Dominance Constraints in Stratified Context Unification

Katrin Erk, Joachim Niehren

Dominance Constraints in Stratified Context Unification

Katrin Erk[2] Joachim Niehren[1]

[1] *INRIA Futurs at Lille, France, Mostrare Project*
[2] *University of Texas at Austin, Linguistics Department*

Abstract

We express dominance constraints in the once-only nesting fragment of stratified context unification, which therefore is NP-complete.

1 Introduction

In this paper we express dominance constraints [15,19,4,5,1] in stratified context unification (CU) [20,13]. This yields a new small NP-hard fragment of stratified CU and shows the potential of stratified CU for use in applications where dominance constraints alone are not expressive enough.

Dominance constraints are positive conjunctive first-order (FO) logic formulas for finite trees that talk about the descendant relation between nodes. Dominance constraints have first been introduced in computational linguistics [15], where they found numerous application in the modeling of natural language syntax, discourse, and semantics [19,4,5].

In the context of XML database theory, dominance constraints are called conjunctive queries in trees [8], in which they select n-tuples of nodes. Conjunctive and FO queries in trees received considerable interest in the context of XPath, the W3C standard query language for XML node selection [7,16]. The link to FO logic becomes even closer with the introduction of variables in XPath 2.0 [6].

More expressive second-order (SO) logics for finite trees have been studied since the sixties, motivated by the close relationship to tree automata [23,3]. A well-studied language is the weak monadic SO logic with 2 successors (WS2S). Monadic SO logic has equally found much recent interest in the context of information extraction from XML documents [7,11].

Context unification is the satisfiability problem of context equations for trees [2]. It is the natural extension of string unification [14] to trees. It can also be seen as

a form of linear second-order unification [10]. Whether context unification is decidable is a prominent open question. Only some fragments of CU could be shown decidable so far [2,20,21,12,14], among them *stratified CU* [20]. It is of particular interest, since it can express one-step rewriting constraints [18] for which decidability could not be shown by more direct methods. Quite recently, it came as a surprise that stratified CU can be solved in NP and is hence NP-complete [13].

Dominance constraints are known to be expressible by CU equations [17]. Their satisfiability problem is NP-complete [9]. Whether dominance constraints can be expressed in any of the decidable fragments of CU was left open. In this paper, we introduce the *once-only SO-nesting fragment of stratified CU*, and show how to express dominance constraints there. Since satisfiability of dominance constraints is NP-hard [9], it follows that the once-only SO-nesting fragment of stratified CU is NP-complete. We also characterize the precise part of the once-only SO-nesting fragment that is equivalent to dominance constraints.

2 Dominance Constraints

Let Σ be ranked signature of *function symbols*. For simplicity, we assume that Σ contains only binary function symbols f, g and constants a, b. All results presented hold for arbitrary ranked signatures, too. It is sufficient to assume that Σ contains at least one constant and one function symbol of an arity ≥ 2.

A tree $\tau \in T_\Sigma$ is a ground term over Σ. We define a node π of τ to be a word over strictly positive natural numbers, identifying a node with the path leading to it from the root. The root node corresponds to the empty word ϵ. We write the concatenation of two words as $\pi \cdot \pi'$. The set of all nodes of a tree is defined by $\mathit{nodes}(f(\tau_1, \tau_2)) = \{\varepsilon\} \cup \{1 \cdot \pi \mid \pi \in \mathit{nodes}(\tau_1)\} \cup \{2 \cdot \pi \mid \pi \in \mathit{nodes}(\tau_2)\}$ and $\mathit{nodes}(a) = \{\epsilon\}$. Every tree τ has a *labeling function* $L_\tau : \mathit{nodes}(\tau) \to \Sigma$ given by $L_a(\varepsilon) = a$, $L_{f(\tau_1,\tau_2)}(\varepsilon) = f$, $L_{f(\tau_1,\tau_2)}(1 \cdot \pi) = L_{\tau_1}(\pi)$ and $L_{f(\tau_1,\tau_2)}(2 \cdot \pi) = L_{\tau_2}(\pi)$.

