory, denoted by $\mathcal{C}(R, Q_1, \ldots, Q_n)$:

$$\sigma(n) \wedge \pi(n) \wedge \bigwedge_{1 \leq i \leq n} \forall x \forall y \big[R(x, y) \rightarrow (\neg Q_i(x) \vee \neg Q_i(y)) \big]. \tag{28}$$

The theory $\mathcal{C}$ expresses the fact that a graph with edges represented by R can be colored using n colors $Q_1, Q_2, \ldots, Q_n$.

If only $\langle D, R\rangle$ is given, then checking n-colorability is expressed by:

$$\langle D, R\rangle \models \exists Q_1 \ldots \exists Q_n \big[(28)\big]. \tag{29}$$

It is well-known that this problem is NPTime-complete already for $n = 3$. ■

By a similar proof we have the following theorem.

Theorem 4.2. *For $m \geq 1$ and $n \geq 3$, the satisfiability problem for annotation theories is* NPTime-*complete.* ■

The *model checking problem* for a Δ^{mn}-theory $\mathcal{T}$ assumes that a structure $M = \langle D, R_1, \ldots, R_m, Q_1, \ldots, Q_n\rangle$ is given and one asks whether

$$\mathcal{M} \models Circ_{\downarrow}(\mathcal{T}, \bar{Q}', \bar{Q}'') \quad \text{(respectively, } \mathcal{M} \models Circ^{\uparrow}(\mathcal{T}, \bar{Q}', \bar{Q}'')\text{)}, \tag{30}$$

where $\bar{Q}', \bar{Q}''$ are chosen from $Q_1, \ldots, Q_n$.

Let us now show that model checking for circumscribed annotation theories is a co-NPTime-complete problem. We adapt the proof given by Kolaitis and Papadimitriou (see pages 11-12 of [23]) for co-NPTime-completeness of model checking for circumscription (Theorem 6 of [23]). We cannot use this result directly, as we want to show that co-NPTime-completeness can be proved for circumscription on annotations, while the result of [23] applies circumscription to edges. For other results concerning the complexity of circumscription see [5, 22] and references there.

We need the following definition.

Definition 4.3. We call a undirected graph *cubic* if all its nodes have degree three. A *circuit* of a graph is a closed path without repetitions of edges. A circuit is *long* if it contains at least twelve nodes. A graph is *simple* if it is a disjoint union of long circuits.

We say that graph $G = \langle N, E\rangle$ is a *subgraph* of graph $G' = \langle N', E'\rangle$ if $N = N'$ and $E \subseteq E'$. G is a *proper subgraph* of G' if it is a subgraph of G' and $E \neq E'$. ■

Of course, cubicity of a graph is a first-order property and can be expressed by a first-order formula $\rho(E)$ on edges.

Observe that simple graphs have all degrees two and there are no circuits of length eleven or less in them. Therefore, simplicity is also a first-order property and can ce expressed by a first-order formula $\eta(E)$.

We have the following lemma.

LEMMA 4.4 (Lemma 2 of Kolaitis and Papadimitriou [23]). *It is* NPTIME-*complete to check whether a cubic connected graph has a simple subgraph.* ■

Now we are in position to prove the announced complexity result.

THEOREM 4.5. *There is an annotation theory and circumscriptive policy on annotations whose model checking is* co-NPTIME-*complete.*

PROOF. Consider the following Δ^{12}-theory, denoted by $\mathcal{B}$ with axioms:

$$\sigma(2) \wedge \pi(2)$$
$$\rho(E) \vee \eta(E)$$
$$\eta(E'), \text{ where } E'(x,y) \stackrel{\text{def}}{\equiv} \big(Q_1(x) \wedge Q_1(y)\big).$$

Theory $\mathcal{B}$ states that graph $G = \langle N, E\rangle$ is either cubic or simple and that Q_1 "selects" a simple subgraph from G, while Q_2 annotates nodes outside of the selected subgraph. Consider $Circ_{\downarrow}(\mathcal{B}; Q_1; Q_2)$. It additionally says that there is no proper subgraph of G which is cubic or simple.

Consider a relational structure $M = \langle N, E, Q_1, Q_2\rangle$ with $\langle N, E\rangle$ being a cubic graph. The question is whether

$$M \models Circ_{\downarrow}(\mathcal{B}; Q_1; Q_2). \tag{31}$$

As noted above, (31) holds when there is no proper subgraph of G which is cubic or simple. No proper subgraph of a cubic graph can be cubic, so (31) holds when no simple subgraph of G exists. By Lemma 4.4, checking existence of a simple subgraph of a given graph is NPTIME-complete. Therefore checking whether (31) holds is an co-NPTIME complete problem. ■

REMARK 4.1. Observe that, in the light of Theorems 4.1, 4.2 and 4.5, the elimination of all second-order quantifiers from formulas (26), (27) and (30) is, in general, problematic. A successful elimination from formulas considered in proofs of Theorems 4.1, 4.2 and 4.5, would imply that PTIME =NPTIME.

The presence of both $\sigma(n)$ and $\pi(n)$ in Δ^{mn}-theories suggests that algorithmic quantifier elimination techniques depending on the syntactic shape of formulas require further simplifications of circumscribed formulas or certain syntactic restrictions as to the remaining axioms. To see the intuition, assume that the underlying database consisting of objects and edges is given. One can now translate a given annotation theory into a propositional theory with propositional variables corresponding to annotations. The resulting propositional theory is non-Schaefer [31]. Due to the Schaefer's dichotomy theorem, the satisfiability problem for theories containing axioms of that shape is NPTIME-complete. We can then strongly expect that second-order quantifier elimination methods depending on the shape of formulas cannot generally be applied here.

For a dichotomy theorem directly concerning circumscribed theories, also supporting this intuition, see [22]. ■

5. Second-Order Quantifier Elimination

5.1. Simultaneous Elimination Theorem

Let us start with a theorem allowing one to eliminate a number of second-order existential quantifiers at the same time. Theorem 5.1 is a special case of Theorem 5.10, but we formulate it separately for two reasons. First, it is useful in some applications which do not require the full strength of Theorem 5.10. Second, it considerably simplifies the proof of Theorem 5.10.

THEOREM 5.1 (Kachniarz and Szałas [21]). *Let $\bar{X} = X_1, \ldots, X_k$ be distinct relation variables and $C(\bar{X})$, $A_i(\bar{X}, \bar{x}_1, \ldots, \bar{x}_k, \bar{z})$ $(1 \leq i \leq k)$ be classical first-order formulas, where the number of distinct variables in $\bar{x}_i$ is equal to the arity of X_i and $A_i(\bar{X}, \ldots)$ is positive w.r.t. $\bar{X}$. Then:*

- *if $C(\bar{X})$ is negative w.r.t. $X_1, \ldots, X_k$ then*

$$\exists X_1 \ldots \exists X_k \left\{ \bigwedge_{1 \leq i \leq k} \forall \bar{x}_i [A_i(\bar{X}, \bar{x}_1, \ldots, \bar{x}_k, \bar{z}) \to X_i(\bar{x}_i)] \wedge \ C(\bar{X}) \right\} \tag{32}$$
$$\equiv$$
$$C(\bar{X})^{X_1(\bar{x}_1), \ldots, X_k(\bar{x}_k)}_{\lambda \bar{x}_1, \ldots, \bar{x}_k \mathrm{SLFP}\, [X_1(\bar{x}_1) \equiv A_1(\bar{X}, \ldots), \ldots, X_k(\bar{x}_k) \equiv A_k(\bar{X}, \ldots)]}.$$

- *if $C(\bar{X})$ is positive w.r.t. $X_1, \ldots, X_k$ then*

$$\exists X_1 \ldots \exists X_k \left\{ \bigwedge_{1 \leq i \leq k} \forall \bar{x}_i [X_i(\bar{x}_i) \to A_i(\bar{X}, \bar{x}_1, \ldots, \bar{x}_k, \bar{z})] \wedge \ C(\bar{X}) \right\} \tag{33}$$
$$\equiv$$
$$C(\bar{X})^{X_1(\bar{x}_1), \ldots, X_k(\bar{x}_k)}_{\lambda \bar{x}_1, \ldots, \bar{x}_k \mathrm{SGFP}\, [X_1(\bar{x}_1) \equiv A_1(\bar{X}, \ldots), \ldots, X_k(\bar{x}_k) \equiv A_k(\bar{X}, \ldots)]}.$$

PROOF. We prove (33). The proof of (32) is analogous. Let M be a relational structure and v be a valuation of variables.

($\rightarrow$) Assume that

$$M, v \models \exists X_1 \ldots \exists X_k \left\{ \bigwedge_{1 \le i \le k} \forall \bar{x}_i [X_i(\bar{x}_i) \rightarrow A_i(\bar{X}, \bar{x}_1, \ldots, \bar{x}_k, \bar{z})] \land \ C(\bar{X}) \right\}. \quad (34)$$

Therefore, there exists a valuation V assigning to $X_1, \ldots, X_k$ relations over the domain of M such that

$$M, v, V \models \bigwedge_{1 \le i \le k} \forall \bar{x}_i [X_i(\bar{x}_i) \rightarrow A_i(\bar{X}, \bar{x}_1, \ldots, \bar{x}_k, \bar{z})] \land \ C(\bar{X}).$$

Note that the greatest (w.r.t. $\rightarrow$) $\bar{X}$ satisfying

$$\bigwedge_{1 \le i \le k} \forall \bar{x}_i [X_i(\bar{x}_i) \rightarrow A_i(\bar{X}, \bar{x}_1, \ldots, \bar{x}_k, \bar{z})]$$

is the greatest $\bar{X}$ satisfying $\bigwedge_{1 \le i \le k} \forall \bar{x}_i [X_i(\bar{x}_i) \equiv A_i(\bar{X}, \bar{x}_1, \ldots, \bar{x}_k, \bar{z})]$, which by Definition 2.5 and equation (4), is given by

$$\text{SGFP}\, [X_1(\bar{x}_1) \equiv A_1(\bar{X}, \ldots), \ldots, X_k(\bar{x}_k) \equiv A_k(\bar{X}, \ldots)].$$

Since $C(\bar{X})$ is positive in $X_1, \ldots, X_k$, it is also monotone in $X_1, \ldots, X_k$. Therefore we have that

$$M, v, V \models C(\bar{X})^{X_1(\bar{x}_1), \ldots, X_k(\bar{x}_k)}_{\lambda \bar{x}_1, \ldots, \bar{x}_k \text{SGFP}\, [X_1(\bar{x}_1) \equiv A_1(\bar{X}, \ldots), \ldots, X_k(\bar{x}_k) \equiv A_k(\bar{X}, \ldots)]}. \quad (35)$$

Relational variables $X_1, \ldots, X_k$ in (35) are bound by the simultaneous fixpoint operator SGFP, so V in (35) becomes redundant and we obtain

$$M, v \models C(\bar{X})^{X_1(\bar{x}_1), \ldots, X_k(\bar{x}_k)}_{\lambda \bar{x}_1, \ldots, \bar{x}_k \text{SGFP}\, [X_1(\bar{x}_1) \equiv A_1(\bar{X}, \ldots), \ldots, X_k(\bar{x}_k) \equiv A_k(\bar{X}, \ldots)]}. \quad (36)$$

($\leftarrow$) Assume now (36). It is easy to observe that $X_1, \ldots, X_k$ defined by respective coordinates of SGFP $[X_1(\bar{x}_1) \equiv A_1(\bar{X}, \ldots), \ldots, X_k(\bar{x}_k) \equiv A_k(\bar{X}, \ldots)]$ satisfy $M, v \models \bigwedge_{1 \le i \le k} \forall \bar{x}_i [X_i(\bar{x}_i) \rightarrow A_i(\bar{X}, \bar{x}_1, \ldots, \bar{x}_k, \bar{z})]$.

By (36), such $X_1, \ldots, X_k$ satisfy $M, v \models C(\bar{X})$. We have then indicated $X_1, \ldots, X_k$ for which $M, v \models \bigwedge_{1 \le i \le k} \forall \bar{x}_i [X_i(\bar{x}_i) \rightarrow A_i(\bar{X}, \bar{x}_1, \ldots, \bar{x}_k, \bar{z})] \land C(\bar{X})$. Therefore (34) holds, too. ■

5.2. Some Consequences of the Elimination Theorem

We have two corollaries of Theorem 5.1, which we also use in further calculations.

The following theorem is a particular case of Theorem 5.1 when we eliminate a single relation variable.

THEOREM 5.2 (Nonnengart and Szałas [28]). *Let X be a relation variable and $A(X,\bar{x},\bar{z})$, $C(X)$ be a classical first-order formula, where the number of distinct variables in $\bar{x}$ is equal to the arity of X, $A(X,\bar{x},\bar{z})$ is positive w.r.t. X. Then*

- *if $C(X)$ is negative w.r.t. X then*

$$\exists X\big\{\forall\bar{x}[A(X,\bar{x},\bar{z})\to X(\bar{x})]\wedge C(X)\big\} \;\equiv\; C(X)^{X(\bar{x})}_{\lambda\bar{x}.\mathrm{LFP}\left[X(\bar{x})\equiv A(X,\bar{x},\bar{z})\right]}. \quad (37)$$

- *if $C(X)$ is positive w.r.t. X then*

$$\exists X\big\{\forall\bar{x}[X(\bar{x})\to A(X,\bar{x},\bar{z})]\wedge C(X)\big\} \;\equiv\; C(X)^{X(\bar{x})}_{\lambda\bar{x}.\mathrm{GFP}\left[X(\bar{x})\equiv A(X,\bar{x},\bar{z})\right]}. \quad (38)$$

■

The following lemma is a particular case of Theorem 5.2 (thus also of Theorem 5.1), when A contains no occurrences of the eliminated relational variable.

LEMMA 5.3 (Ackermann [2]). *Let X be a relation variable and $A(\bar{x},\bar{z})$, $C(X)$ be classical first-order formulas, where the number of distinct variables in $\bar{x}$ is equal to the arity of X. Let A contain no occurrences of X. Then*

- *if $C(X)$ is negative w.r.t. X then*

$$\exists X\big\{\forall\bar{x}[A(\bar{x},\bar{z})\to X(\bar{x})]\wedge C(X)\big\} \;\equiv\; C(X)^{X(\bar{x})}_{\lambda\bar{x}.A(\bar{x},\bar{z})}. \quad (39)$$

- *if $C(X)$ is positive w.r.t. X then*

$$\exists X\big\{\forall\bar{x}[X(\bar{x})\to A(\bar{x},\bar{z})]\wedge C(X)\big\} \;\equiv\; C(X)^{X(\bar{x})}_{\lambda\bar{x}.A(\bar{x},\bar{z})}. \quad (40)$$

■

5.3. Strengthening the Method

Observe that axioms of annotation theories are often formulated in the form of "rules". In considered examples of annotation theories, formulas (9)–(11), (13)–(15) and (23)–(25) have a form of rules. Also formula $\sigma(n)$ can be considered as a rule due to its form expressed as (7).

We shall strengthen the method formalized as Theorem 5.1 by using intuitions from the semantics of stratified logic programs and Datalog$^{\neg}$, where recursion is not allowed to pass negation (cf. [1]). Namely, when we have rules with negated atoms in bodies, Theorem 5.1 is not applicable, but actually we can apply the theorem for separate strata and collect the results. The following example illustrates the idea.

Example 5.4. Consider the following second-order formula

$$\exists X \exists Y \big\{ \forall x[\neg X(x) \wedge \neg Y(x)] \tag{41}$$
$$\forall x[\exists y[R(x,y) \vee X(y)] \to X(x)] \wedge \tag{42}$$
$$\forall x\big[\big(\exists y[R(x,y) \wedge Y(y)] \vee \neg X(x)\big) \to Y(x)\big]\big\}. \tag{43}$$

Theorem 5.1 cannot be applied due to the negative literal $\neg X(x)$ in (43). On the other hand, one can first "compute" X using (42) and then use its definition "computing" Y. Here we can even apply Theorem 5.2. The definition of X is given by:

$$\mathrm{LFP}\big[X(x) \equiv \exists y[R(x,y) \vee X(y)]\big] \tag{44}$$

and, given (44) and (43), the definition of Y is given by

$$\mathrm{LFP}\big[Y(x) \equiv \exists y[R(x,y) \wedge Y(y)] \vee \neg X(x)\big]. \tag{45}$$

Formula (41)–(41) is then equivalent to $\forall x[\neg X(x) \wedge \neg Y(x)]$, where X and Y are given by their definitions (44) and (45). ■

Other examples of applications of this method are provided in the next section.

In the rest of this section we formulate and prove the main second-order quantifier elimination result of this paper, substantially extending Theorem 5.1. The strengthened version of the theorem can be formalized as follows.

Definition 5.5. Let $A(P, \bar{T}, \bar{x})$ be a classical first-order formula positive w.r.t. P with $P, \bar{T}$ being its all relation symbols.

A PIA *formula w.r.t.* P is any formula of the form $\forall\bar{x}[A(P,\bar{T},\bar{x})\rightarrow P(\bar{x})]$.[5] By a PIA *formula* we understand a PIA formula w.r.t. P for some P.

Dually, by an AIP *formula w.r.t.* P we understand any formula of the form $\forall\bar{x}[P(\bar{x}) \rightarrow A(P,\bar{T},\bar{x})]$.[6] By an AIP *formula* we understand an AIP formula w.r.t. P for some P.

A *rule* is a PIA formula or an AIP formula. In both cases the atom is called the *head* and the antecedent (consequent) is called the *body* of the rule. For a rule ρ, the head of ρ is denoted by $head(\rho)$ and the body of ρ is denoted by $body(\rho)$.