We freely identify a tree τ with a logical structure with domain $\mathit{nodes}(\tau)$. This structure provides the dominance relation so that $\tau \models \pi \triangleleft^* \pi'$ iff π is a prefix of π', i.e., if node π is an ancestor of or equal to π'. There are ternary labeling relations for all binary function symbols $f \in \Sigma$ so that $\tau \models \pi{:}f(\pi_1, \pi_2)$ is valid iff $L_\tau(\pi) = f$, $\pi_1 = \pi \cdot 1$ and $\pi_2 = \pi \cdot 2$. And finally, there are unary labeling relations for all constants $a \in \Sigma$ so that $\tau \models \pi{:}a$ iff $L_\tau(\pi) = a$.

For defining dominance constraints over Σ, we assume an infinite set of node variables X, Y, Z. A *dominance constraint* φ over Σ is a conjunction of dominance and labeling literals, where $a, f \in \Sigma$:

$$\varphi ::= X \triangleleft^* Y \mid X{:}f(X_1, X_2) \mid X{:}a \mid \varphi \wedge \varphi'$$

A (FO formula over) dominance constraints φ is interpreted over tree structures. A *model* of φ consists of a tree structure τ and a variable assignment α of variables to

$$X{:}f(X_1, X_2) \wedge X_1 \triangleleft^* Y \wedge X_2 \triangleleft^* Y$$

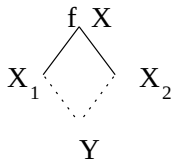

Fig. 1. An unsatisfiable dominance constraint

nodes(τ) which satisfy φ in the usual Tarskian way. We write $\tau, \alpha \models \varphi$ if the pair (τ, α) is a model of φ and say that φ is satisfiable.

The dominance constraint $X{:}f(Y, Y)$ is an example of an unsatisfiable constraint, since brothers are always distinct. Similarly, the constraint in Fig. 1 is unsatisfiable, since siblings in a tree cannot have common descendants. The conflict at variable Y becomes obvious when we draw the constraint as a digraph whose nodes are the variables of the constraint. The variables in labeling literals are connected by solid edges (representing fatherhood) and those in dominance literals by dotted edges (standing for ancesterhood).

Proposition 1 *Satisfiability of dominance constraints is NP-complete.*

This was shown by Koller, Niehren, and Treinen [9]. The difficult part is the NP-hardness proof. Membership to NP is easy. It is sufficient to guess one of the relations *equal*, *above*, *below* and *beside* for each pair of variables in the constraint and to test for consistency.

3 Stratified Context Unification

Context unification talks about relations between trees and relations between contexts. The notion of nodes in trees is hidden in the notion of contexts.

A context over Σ is a function $\gamma : T_\Sigma \to T_\Sigma$ of the form $\gamma = \lambda x.t$, where t is a ground term over $\Sigma \cup \{x\}$ with a single occurrence of the *hole marker* x. The *hole* of γ is the unique node of t that is labeled by x. Function application fills the hole. It replaces the hole marker by the argument that the function receives, i.e. $\gamma(t') = t[t'/x]$ for $t' \in T_\Sigma$.

We assume two sorts of variables, a infinite set of *FO-variables* x, y and a infinite set of *SO-variables* C. Variable assignments σ map FO-variables to trees and SO-variables to contexts. There are two sorts of terms:

$$\begin{array}{ll} \text{FO-terms} & t ::= x \mid a \mid f(t, t') \mid s(t) \\ \text{SO-terms} & s ::= C \mid \bullet \mid f(t, s') \mid f(s', t) \mid s \circ s' \end{array}$$

FO-terms t evaluate to trees $\sigma(t)$ and SO-terms s to contexts $\sigma(s)$ as defined in Fig. 2. The constant $\bullet$ stands for the empty context, i.e., the identity function.

An *FO-equation* is a pair of FO-terms $t{=}t'$ and a *SO-equation* a pair of SO-terms

$$\sigma(a) = a \qquad \sigma(\bullet) = \lambda x.x$$
$$\sigma(f(t,t')) = f(\sigma(t),\sigma(t')) \qquad \sigma(f(t,s)) = \lambda x.f(\sigma(t),\sigma(s)(x))$$
$$\sigma(s(t)) = \sigma(s)(\sigma(t)) \qquad \sigma(f(s,t)) = \lambda x.f(\sigma(s)(x),\sigma(t))$$
$$\sigma(s \circ s') = \sigma(s) \circ \sigma(s')$$

Fig. 2. Evaluation of FO-terms to trees $\sigma(t)$ and SO-terms to contexts $\sigma(s)$

$s{=}s'$. A *CU-equation system* E is a finite conjunction of FO and SO-equations. Whether a variable assignment σ satisfies an (FO-formula over an) equation system E is defined as usual. We write $\sigma \models E$ in this case and call σ a model of E. CU for Σ is the satisfiability problem of CU-equation systems over Σ.