By a PIA *set* we understand any finite set of PIA formulas and by an AIP *set* we understand any finite set of AIP formulas. A *set of rules* is either a PIA set or an AIP set $\mathcal{S}$ such that any head of a rule of $\mathcal{S}$ appears in $\mathcal{S}$ in exactly one rule.[7] ■

Observe that the restriction as to uniqueness of heads' occurrences in rules is introduced solely to simplify presentation. Any set of PIA formulas and of AIP formulas can easily be transformed to a set satisfying this requirement. It suffices to rename variables and use the following tautologies:

$$\begin{array}{l}
\big(\forall\bar{x}[A(\bar{x})] \wedge \forall\bar{x}[B(\bar{x})]\big) \equiv \forall\bar{x}[A(\bar{x}) \wedge B(\bar{x})]\\
\big((A \rightarrow C) \wedge (B \rightarrow C)\big) \equiv \big((A \vee B) \rightarrow C\big)\\
\big((A \rightarrow B) \wedge (A \rightarrow C)\big) \equiv \big(A \rightarrow (B \wedge C)\big).
\end{array}$$

The following definition generalizes the well-known definition of stratification of logic programs.

DEFINITION 5.6. A stratification of a set of rules $\mathcal{S}$ is a partition $\mathcal{S}_1,\ldots,\mathcal{S}_l$ of $\mathcal{S}$ such that there is a mapping δ from the set of heads appearing in $\mathcal{S}$ to $\{1,\ldots,l\}$, satisfying:

- all rules with the same head P are in the same partition $\mathcal{S}_{\delta(P)}$
- if $\big(\forall\bar{x}[A(P,\bar{T},\bar{x}) \rightarrow P(\bar{x})]\big) \in \mathcal{S}$ (dually, if $\big(\forall\bar{x}[P(\bar{x}) \rightarrow A(P,\bar{T},\bar{x})]\big) \in \mathcal{S}$) and $P,\bar{T}$ are all relation symbols occurring in A, then for any T_i in $\bar{T}$:
  - if A is positive w.r.t. T_i then $\delta(T_i) \leq \delta(P)$
  - if there is a negative occurrence of T_i in A then $\delta(T_i) < \delta(P)$.

Given a stratification $\mathcal{S}_1,\ldots\mathcal{S}_l$ of $\mathcal{S}$, each $\mathcal{S}_i$ is called a *stratum* of the stratification, and δ is called the *stratification mapping.* ■

---

[5] The acronym PIA, introduced in [35], stands for "Positive antecedent Implies Atom".

[6] AIP stands for "Atom Implies Positive consequent".

[7] Let us emphasize that we do not consider sets containing both PIA and AIP formulas.

We shall need an ordering on the set of heads of considered sets of rules, preserving the stratification mapping.

DEFINITION 5.7. Let r be the cardinality of a set of rules $\mathcal{S}$ stratifiable with a stratification mapping δ. A mapping $\gamma : \{1, \dots, r\} \longrightarrow \{1, \dots, r\}$ is a δ-*order* if it is one-to-one, onto and, for $1 \leq i \leq j \leq r$, satisfies the condition:

$$\delta\big(head(\text{rule } \gamma(i))\big) \leq \delta\big(head(\text{rule } \gamma(j))\big). \qquad \blacksquare$$

Let $\bar{Q} = \langle Q_1, \dots, Q_k \rangle$ be a tuple of relation symbols and $A_i(\bar{Q}, \bar{x}_i, \bar{y}_i)$, for $i = 1, \dots, k$, be classical first-order formulas, where

- $\bar{x}_i$ and $\bar{y}_i$ are all free first-order variables of A_i
- the number of variables in $\bar{x}$ is k_i
- none of the x's is among the y's
- for $i = 1, \dots, k$, Q_i is a k_i-argument relation symbol
- the set of rules with bodies A_i and heads Q_i is stratifiable.

DEFINITION 5.8. Under the above assumptions, the expression

$$\text{StLFP}[Q_1(\bar{x}_1) \equiv A_1(\bar{Q},\, \bar{x}_1, \bar{y}_1), \dots, Q_k(\bar{x}_k) \equiv A_k(\bar{Q},\, \bar{x}_k, \bar{y}_k)] \tag{46}$$

is called the *stratified least fixpoint of* $A_1, \dots, A_k$, and the expression

$$\text{StGFP}[Q_1(\bar{x}_1) \equiv A_1(\bar{Q},\, \bar{x}_1, \bar{y}_1), \dots, Q_k(\bar{x}_k) \equiv A_k(\bar{Q},\, \bar{x}_k, \bar{y}_k)] \tag{47}$$

is called the *stratified greatest fixpoint of* $A_1, \dots, A_k$. ■

DEFINITION 5.9. Let $\mathcal{S}_1, \dots, \mathcal{S}_l$ $(1 \leq i \leq l)$ be strata of sets of rules considered in Definition 5.8. The semantics of $\text{StLFP}[\bar{Q} \equiv \bar{A}]$ is given by (the unique) tuple of relations $\bar{Q}$ given by $\text{SLFP}\,[\mathcal{S}_1], \dots, \text{SLFP}\,[\mathcal{S}_l]$. The semantics of $\text{StGFP}[\bar{Q} \equiv \bar{A}]$ is given by (the unique) tuple of relations $\bar{Q}$ given by $\text{SGFP}\,[\mathcal{S}_1], \dots, \text{SGFP}\,[\mathcal{S}_l]$. ■

We are now in position to formulate the elimination theorem. The intuition behind its proof is that one eliminates quantifiers starting from the first stratum and proceeds stratum by stratum until the last one. So, in the result, for each stratum there is a corresponding simultaneous fixpoint.

THEOREM 5.10. *Let* $\bar{X} = X_1, \dots, X_k$ *be distinct relation variables and* $C(\bar{X})$, $A_i(\bar{X}, \bar{x}_1, \dots, \bar{x}_k, \bar{z})$ $(1 \leq i \leq k)$ *be classical first-order formulas, where the number of distinct variables in* $\bar{x}_i$ *is equal to the arity of* X_i. *Then:*

- *if* $\{\forall\bar{x}[A_i(\bar{X},\ldots) \to X_i(\bar{x}_i)] \mid 1 \le i \le k\}$ *is stratifiable with a stratification mapping* δ*,* γ *is a* δ*-order and* $C(\bar{X})$ *is negative w.r.t.* $X_1,\ldots,X_k$ *then:*

$$\exists X_1 \ldots \exists X_k \left\{ \bigwedge_{1\le i\le k} \forall\bar{x}_i[A_i(\bar{X},\bar{x}_1,\ldots,\bar{x}_k,\bar{z}) \to X_i(\bar{x}_i)] \wedge\ C(\bar{X}) \right\} \equiv C(\bar{X})^{X_{\gamma(1)}(\bar{x}_{\gamma(1)}),\ldots,X_{\gamma(k)}(\bar{x}_{\gamma(k)})}_{\lambda\bar{x}_{\gamma(1)},\ldots,\bar{x}_{\gamma(k)}\mathrm{STLFP}[X_{\gamma(1)}(\bar{x}_{\gamma(1)})\equiv A_{\gamma(1)}(\bar{X},\ldots),\ldots,X_{\gamma(k)}(\bar{x}_{\gamma(k)})\equiv A_{\gamma(k)}(\bar{X},\ldots)]}. \tag{48}$$

- *if* $\{\forall\bar{x}[X_i(\bar{x}_i) \to A_i(\bar{X},\ldots)] \mid 1 \le i \le k\}$ *is stratifiable with a stratification mapping* δ*,* γ *is a* δ*-order and* $C(\bar{X})$ *is positive w.r.t.* $X_1,\ldots,X_k$ *then:*

$$\exists X_1 \ldots \exists X_k \left\{ \bigwedge_{1\le i\le k} \forall\bar{x}_i[X_i(\bar{x}_i) \to A_i(\bar{X},\bar{x}_1,\ldots,\bar{x}_k,\bar{z})] \wedge\ C(\bar{X}) \right\} \equiv C(\bar{X})^{X_{\gamma(1)}(\bar{x}_{\gamma(1)}),\ldots,X_{\gamma(k)}(\bar{x}_{\gamma(k)})}_{\lambda\bar{x}_{\gamma(1)},\ldots,\bar{x}_{\gamma(k)}\mathrm{STGFP}[X_{\gamma(1)}(\bar{x}_{\gamma(1)})\equiv A_{\gamma(1)}(\bar{X},\ldots),\ldots,X_{\gamma(k)}(\bar{x}_{\gamma(k)})\equiv A_{\gamma(k)}(\bar{X},\ldots)]}. \tag{49}$$

PROOF. We prove the theorem by induction on the number of strata $l \ge 1$.

The case of a single stratum, $l = 1$, is formulated and proved as Theorem 5.1.