Note that SO-equations are not needed from the view point of expressiveness. They can be defined by FO equations as follows, and thus be eliminated:

$$s{=}s' \models\!\!\models s(a){=}s'(a) \wedge s(b){=}s'(b) \qquad (*)$$

Here, a and b are two different ground terms over Σ which exist by our assumption in Sec. 2. Furthermore, we can always reduce complex applications $s(t)$ in FO-terms to more basic applications of context variables or function symbols. It is sufficient to rewrite exhaustively using the following universally valid equations:

$$\bullet(t) = t \qquad f(t',s')(t) = f(t',s'(t))$$
$$f(s',t')(t) = f(s'(t),t') \qquad (s \circ s')(t) = s(s'(t))$$

The richer syntax with SO-equations is advantageous for defining the *once-only SO-nesting fragment* of CU that we introduce here. We call a variable *SO-nested in a term* if it occurs in the argument of some SO-variable C, or on the right of some C with respect to function composition $\circ$. For instance, x and C' are SO-nested in $f(C(C'(x)),y)$ and $f(y,\bullet) \circ C \circ f(\bullet,x) \circ C'$ while C and y not. A variable is *SO-nested in an equation* if it is SO-nested in one of its sides.

Definition 1 *An equation system E belongs to the* once-only SO-nesting fragment of CU *if all variables that are SO-nested in E occur at most once in E.*

The elimination procedure for SO-equations based on (*) may copy SO-nested variables, leaving the once-only SO-nesting fragment. For instance, $C = C' \circ f(x,\bullet)$ becomes $C(a){=}C'(f(x,a)) \wedge C(b){=}C'(f(x,b))$ where the SO-nested variable x has been copied. This is why SO-equations are crucial for defining the fragment.

Stratified CU is the restriction of CU to stratified CU-equations [20,13]. We extend the usual setting by SO-equations, in order to account for once-only SO-nesting.

Definition 2 *The* SO-prefix *of a node π in a FO or SO-term is the word of SO-variables, seen on the way from the root down to π properly before π. A system of CU-equations E (FO or SO) is* stratified *if all its variables (FO and SO) have at*

most one SO-prefix in all terms on left and right hand sides of equations in E.

The equations $C = C' \circ C$ and $C(x) = C'(C(y))$ are not stratified since the occurrence of C on the left has SO-prefix ϵ, while that on the right has SO-prefix C'. The equation $x = f(x)$ is stratified, while $x = C(x)$ is not.

Theorem 1 (Levy, Schmidt-Schauß, Villaret 2006) *Stratified CU is NP-complete.*

For FO-equations, this results was shown in [13]. It extends to SO-equations since they can be eliminated while preserving stratification. Next note that once-only SO-nesting is subsumed by stratification: once-only variables have a unique SO-prefix, so they cannot violate stratification. All other variables are not SO-nested so they have the unique SO-prefix ϵ.

Corollary 1 *The once-only SO-nesting fragment of CU is in NP.*

This follows from Theorem 1 and the fact that once-only SO-nesting implies stratification. Note that the NP-hardness result for stratified CU from [22] does not carry over. That it holds nevertheless will follow from the relation to dominance constraints.

4 Expressing Dominance Constraints

We now show how to express dominance constraints in the once-only SO-nesting fragment of CU. The main problem is that the two languages describe trees from different perspectives. Dominance constraints model relations between the *nodes* of the same tree, while CU-equations talk about relations between *trees* and *contexts*. While it is possible that a tree contains two structurally equal subtrees, is impossible that two siblings have the same descendants. Hence, the FO-equation $x{=}f(y,y)$ is satisfiable, while the dominance constraint $X{:}f(Y,Y)$ is not.

The idea for the encoding is that contexts can locate nodes in trees. This ideas was already used in [17] for encoding dominance constraints in CU, but unfortunately, not in a stratified manner. In order to overcome that problem, one has to limit the usage of first-order variables. This means that nodes should be identified by their contexts exclusively, without talking about the subtree that starts at them.

The translation $[\![\varphi]\!]$ is given in Fig. 3. Every node variable X is associated with a fresh context variable C_X that denotes the context of node X in tree models of φ, starting from the root. The whole tree is modeled by the FO-variable x_{root}. In order to force all contexts C_X to start at the root of the tree, the translation imposes $x_{\mathit{root}} = C_X(\_)$ for all variables X in ϕ. The underscore $\_$ stands for fresh existentially quantified FO-variables. The pretranslation $[\![\varphi]\!]_p$ expresses conditions for all literals of φ. Literals $X{\lhd^*}Y$ state that the context of Y extends that of X somehow: $\exists C\, C_Y = C_X \circ C$. Literals $X{:}f(X_1, X_2)$ express the fact that the context

$$
\begin{array}{rcl}
[\![\varphi]\!] & = & [\![\varphi]\!]_p \wedge \bigwedge_{X \in \mathcal{V}(\varphi)} \exists x\; x_{\mathit{root}} {=} C_X(x) \\
[\![X \triangleleft^* Y]\!]_p & = & \exists C\; C_Y {=} C_X \circ C \\
[\![X{:}f(X_1, X_2)]\!]_p & = & \exists x_2\; (C_{X_1} {=} C_X \circ f(\bullet, x_2)) \;\wedge\; \exists x_1\; (C_{X_2} {=} C_X \circ f(x_1, \bullet)) \\
[\![X{:}a]\!]_p & = & x_{\mathit{root}} {=} C_X(a) \\
[\![\varphi_1 \wedge \varphi_2]\!]_p & = & [\![\varphi_1]\!]_p \wedge [\![\varphi_2]\!]_p
\end{array}
$$

Fig. 3. Encoding dominance constraints

of X_1 extends that of X by $f(\bullet, \_)$ and that the context of X_2 extends that of X by $f(\_, \bullet)$. Hence $C_{X_1} = C_X \circ f(\bullet, \_)$ and $C_{X_2} = C_X \circ f(\_, \bullet)$. Finally, leaf labeling constraints $X{:}a$ state that the context of X is to be filled by a, i.e. $x_{\mathit{root}} = C_X(a)$.

As a first example, we consider the unsatisfiable constraint $Y{:}f(X, X)$. Its pretranslation $C_X{=}C_Y \circ f(\bullet, \_) \wedge C_X{=}C_Y \circ f(\_, \bullet)$ is unsatisfiable, since it contains two equations for C_X which yield $C_Y \circ f(\bullet, \_) = C_Y \circ f(\_, \bullet)$ and thus $f(\bullet, \_) = f(\_, \bullet)$, which is impossible. The translation of the constraint in Fig. 1 can be shown unsatisfiable in the same manner.

Next we examine the unsatisfiable constraint $X{:}f(Y) \wedge X{:}g(Z)$ with $f \neq g$, whose pretranslation $C_Y = C_X \circ f(\bullet) \wedge C_Z = C_X \circ g(\bullet)$ is satisfiable. Joined with the rootedness condition of the full translation, we obtain $x_{\mathit{root}} = C_Y(\_) = C_Z(\_)$, so that $C_X(f(\_)) = C_X(g(\_))$ and thus $f = g$, which is impossible.

In the third example, we consider a case of unsatisfiable leaf labeling: $X{:}a \wedge Y{:}b \wedge X \triangleleft^* Y$ where $a \neq b$. Its pretranslation $x_{\mathit{root}} = C_X(a) \wedge x_{\mathit{root}} = C_Y(b) \wedge C_Y = C_X \circ C$ is unsatisfiable, since the two equations for x_{root} yield $C_X(a) = C_Y(b)$, thus $C_X(a) = C_X(C(b))$, so that $a = C(b)$ which is impossible.

Proposition 2 (Correctness) *Models of φ are in bijection to models of $[\![\varphi]\!]$.*

The proof is very similar to that of [9]. It is elaborated in Section 5.