Assume that the theorem holds for sets of rules with $l > 1$ strata. Let $\mathcal{S}$ a PIA set with $l+1$ strata, $\mathcal{S}_1,\ldots,\mathcal{S}_l,\mathcal{S}_{l+1}$, consisting of k rules. First, reorder existential second-order quantifiers in (48) (respectively, (49)) according to γ from right to left, so that the resulting sequence of quantifiers is $\exists\, head(\text{rule } \gamma(k)) \ldots \exists\, head(\text{rule } \gamma(1))$.

Existential second-order quantifiers binding heads of stratum $l+1$ are the outermost ones. Use the inductive assumption to eliminate all quantifiers except those binding heads of stratum $l+1$. In the resulting formula substitute any occurrence of a fixpoint operator of the form $\mathrm{SLFP}\,[\bar{R} \equiv \bar{U}]$ by a new relation symbol, say N applied to the same arguments as $\mathrm{SLFP}\,[\bar{R} \equiv \bar{U}]$. Next, add to the theory definitions of the new symbols as the respective fixpoints by using equivalences of the form:

$$\forall x\big[N(\bar{x}) \equiv \mathrm{SLFP}\,[\bar{R} \equiv \bar{U}]\big]. \tag{50}$$

Now we apply Theorem 5.1, replace N's by fixpoints according to respective definitions (50) and finally remove those definitions. ■

REMARK 5.1.

1. As noted in the above proof, Theorem 5.1 is a corollary of Theorem 5.10 in the case when there is a stratification consisting of a single stratum.
2. Observe that Theorem 5.10 can be extended to higher-order contexts along the lines of the elimination theorem of Gabbay and Szałas proved in [16]. ■

6. Reducing Circumscription Formulas in Annotation Theories

6.1. Discussion

We have already indicated in Remark 4.1 that axioms of the form $\sigma(n) \wedge \pi(n)$, appearing in Δ^{mn}-theories, make second-order quantifier techniques presented in Section 5 rarely applicable.

In all examples of annotation theories we consider, formulas contain recursive clauses, which excludes the possibility of eliminating all second-order quantifiers using the lemma of Ackermann (i.e., Lemma 5.3). The resulting formulas are then at least formulas of the fixpoint logic. This makes the SCAN algorithm [14] inapplicable, too.

So a candidate could be the Theorem 5.1, which to our best knowledge provides the strongest second-order quantifier elimination method that could be applied here.[8] As we show below, this theorem is not directly applicable, too. The argument is based on van Benthem's result [35]. To present the result we need the following definition.

DEFINITION 6.1. A first-order formula $A(P, \bar{T})$ has the *intersection property w.r.t.* P iff in any relational structure M, whenever $M, P_i \models A(P, \bar{T})$ for all predicates in a family $\{P_i \mid i \in I\}$, $A(P, \bar{T})$ also holds for their intersection, i.e., we have that $M, \bigcap_{i \in I} P_i \models A(P, \bar{T})$. ■

The following theorem has been proved by van Benthem in [35].

THEOREM 6.2 (van Benthem [35]). *The following are equivalent for all first-order formulas $A(P, \bar{T})$:*

1. *$A(P, \bar{T})$ has the intersection property w.r.t. P;*
2. *There is a* PIA *formula equivalent to $A(P, \bar{T})$.* ■

---

[8] Except, of course, for our Theorem 5.10.

We now have the following corollary.

Corollary 6.3. *For $n > 1$ and $m \geq 1$, no Δ^{mn}-theory theory is equivalent to a conjunction of* Pia *formulas.*

Proof. Let $n > 1$. Then the formula $\sigma(n) \wedge \pi(n)$ of Δ^{mn}-theory expresses the fact that every node of a graph is uniquely annotated. The intersection of two different annotations is then inconsistent. ■

The above result shows that the method based on Theorem 5.1 is not directly applicable in the case of the first form (32) of formulas required there.[9]

This shows that the method introduced in Section 5.3, based on Theorem 5.10 is indeed substantial.

6.2. Useful Simplifications

As already discussed, axioms $\sigma(n) \wedge \pi(n)$ of Δ^{mn}-theories are a source of difficulties in applying second-order quantifier techniques. The following two lemmas allow us to simplify the formulas. Namely, in the case of minimization one can remove $\pi(n)$ (together with other conjuncts negative w.r.t. minimized relations) from the second-order part of circumscription formula (1). Dually, in the case of maximization one can remove $\sigma(n)$ (together with other conjuncts positive w.r.t. maximized relations) from the second-order part of circumscription formula (2).

Below we first consider circumscribed theories without varied predicates, so theories are denoted by $T(\bar{P})$ rather than $T(\bar{P}, \bar{S})$, as $\bar{S} = \emptyset$.

Lemma 6.4. *Let $T(\bar{P})$ be a theory. Assume that T is of the form of conjunction $T_{\pm}(\bar{P}) \wedge T_{+}(\bar{P})$, where $T_{+}(\bar{P})$ is positive w.r.t. all relation symbols in $\bar{P}$. Then* $\mathrm{Circ}^{\uparrow}(T; \bar{P}, \emptyset)$ *is equivalent to*

$$T(\bar{P}) \wedge \forall \bar{X} \Big\{ \big[T_{\pm}(\bar{P})^{\bar{P}}_{\bar{X}} \wedge \bigwedge_{i=1}^{k} \forall \bar{x}_i [P_i(\bar{x}_i) \to X_i(\bar{x}_i)] \,\big] \to \bigwedge_{i=1}^{k} \forall \bar{x}_i [X_i(\bar{x}_i) \to P_i(\bar{x}_i)] \Big\}. \tag{51}$$

[9] Similar argument applies to the form required in (33). This can be seen by considering the contraposition of the implication and respectively replace $\neg X_i$'s by X_i's.

PROOF. [10] Since $T_+(\bar{P})$ positive w.r.t. all relation symbols in $\bar{P}$, it is also monotone w.r.t. all relation symbols in $\bar{P}$. We then have that for any relational structure M and valuation v:

$$M, v \models \left[T_+(\bar{P}) \wedge \bigwedge_{i=1}^{k} \forall \bar{x}_i [P_i(\bar{x}_i) \rightarrow X_i(\bar{x}_i)] \right] \rightarrow T_+(\bar{P})^{\bar{P}}_{\bar{X}}.$$

Thus, in the presence of the conjunct $T(\bar{P})$, formula (51) obtained from (2) by removing $T_+(\bar{P})^{\bar{P}}_{\bar{X}}$ is equivalent to (2). ■

LEMMA 6.5. *Let $T(\bar{P})$ be a theory. Assume that T is of the form of conjunction $T_\pm(\bar{P}) \wedge T_-(\bar{P})$, where $T_-(\bar{P})$ is negative w.r.t. all relation symbols in $\bar{P}$. Then* $\mathrm{Circ}_\downarrow(T; \bar{P}; \emptyset)$ *is equivalent to*

$$T(\bar{P}) \wedge \forall \bar{X} \Big\{ \big[T_\pm(\bar{P})^{\bar{P}}_{\bar{X}} \wedge \bigwedge_{i=1}^{k} \forall \bar{x}_i [X_i(\bar{x}_i) \rightarrow P_i(\bar{x}_i)] \; \big] \rightarrow \bigwedge_{i=1}^{k} \forall \bar{x}_i [P_i(\bar{x}_i) \rightarrow X_i(\bar{x}_i)] \Big\}. \tag{52}$$

PROOF. Similar to the proof of Lemma 6.4, by observing that contraposition of all implications $\forall \bar{x}_i [X_i(\bar{x}_i) \rightarrow P_i(\bar{x}_i)]$ together with the monotonicity of T_- w.r.t. $\neg P_i$ imply the result. ■

In order to deal with varied predicates we use the following observation, allowing one to eliminate varied predicates.

PROPOSITION 6.6 (Lifschitz [24]). *The circumscription* $\mathrm{Circ}_\downarrow(T(\bar{P}, \bar{S}); \bar{P}; \bar{S})$ *is equivalent to* $T(\bar{P}, \bar{S}) \wedge \mathrm{Circ}_\downarrow(\exists \bar{Y} [T(\bar{P}, \bar{S})^{\bar{S}}_{\bar{Y}}]; \bar{P}; \emptyset)$. ■

Similarly, we have analogous proposition for the dual form of circumscription.

PROPOSITION 6.7. *The circumscription* $\mathrm{Circ}^\uparrow(T(\bar{P}, \bar{S}); \bar{P}; \bar{S})$ *is equivalent to* $T(\bar{P}, \bar{S}) \wedge \mathrm{Circ}^\uparrow(\exists \bar{Y} [T(\bar{P}, \bar{S})^{\bar{S}}_{\bar{Y}}]; \bar{P}; \emptyset)$. ■

The following proposition, known as the *purity deletion principle*, is sometimes useful.

[10] The proof is similar to the one given by Lifschitz [26] for proving a reduction result for separated formulas, but we deal with a more general context here.