Proposition 3 (Stratification) *The translation of dominance constraints maps to the once-only SO-nesting fragment of stratified CU.*

Proof: The SO-nested variables introduced in $[\![\varphi]\!]$ are exactly those that are existentially quantified. It is easy to check that these variables are used only once in the scope of the quantifier. □

Theorem 2 *Satisfiability of dominance constraints over Σ can be encoded in linear time into the once-only SO-nesting fragment of stratified context unification over Σ.*

Proof: The translation $[\![\ .\ ]\!]$ is clearly in linear time. It preserves satisfiability by Proposition 2 and maps to the once-only SO-nesting fragment of stratified CU by Proposition 3. □

$$[\![x{=}C(y)]\!]^{-1} = X_C \triangleleft^* X_C \qquad [\![C{=}C' \circ f(\bullet, x)]\!]^{-1} = X_{C'}{:}f(X_C, \_)$$
$$[\![x{=}C(a)]\!]^{-1} = X_C{:}a \qquad [\![C{=}C' \circ f(x, \bullet)]\!]^{-1} = X_{C'}{:}f(\_, X_C)$$
$$[\![C{=}C_1 \circ C_2]\!]^{-1} = X_{C_1} \triangleleft^* X_C \qquad [\![E \wedge E']\!]^{-1} = [\![E]\!]^{-1} \wedge [\![E']\!]^{-1}$$

Fig. 4. Back translation of once-only SO nesting rooted CU equations with simple literals.

Corollary 2 *The once-only SO-nesting fragment of stratified CU is NP-complete.*

Proof: This follows from Theorem 2 and the NP-hardness of dominance constraint satisfiability [17]. □

We next characterize the range of the translation precisely, in order to distinguish an even smaller NP-hard fragment of stratified CU. We impose the following conditions beside once-only SO-nesting:

Rootedness: there exists a unique FO-variable x_{root} in E that is not SO-nested. All non-nested SO-variables x in E are rooted at x_{root}, i.e., $x_{root} = C(x)$ belongs to E for some FO-variable x.

Simple literals: all literals have one of the following forms:
$x{=}C(y)$, $x{=}C(a)$, $C{=}C_1 \circ C_2$, $C{=}C' \circ f(\bullet, x)$, or $C{=}C' \circ f(x, \bullet)$.

All CU-equations $[\![\varphi]\!]$ obtained from dominance constraints satisfy these conditions. Conversely, we can back-translate CU-equations E with these properties to dominance constraints $[\![E]\!]^{-1}$. For all SO-variables C in E that are not SO-nested we introduce a node variable X_C and define the translation $[\![E]\!]^{-1}$ in Fig. 4. Note that the x in the first equation has to be x_{root} by the rootedness constraint. For the set $N(E)$ of variables nested in E it obviously holds that $\exists N(E).E \models\!\models [\![[\![E]\!]^{-1}]\!]$.

5 Correctness Proof

We show the correctness of the translation as stated in Proposition 2. We have to show that models $\tau, \alpha \models \varphi$ correspond one-to-one and onto to models $\sigma \models [\![\varphi]\!]$. To that end, we translate pairs (τ, α) into substitutions σ.

Let π_1 and π_2 be nodes of τ such that π_1 dominates π_2. We write $\tau.\pi_1$ for the subtree of τ rooted at π_1, which is an FO-term, and $\tau^{\pi_1}_{\pi_2}$ for the SO-term in τ starting at π_1 and ending at π_2, i.e., the SO-term obtained by first replacing the subtree of τ at π_2 by $\bullet$ and then selecting the subtree at π_1.

We fix a mapping of node variables X to SO-variables C_X that is one-to-one, and we fix a single FO-variable x_{root}. We define a variable assignment $[\![\tau, \alpha]\!]$ such that it satisfies for all node variables X:

$$[\![\tau, \alpha]\!](C_X) = \tau^{\epsilon}_{\alpha(X)} \qquad \text{and} \qquad [\![x_{root}]\!] = \tau$$

It is easy to see that $[\![\tau, \alpha]\!] = [\![\tau', \alpha']\!]$ if and only if $\tau = \tau'$ and $\alpha = \alpha'$. Thus, the translations is one-to-one. To see that it is onto, we have to identify groups of models that coincide on all variables of the equation system under consideration.