PROPOSITION 6.8 (Szałas [34]). *Let A be a classical first-order formula of the form $Q_1x_1....Q_rx_r[A_1 \wedge ... \wedge A_q]$, where $Q_1,\ldots,Q_r \in \{\exists,\forall\}$ and each A_1, ..., A_q containing an occurrence of P is of the form $\big(B \vee P(\bar{z})\big)$ with B being any first-order formula, possibly containing arbitrary occurrences of P. Then the formula $\exists P[A]$ is equivalent to $Q_1x_1....Q_rx_r[A_{i_1} \wedge ... \wedge A_{i_s}]$, where $i_1,...,i_s \in \{1,...,q\}$ and A_{i_1}, ..., A_{i_s} are all conjuncts that do not contain occurrences of P (the empty conjunction is, by convention,* TRUE*).*

The same holds when each A_1, ..., A_q containing an occurrence of P is of the form $\big(B \vee \neg P(\bar{z})\big)$. ■

6.3. Reducing Circumscription in Phan's Argumentation Theory

In this section we consider the Phan's theory $\mathcal{A}(R,Q_1,Q_2,Q_3)$ specified in Section 3.2. We show that all circumscriptive policies considered there are reducible to fixpoint logic.

6.3.1. Minimization on Q_2

Consider the circumscription formula $Circ_{\downarrow}(\mathcal{A}(R,Q_1,Q_2,Q_3);Q_2;Q_1,Q_3)$:

$$\mathcal{A}(R,Q_1,Q_2,Q_3)\wedge \tag{53}$$

$$\forall X_1\forall X_2\forall X_3[(\mathcal{A}(R,X_1,X_2,X_3)\wedge\forall x[X_2(x)\to Q_2(x)])\to \forall x[Q_2(x)\to X_2(x)]]. \tag{54}$$

We focus on (54), which is equivalent to

$$\begin{aligned}&\neg\exists X_1\exists X_2\exists X_3\big\{\forall x[X_1(x)\vee X_2(x)\vee X_3(x)]\wedge\\ &\forall x[\neg X_1(x)\vee\neg X_2(x)]\wedge\forall x[\neg X_1(x)\vee\neg X_3(x)]\wedge\forall x[\neg X_2(x)\vee\neg X_3(x)]\wedge\\ &\forall x[\forall y[R(y,x)\to X_1(y)]\to X_2(x)]\wedge\\ &\forall x[\exists y[R(y,x)\wedge X_2(y)]\to X_1(x)]\wedge\\ &\forall x[(\forall y[R(y,x)\to(X_1(y)\vee X_3(y))]\wedge\exists y[R(y,x)\wedge X_3(y)])\to X_3(x)]\wedge\\ &\forall x[X_2(x)\to Q_2(x)]\wedge\exists z[Q_2(z)\wedge\neg X_2(z)]]\big\}.\end{aligned}$$

Using (7) and minor transformations we obtain

$$\neg\exists X_1\exists X_2\exists X_3\big\{\forall x[X_2(x)\to Q_2(x)]\wedge\exists z[Q_2(z)\wedge\neg X_2(z)]\wedge \tag{55}$$

$$\forall x[\neg X_1(x)\vee\neg X_2(x)]\wedge\forall x[\neg X_1(x)\vee\neg X_3(x)]\wedge\forall x[\neg X_2(x)\vee\neg X_3(x)]\wedge \tag{56}$$

$$\forall x[\forall y[R(y,x)\to X_1(y)]\to X_2(x)]\wedge \tag{57}$$

$$\forall x[\exists y[R(y,x)\wedge X_2(y)]\to X_1(x)]\wedge \tag{58}$$

$$\forall x\big[\big(\forall y[R(y,x)\to(X_1(y)\vee X_3(y))]\wedge\exists y[R(y,x)\wedge X_3(y)]\vee (\neg X_1(x)\wedge\neg X_2(x))\big)\to X_3(x)\big]\big\}. \tag{59}$$

We have two strata: $\{(57),(58)\}$ and $\{(59)\}$. Using Theorem 5.10, we first simultaneously eliminate $\exists X_1 \exists X_2$ and then $\exists X_3$.

The elimination of $\exists X_1 \exists X_2$ provides us with the following definition of X_1 and X_2:

$$\textsc{Slfp}\big[X_1(x) \equiv \exists y[R(y,x) \wedge X_2(y)],\ X_2(x) \equiv \forall y[R(y,x) \rightarrow X_1(y)]\big]. \quad (60)$$

The elimination of $\exists X_3$ provides us with the following definition of X_3:

$$\begin{aligned}\textsc{Lfp}\big[X_3(x) \equiv \forall y[R(y,x) \rightarrow (X_1(y) \vee X_3(y))] \wedge \exists y[R(y,x) \wedge X_3(y)]\\ \vee(\neg X_1(x) \wedge \neg X_2(x))\big].\end{aligned} \quad (61)$$

The result of elimination is then

$$\begin{aligned}\neg\big\{\forall x[X_2(x) \rightarrow Q_2(x)] \wedge \exists z[Q_2(z) \wedge \neg X_2(z)] \wedge\\ \forall x[\neg X_1(x) \vee \neg X_2(x)] \wedge \forall x[\neg X_1(x) \vee \neg X_3(x)] \wedge \forall x[\neg X_2(x) \vee \neg X_3(x)]\big\}\end{aligned} \quad (62)$$

with X_1, X_2, X_3 respectively substituted by definitions given by formulas (60) and (61).

Formula (62) can then be presented in a more readable form as:

$$\left.\begin{aligned}\forall x[X_2(x) \rightarrow Q_2(x)] \wedge \forall x[\neg X_1(x) \vee \neg X_2(x)] \wedge\\ \forall x[\neg X_1(x) \vee \neg X_3(x)] \wedge \forall x[\neg X_2(x) \vee \neg X_3(x)]\end{aligned}\right\} \rightarrow \forall z[Q_2(z) \rightarrow X_2(z)].$$

Since the result is a fixpoint formula, we have the following corollary.

Corollary 6.9. *Model checking problem for the theory* $\mathcal{A}(R, Q_1, Q_2, Q_3)$ *with circumscriptive policy expressed by* $\mathrm{Circ}_{\downarrow}(\mathcal{A}(R, Q_1, Q_2, Q_3); Q_2; Q_1, Q_3)$ *is in* PTime *in the size of the structure.* ■

6.3.2. Maximization on Q_2

Consider the dual circumscription formula $\mathit{Circ}^{\uparrow}(\mathcal{A}(R, Q_1, Q_2, Q_3); Q_2; Q_1, Q_3)$:

$$\mathcal{A}(R, Q_1, Q_2, Q_3) \wedge \quad (63)$$

$$\begin{aligned}\forall X_1 \forall X_2 \forall X_3\big[(\mathcal{A}(R, X_1, X_2, X_3) \wedge \forall x[Q_2(x) \rightarrow X_2(x)]) \rightarrow\\ \forall x[X_2(x) \rightarrow Q_2(x)]\big].\end{aligned} \quad (64)$$

We focus on (64), which is equivalent to

$$\begin{aligned}&\neg\exists X_1 \exists X_2 \exists X_3 \big\{\forall x[X_1(x) \vee X_2(x) \vee X_3(x)] \wedge\\ &\quad \forall x[\neg X_1(x) \vee \neg X_2(x)] \wedge \forall x[\neg X_1(x) \vee \neg X_3(x)] \wedge \forall x[\neg X_2(x) \vee \neg X_3(x)] \wedge\\ &\quad \forall x[\forall y[R(y,x) \rightarrow X_1(y)] \rightarrow X_2(x)] \wedge\\ &\quad \forall x[\exists y[R(y,x) \wedge X_2(y)] \rightarrow X_1(x)] \wedge\\ &\quad \forall x[(\forall y[R(y,x) \rightarrow (X_1(y) \vee X_3(y))] \wedge \exists y[R(y,x) \wedge X_3(y)]) \rightarrow X_3(x)] \wedge\\ &\quad \forall x[Q_2(x) \rightarrow X_2(x)] \wedge \exists z[X_2(z) \wedge \neg Q_2(z)]]\big\}\end{aligned}$$

and further to

$$\neg\exists z\exists X_1\exists X_2\exists X_3\big\{\neg Q_2(z)\wedge\forall x[\neg X_1(x)\vee\neg X_2(x)]\wedge \tag{65}$$

$$\forall x[\neg X_1(x)\vee\neg X_3(x)]\wedge\forall x[\neg X_2(x)\vee\neg X_3(x)]\wedge \tag{66}$$

$$\forall x[\big(\forall y[R(y,x)\to X_1(y)]\vee Q_2(x)\vee x=z\big)\to X_2(x)]\wedge \tag{67}$$

$$\forall x[\exists y[R(y,x)\wedge X_2(y)]\to X_1(x)]\wedge \tag{68}$$

$$\forall x\big[\big(\forall y[R(y,x)\to(X_1(y)\vee X_3(y))]\wedge\exists y[R(y,x)\wedge X_3(y)]\vee (\neg X_1(x)\wedge\neg X_2(x))\big)\to X_3(x)\big]\big\}. \tag{69}$$

We have two strata: $\{(67),(68)\}$, $\{(69)\}$. Using Theorem 5.10, we first simultaneously eliminate $\exists X_1\exists X_2$ and then $\exists X_3$.

The elimination of $\exists X_1\exists X_2$ provides us with the following definition of X_1 and X_2:

$$\begin{aligned}\textsc{Slfp}\big[X_1(x)&\equiv\exists y[R(y,x)\wedge X_2(y)],\\ X_2(x)&\equiv\big(\forall y[R(y,x)\to X_1(y)]\vee Q_2(x)\vee x=z\big)\big].\end{aligned} \tag{70}$$

The elimination of $\exists X_3$ provides us with the definition of X_3 given by (61). The result of elimination is then

$$\begin{aligned}\neg\exists z\big\{&\neg Q_2(z)\wedge\forall x[\neg X_1(x)\vee\neg X_2(x)]\wedge\\ &\forall x[\neg X_1(x)\vee\neg X_3(x)]\wedge\forall x[\neg X_2(x)\vee\neg X_3(x)]\big\}\end{aligned} \tag{71}$$

with X_1, X_2, X_3 respectively substituted by definitions provided by formulas (61) and (70).

Formula (71) can then be presented in a more readable form as:

$$\forall z\left\{\left(\begin{array}{l}\forall x[\neg X_1(x)\vee\neg X_2(x)]\wedge\\ \forall x[\neg X_1(x)\vee\neg X_3(x)]\wedge\forall x[\neg X_2(x)\vee\neg X_3(x)]\end{array}\right)\to Q_2(z)\right\}.$$

Note that the quantifier $\forall z$ cannot be moved to $Q_2(z)$, since z appears in the antecedent as a part of definition of X_2 given by (70).

Since the resulting formula is a fixpoint formula, we have the following corollary.

Corollary 6.10. *Model checking problem for the theory $\mathcal{A}(R,Q_1,Q_2,Q_3)$ with circumscriptive policy expressed by* $\mathrm{Circ}^{\uparrow}(\mathcal{A}(R,Q_1,Q_2,Q_3);Q_2;Q_1,Q_3)$ *is in* PTime *in the size of the model.* ■

6.3.3. Minimization on Q_3

Consider the circumscription formula $Circ_{\downarrow}(\mathcal{A};Q_3;Q_1,Q_2)$:

$$\mathcal{A}(R,Q_1,Q_2,Q_3)\wedge \tag{72}$$

$$\forall X_1\forall X_2\forall X_3[(\mathcal{A}(R,X_1,X_2,X_3)\wedge\forall x[X_3(x)\to Q_3(x)])\to \forall x[Q_3(x)\to X_3(x)]]. \tag{73}$$

We focus on (73), which is equivalent to

$$\begin{aligned}&\neg\exists X_1\exists X_2\exists X_3\big\{\forall x[X_1(x)\vee X_2(x)\vee X_3(x)]\wedge\\ &\quad\forall x[\neg X_1(x)\vee\neg X_2(x)]\wedge\forall x[\neg X_1(x)\vee\neg X_3(x)]\wedge\forall x[\neg X_2(x)\vee\neg X_3(x)]\wedge\\ &\quad\forall x[\forall y[R(y,x)\to X_1(y)]\to X_2(x)]\wedge\\ &\quad\forall x[\exists y[R(y,x)\wedge X_2(y)]\to X_1(x)]\wedge\\ &\quad\forall x[(\forall y[R(y,x)\to(X_1(y)\vee X_3(y))]\wedge\exists y[R(y,x)\wedge X_3(y)])\to X_3(x)]\wedge\\ &\quad\forall x[X_3(x)\to Q_3(x)]\wedge\exists z[Q_3(z)\wedge\neg X_3(z)]]\big\}.\end{aligned}$$

As in Section 6.3.1, we obtain

$$\neg\exists X_1\exists X_2\exists X_3\big\{\forall x[X_3(x)\to Q_3(x)]\wedge\exists z[Q_3(z)\wedge\neg X_3(z)]\wedge \tag{74}$$

$$\forall x[\neg X_1(x)\vee\neg X_2(x)]\wedge\forall x[\neg X_1(x)\vee\neg X_3(x)]\wedge\forall x[\neg X_2(x)\vee\neg X_3(x)]\wedge \tag{75}$$

$$\forall x[\forall y[R(y,x)\to X_1(y)]\to X_2(x)]\wedge \tag{76}$$

$$\forall x[\exists y[R(y,x)\wedge X_2(y)]\to X_1(x)]\wedge \tag{77}$$

$$\begin{aligned}&\forall x\big[\big(\forall y[R(y,x)\to(X_1(y)\vee X_3(y))]\wedge\exists y[R(y,x)\wedge X_3(y)]\vee\\ &\qquad\qquad(\neg X_1(x)\wedge\neg X_2(x))\big)\to X_3(x)\big]\big\}.\end{aligned} \tag{78}$$

We have two strata: $\{(76),(77)\}$ and $\{(78)\}$, which are the same as in Section 6.3.1. The result of elimination is then

$$\begin{aligned}&\neg\big\{\forall x[X_3(x)\to Q_3(x)]\wedge\exists z[Q_3(z)\wedge\neg X_3(z)]\wedge\\ &\quad\forall x[\neg X_1(x)\vee\neg X_2(x)]\wedge\forall x[\neg X_1(x)\vee\neg X_3(x)]\wedge\forall x[\neg X_2(x)\vee\neg X_3(x)]\big\}\end{aligned} \tag{79}$$

with X_1, X_2, X_3 respectively substituted by definitions given by formulas (60) and (61).

Formula (79) can be presented as:

$$\left.\begin{aligned}&\forall x[X_3(x)\to Q_3(x)]\wedge\forall x[\neg X_1(x)\vee\neg X_2(x)]\wedge\\ &\forall x[\neg X_1(x)\vee\neg X_3(x)]\wedge\forall x[\neg X_2(x)\vee\neg X_3(x)]\end{aligned}\right\}\to\forall z[Q_3(z)\to X_3(z)].$$

Since the result is a fixpoint formula, we have the following corollary.

COROLLARY 6.11. *Model checking problem for the theory* $\mathcal{A}(R, Q_1, Q_2, Q_3)$ *with circumscriptive policy expressed by* $\mathrm{Circ}_\downarrow(\mathcal{A}(R, Q_1, Q_2, Q_3); Q_3; Q_1, Q_2)$ *is in* PTIME *in the size of the structure.* ■

6.4. Reducing Circumscription in the Annotation Theory Related to Approximate Reasoning

Let us now illustrate techniques introduced in Section 6.2.

Consider the circumscription formula $Circ_{\downarrow}(\mathcal{R}; Q_1, Q_3; Q_2)$:

$$\mathcal{R}(R, Q_1, Q_2, Q_3)\wedge \tag{80}$$

$$\begin{aligned}&\forall X_1 \forall X_2 \forall X_3[(\mathcal{R}(R, X_1, X_2, X_3)\wedge\\ &\quad \forall x[X_1(x) \to Q_1(x)] \wedge \forall x[X_3(x) \to Q_3(x)]) \to\\ &\qquad \forall x[Q_1(x) \to X_1(x)] \wedge \forall x[Q_3(x) \to X_3(x)]].\end{aligned} \tag{81}$$

We focus on (81).

We first apply Proposition 6.6, so attempt to eliminate second-order quantifiers from

$$\begin{aligned}&\exists X_2\big\{\forall x[X_1(x) \vee X_2(x) \vee X_3(x)]\wedge\\ &\quad \forall x[\neg X_1(x) \vee \neg X_2(x)] \wedge \forall x[\neg X_1(x) \vee \neg X_3(x)] \wedge \forall x[\neg X_2(x) \vee \neg X_3(x)]\wedge\\ &\quad \forall x \forall y[(R(x, y) \wedge X_1(x)) \to X_1(y)]\wedge\\ &\quad \forall x \forall y[(R(x, y) \wedge X_2(x)) \to X_2(y)]\wedge\\ &\quad \forall x \forall y[(R(x, y) \wedge X_3(x)) \to X_3(y)]\big\}.\end{aligned}$$

The above formula is equivalent to

$$\begin{aligned}&\exists X_2\big\{\forall x[\neg X_1(x) \vee \neg X_2(x)] \wedge \forall x[\neg X_1(x) \vee \neg X_3(x)] \wedge \forall x[\neg X_2(x) \vee \neg X_3(x)]\wedge\\ &\quad \forall x \forall y[(R(x, y) \wedge X_1(x)) \to X_1(y)]\wedge\\ &\quad \forall y\big[\big((\neg X_1(y) \wedge \neg X_3(y)) \vee \exists x[R(x, y) \wedge X_2(x)]\big) \to X_2(y)\big]\wedge\\ &\quad \forall x \forall y[(R(x, y) \wedge X_3(x)) \to X_3(y)]\big\}.\end{aligned}$$

Using Theorem 5.2, we obtain the following definition of X_2:

$$\textsc{Lfp}\big[X_2(y) \equiv \big((\neg X_1(y) \wedge \neg X_3(y)) \vee \exists x[R(x, y) \wedge X_2(x)]\big)\big].$$

Now we can use Lemma 6.5, which allows us to remove formulas where X_1 and X_3 appear only negatively, and we obtain that (81) is equivalent to

$$\begin{aligned}&\neg \exists X_1 \exists X_3\big[\mathcal{R}'(R, X_1, X_2, X_3)\wedge\\ &\qquad \forall x[X_1(x) \to Q_1(x)] \wedge \forall x[X_3(x) \to Q_3(x)]\wedge\\ &\qquad \big(\exists x[Q_1(x) \wedge \neg X_1(x)] \vee \exists x[Q_3(x) \wedge \neg X_3(x)]\big)\big],\end{aligned} \tag{82}$$

where $\mathcal{R}'$ is the conjunction

$$\begin{aligned}&\forall x \forall y[(R(x, y) \wedge X_1(x)) \to X_1(y)]\wedge\\ &\forall x \forall y[(R(x, y) \wedge X_3(x)) \to X_3(y)].\end{aligned} \tag{83}$$

Formula (82) is then equivalent to

$$\neg\big\{\exists X_1\exists X_3\big[\mathcal{R}'(R,X_1,X_2,X_3)\wedge \tag{84}$$
$$\forall x[X_1(x)\rightarrow Q_1(x)]\wedge\forall x[X_3(x)\rightarrow Q_3(x)]\wedge \tag{85}$$
$$\exists x[Q_1(x)\wedge\neg X_1(x)]\big]\vee \tag{86}$$
$$\exists X_1\exists X_2\exists X_3\big[\mathcal{R}'(R,X_1,X_2,X_3)\wedge \tag{87}$$
$$\forall x[X_1(x)\rightarrow Q_1(x)]\wedge\forall x[X_3(x)\rightarrow Q_3(x)]\wedge \tag{88}$$
$$\exists x[Q_3(x)\wedge\neg X_3(x)]\big)\big]\big\}, \tag{89}$$

so in the scope of the outermost negation we have a disjunction of two second-order formulas, $\big((84)-(86)\big)\vee\big((87)-(89)\big)$. We start with the first disjunct:

$$\begin{aligned}\exists x\exists X_1\exists X_3\big[&\forall y[\exists x[R(x,y)\wedge X_1(x)]\rightarrow X_1(y)]\wedge\\ &\forall y[\exists x[R(x,y)\wedge X_3(x)]\rightarrow X_3(y)]\wedge\\ &\forall x[X_1(x)\rightarrow Q_1(x)]\wedge\forall x[X_3(x)\rightarrow Q_3(x)]\wedge\\ &Q_1(x)\wedge\neg X_1(x)\big].\end{aligned}$$

Observe that for both X_1 and X_3 satisfy assumptions of Proposition 6.8, so the formula reduces to $\exists x[Q_1(x)]$. Similarly, the second disjunct reduces to $\exists x[Q_3(x)]$. When we move negation inside, disjunction is switched to conjunction, so the final result is

$$\forall x[\neg Q_1(x)]\wedge\forall x[\neg Q_3(x)]. \tag{90}$$

Since (90) is a classical first-order formula, we have the following corollary.

COROLLARY 6.12. *Model checking problem for the theory* $\mathcal{R}(R,Q_1,Q_2,Q_3)$ *with circumscriptive policy expressed by* $\mathrm{Circ}_{\downarrow}(\mathcal{R};Q_1,Q_3;Q_2)$ *is in* LOGSPACE *(so also in* PTIME*) in the size of the structure.* ■

We have Corollary 6.12 because we did not place any positive facts as to Q_1 and Q_3. Such facts would contribute to the result. In such a case Theorem 5.1 would be applicable, so we have the following proposition.

PROPOSITION 6.13. *Model checking problem for the theory* $\mathcal{R}(R,Q_1,Q_2,Q_3)$ *with circumscriptive policy expressed by* $\mathrm{Circ}_{\downarrow}(\mathcal{R};Q_1,Q_3;Q_2)$ *and additional positive facts concerning* Q_1 *and/or* Q_2 *is in* PTIME *in the size of the model.* ■

6.5. Reducing Circumscription in the Formalization of Semantics of Logic Programs

Consider the circumscription formula $Circ_{\downarrow}(\mathcal{L}; Q_2; Q_1, Q_3)$:

$$\mathcal{L}(R_+, R_-, Q_1, Q_2, Q_3) \wedge \tag{91}$$

$$\forall X_1 \forall X_2 \forall X_3 \big[(\mathcal{L}(R_+, R_-, X_1, X_2, X_3) \wedge \forall x[X_2(x) \to Q_2(x)]) \to \\ \forall z[Q_2(z) \to X_2(z)]\big]. \tag{92}$$

We focus on (92), which is equivalent to

$$\neg \exists z \exists X_1 \exists X_2 \exists X_3 \big\{\forall x[X_2(x) \to Q_2(x)] \wedge Q_2(z) \wedge \neg X_2(z) \wedge \tag{93}$$

$$\forall x[\neg X_1(x) \vee \neg X_2(x)] \wedge \forall x[\neg X_1(x) \vee \neg X_3(x)] \wedge \forall x[\neg X_2(x) \vee \neg X_3(x)] \wedge \tag{94}$$

$$\forall x\big[\big(\exists y[R_+(y,x) \wedge X_1(y)] \vee \exists y[R_-(y,x) \wedge X_2(y)]\big) \to X_1(x)\big] \wedge \tag{95}$$

$$\forall x\big[\big(\forall y[R_+(y,x) \to X_2(y)] \wedge \forall y[R_-(y,x) \to X_1(y)]\big) \to X_2(x)\big] \wedge \tag{96}$$

$$\forall x\big[\big(\forall y[R_+(y,x) \to (X_2(y) \vee X_3(y))] \wedge \\ \forall y[R_-(y,x) \to (X_1(y) \vee X_3(y))] \wedge \\ \exists y[\big((R_-(y,x) \vee R_+(y,x)) \wedge X_3(y)\big)] \vee (\neg X_1(x) \wedge \neg X_2(x))\big) \to X_3(x)\big]. \tag{97}$$

We have two strata: $\{(95), (96)\}$ and $\{(97)\}$. Using Theorem 5.10, we first simultaneously eliminate $\exists X_1 \exists X_2$ and then $\exists X_3$.

The elimination of $\exists X_1 \exists X_2$ provides us with the following definition of X_1 and X_2:

$$\text{SLFP}\big[X_1(x) \equiv \big(\exists y[R_+(y,x) \wedge X_1(y)] \vee \exists y[R_-(y,x) \wedge X_2(y)]\big), \\ X_2(x) \equiv \big(\forall y[R_+(y,x) \to X_2(y)] \wedge \forall y[R_-(y,x) \to X_1(y)]\big)\big]. \tag{98}$$

The elimination of $\exists X_3$ provides us with the following definition of X_3:

$$\text{LFP}\big[X_3(x) \equiv \big(\forall y[R_+(y,x) \to (X_2(y) \vee X_3(y))] \wedge \\ \forall y[R_-(y,x) \to (X_1(y) \vee X_3(y))] \wedge \\ \exists y[\big((R_-(y,x) \vee R_+(y,x)) \wedge X_3(y)\big)] \vee (\neg X_1(x) \wedge \neg X_2(x))\big)\big]. \tag{99}$$

The result of elimination is then

$$\neg \exists z\big[\forall x[X_2(x) \to Q_2(x)] \wedge Q_2(z) \wedge \neg X_2(z) \wedge \forall x[\neg X_1(x) \vee \neg X_2(x)] \wedge \\ \forall x[\neg X_1(x) \vee \neg X_3(x)] \wedge \forall x[\neg X_2(x) \vee \neg X_3(x)]\big], \tag{100}$$

with X_1, X_2, X_3 respectively substituted by definitions given by formulas (98) and (99).

Formula (100) can be presented in a more readable form:

$$\left.\begin{array}{l} \forall x\{X_2(x) \to Q_2(x)] \wedge \forall x[\neg X_1(x) \vee \neg X_2(x)] \wedge \\ \forall x[\neg X_1(x) \vee \neg X_3(x)] \wedge \forall x[\neg X_2(x) \vee \neg X_3(x)] \end{array}\right\} \to \forall z[Q_2(z) \to X_2(z)].$$

Since the result is a fixpoint formula, we have the following corollary.

COROLLARY 6.14. *Model checking problem for theory* $\mathcal{L}(R_+, R_-, Q_1, Q_2, Q_3)$ *with circumscriptive policy expressed by* $\mathrm{Circ}_{\downarrow}(\mathcal{L}; Q_2; Q_1, Q_3)$ *is in* PTIME *in the size of the structure.* ■

7. Conclusions

In the paper we introduced the concept of annotation theories and showed that such theories, together with minimization/maximization policies expressed by means of circumscription, are rich enough to capture important phenomena appearing in many applications, including specific theories of argumentation, approximate reasoning as well as semantics of logic programs with negation.

Even if circumscription is substantially a second-order formalism, we provided a number of results allowing to eliminate second-order quantifiers. Even simpler methods, based on results from [11, 28] appear quite powerful and applicable to a wide class of circumscribed formulas (see also [10, 15]). The problem of quantifier elimination in annotation theories appears, in general, as difficult as the question whether PTIME =NPTIME, so considering particular annotation theories is an interesting research area, far from being completed.

Annotation theories deserve further investigations. In particular, an interesting problem is to search for algorithms for finding annotations, especially ones that construct the model incrementally on the graph. In general this problem is as difficult as PTIME =NPTIME, as shown in Section 4. However, in the case of stratified theories we can apply Theorem 5.10, allowing us to reduce complexity to PTIME.[11]

Also, using theory approximation [32, 33, 5, 6, 8] is worth investigating in the context of second-order quantifier elimination from circumscribed annotation theories. Namely, when such elimination is not possible using Theorem 5.10, one can approximate considered theories by theories admitting quantifier elimination. We leave these subjects for future research.

Acknowledgements. We would like to thank Linh Anh Nguyen for careful reading of this paper and many helpful comments and suggestions.

This work has been supported in part by the grant N N206 399134 of Polish Ministry of Science and Higher Education and by grants from the Swedish Foundation for Strategic Research (SSF) Strategic Research Center MOVIII and the Swedish Research Council (VR) Linnaeus Center CADICS.

---

[11] This method is applicable to annotation theories considered in Sections 3.2, 3.3 and 3.4.

References

[1] Abiteboul, S., R. Hull, and V. Vianu, *Foundations of Databases*, Addison-Wesley Pub. Co., 1996.

[2] Ackermann, W., 'Untersuchungen über das eliminationsproblem der mathematischen logik', *Mathematische Annalen*, 110 (1935), 390–413.

[3] Baader, F., D. Calvanese, D.L. McGuinness, D. Nardi, and P.F. Patel-Schneider, (eds.) *Description Logic Handbook*, Cambridge University Press, 2002.

[4] Blackburn, P., and J. Seligman, 'Hybrid languages', *Journal of Logic, Language and Information*, 4 (1995), 41–62.

[5] Cadoli, M., *Tractable Reasoning in Artificial Intelligence*, vol. 941 of *LNCS*, Springer-Verlag, 1995.

[6] Cadoli, M, and F.M. Donini, 'A survey on knowledge compilation', *AI Communications*, 10 (1997), 3-4, 137–150.

[7] Caminada, M., and D.M. Gabbay, 'A logical account of formal argumentation', *Studia Logica*, 93 (2009), 2-3, 109–145, this issue.

[8] Darwiche, A., and P. Marquis, 'A knowledge compilation map', *Journal of Artificial Intelligence Research*, 17 (2002), 229–264.

[9] Doherty, P., W. Łukaszewicz, A. Skowron, and A. Szałas, *Knowledge representation techniques. A rough set approach*, vol. 202 of *Studies in Fuziness and Soft Computing*, Springer-Verlag, 2006.

[10] Doherty, P., W. Łukaszewicz, and A. Szałas, 'A reduction result for circumscribed semi-Horn formulas', *Fundamenta Informaticae*, 28 (1996), 3–4, 261–271.

[11] Doherty, P., W. Łukaszewicz, and A. Szałas, 'Computing circumscription revisited', *Journal of Automated Reasoning*, 18 (1997), 3, 297–336.

[12] Doherty, P., and A. Szałas, 'On the correspondence between approximations and similarity', in S. Tsumoto, R. Slowinski, J. Komorowski, and J.W. Grzymala-Busse, (eds.), *Proc. RSCTC'2004*, vol. 3066 of *LNAI*, 2004, pp. 143–152.

[13] Ebbinghaus, H-D., and J. Flum, *Finite Model Theory*, Springer-Verlag, Heidelberg, 1995.

[14] Gabbay, D. M., and H. J. Ohlbach, 'Quantifier elimination in second-order predicate logic', *South African Computer Journal*, 7 (1992), 35–43.

[15] Gabbay, D.M., R. Schmidt, and A. Szałas, *Second-Order Quantifier Elimination. Foundations, Computational Aspects and Applications*, vol. 12 of *Studies in Logic*, College Publications, 2008.

[16] Gabbay, D.M., and A. Szałas, 'Second-order quantifier elimination in higher-order contexts with applications to the semantical analysis of conditionals', *Studia Logica*, 87 (2007), 37–50.

[17] Gelfond, M., and V. Lifschitz, 'The stable model semantics for logic programming', in *Proceedings of the Fifth International Conference on Logic Programming (ICLP)*, 1988, pp. 1070–1080.

[18] Goranko, V., 'Temporal logics with reference pointers and computation tree logics', *Journal of Applied Non-Classical Logics*, 10 (2000), 3-4.

[19] Horrocks, I., and U. Sattler, 'Ontology reasoning in the shoq(d) description logic', in *In Proc. of the 17th Int. Joint Conf. on Artificial Intelligence (IJCAI 2001*, 2001, pp. 199–204.

[20] Immerman, N., *Descriptive Complexity*, Springer-Verlag, New York, Berlin, 1998.

[21] Kachniarz, J., and A. Szałas, 'On some extensions of second-order quantifier techniques', *Unpublished manuscript*, (2001).

[22] Kirousis, L.M., and P.G. Kolaitis, 'A dichotomy in the complexity of propositional circumscription', in *In Proceedings of the 16th Annual IEEE Symposium on Logic in Computer Science – LICS'01*, 2001, pp. 71–80.

[23] Kolaitis, P.G., and C.H. Papadimitriou, 'Some computational aspects of circumscription', *Journal of the ACM*, 37 (1990), 1, 1–14.

[24] Lifschitz, V., 'Computing circumscription', in *Readings in Nonmonotonic Reasoning*, Morgan Kaufmann, 1985, pp. 121–127.

[25] Lifschitz, V., 'On the satisfiability of circumscription', *Artificial Intelligence J.*, 28 (1986), 17–27.

[26] Lifschitz, V., 'Circumscription', in D.M. Gabbay, C.J. Hogger, and J.A. Robinson, (eds.), *Handbook of Artificial Intelligence and Logic Programming*, vol. 3, Oxford University Press, 1991, pp. 297–352.

[27] McCarthy, J., 'Circumscription: A form of non-monotonic reasoning', *Artificial Intelligence J.*, 13 (1980), 27–39.

[28] Nonnengart, A., and A. Szałas, 'A fixpoint approach to second-order quantifier elimination with applications to correspondence theory', in E. Orłowska, (ed.), *Logic at Work: Essays Dedicated to the Memory of Helena Rasiowa*, vol. 24 of *Studies in Fuzziness and Soft Computing*, Springer Physica-Verlag, 1998, pp. 307–328.

[29] Pawlak, Z., *Rough Sets. Theoretical Aspects of Reasoning about Data*, Kluwer Academic Publishers, Dordrecht, 1991.

[30] Phan, M.D., 'On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games', *Artificial Intelligence*, 77 (1995), 321–357.

[31] Schaefer, T.J., 'The complexity of satisfiability problems', in *STOC '78: Proceedings of the 10th Annual ACM symposium on Theory of Computing*, 1978, pp. 216–226.

[32] Selman, B., and H. Kautz, 'Knowledge compilation using horn approximations', in *In Proceedings of AAAI-91*, 1991, pp. 904–909.

[33] Selman, B., and H. Kautz, 'Knowledge compilation and theory approximation', *Journal of the ACM*, 43 (1996), 193–224.

[34] Szałas, A., 'On the correspondence between modal and classical logic: An automated approach', *Journal of Logic and Computation*, 3 (1993), 605–620.

[35] van Benthem, J., 'Minimal predicates, fixed-points, and definability', *Journal of Symbolic Logic*, 70 (2005), 3, 696–712.

[36] Wu, Yining, M. Caminada, and D.M. Gabbay, 'Complete extensions in argumentation coincide with 3-valued stable models in logic programming', *Studia Logica*, 93 (2009), 2-3, 383–403, this issue.

Dov M. Gabbay
Department of Computer Science
King's College, London, UK
Bar-Ilan University, Israel
University of Luxembourg
dov.gabbay@kcl.ac.uk

Andrzej Szałas
Institute of Informatics
Warsaw University, Poland
Dept. of Comp. and Information Sci.
University of Linköping, Sweden
andsz@mimuw.edu.pl