Lemma 1 *All models $\sigma \models [\![\varphi]\!]$ are equal to some translation $[\![\tau, \alpha]\!]$, under the assumption that model equality is limited to the variables occurring in $[\![\varphi]\!]$.*

Proof: Let V be the set of variables in the constraint φ. All variables $X \in V$ satisfy $\sigma \models x_{root}{=}C_X(\_)$. We define the tree τ by $\tau = \sigma(x_{root})$ and the variable assignment α into τ such that $\alpha(X)$ is the hole of $\sigma(C_X)$. It then holds that $[\![\tau, \alpha]\!](C_X) = \sigma(C_X)$ for all variables $X \in V$ and that $[\![\tau, \alpha]\!](x_{root}) = \sigma(x_{root})$. Hence, $[\![\tau, \alpha]\!] = \sigma$ with respect to model equality that is restricted to the variables occurring in $[\![\varphi]\!]$. □

It remains to show that the translation of dominance constraints preserves the models modulo model translation.

Lemma 2 *If $\tau, \alpha \models \varphi$ then $[\![\tau, \alpha]\!] \models [\![\varphi]\!]$.*

Proof: Let $\sigma = [\![\tau, \alpha]\!]$ and assume $\tau, \alpha \models \varphi$. The rootedness conditions $\sigma \models x_{root}{=}C_X(\_)$ holds for all X since $\sigma(x_{root}) = \sigma(C_X)(\tau.\alpha(X))$. It remains to show that $\sigma \models [\![\varphi]\!]_p$, by induction on the structure of formula φ.

- Case φ is $X \triangleleft^* Y$. We have to show that $\sigma \models \exists C.\ C_Y{=}C_X \circ C$. The assumption $\tau, \alpha \models \varphi$ means that $\alpha(X)$ is a prefix of $\alpha(Y)$. Hence $\tau^{\alpha(X)}_{\alpha(Y)}$ is well-defined and satisfies $\sigma(C_Y) = \sigma(C_X) \circ \tau^{\alpha(X)}_{\alpha(Y)}$.
- Case φ is $X{:}f(X_1, X_2)$. We need to show that $\sigma \models \exists x_2.C_{X_1}{=}C_X \circ f(\bullet, x_2)$ and $\sigma \models \exists x_1.C_{X_2}{=}C_X \circ f(x_1, \bullet)$. By symmetry it is sufficient to prove the former. It follows from $\sigma(C_{X_1}) = \sigma(C_X) \circ f(\bullet, \tau.\alpha(Y))$.
- Case φ is $X{:}a$. Since $\tau, \alpha \models \varphi$, it holds that $\alpha(X)$ is a leaf labeled by a in τ. Hence, $\sigma(x_{root}) = \tau = \sigma(C_X)(a)$, so that $\sigma \models x_{root}{=}C_X(a)$.

□

Lemma 3 *If $[\![\tau, \alpha]\!] \models [\![\varphi]\!]$ then $\tau, \alpha \models \varphi$.*

Proof: Again, let $\sigma = [\![\tau, \alpha]\!]$ and assume $\sigma \models [\![\varphi]\!]$. The proof is by induction on the structure of φ.

- Case φ is $X \triangleleft^* Y$. By $\sigma \models [\![\varphi]\!]_p$ there is some context γ such that $\tau^{\epsilon}_{\alpha(Y)} = \tau^{\epsilon}_{\alpha(X)} \circ \gamma$. Thus, $\alpha(X)$ is a prefix of $\alpha(Y)$ so that $\tau, \alpha \models X \triangleleft^* Y$.
- Case φ is $X{:}f(X_1, X_2)$. Since $\sigma \models [\![\varphi]\!]_p$ there exist trees τ_2 and τ_1 such that $\tau^{\epsilon}_{\alpha(X_1)} = \tau^{\epsilon}_{\alpha(X)} \circ \lambda x.f(x, \tau_2)$ and $\tau^{\epsilon}_{\alpha(X_2)} = \tau^{\epsilon}_{\alpha(X)} \circ \lambda x.f(\tau_1, x)$. Hence, $\alpha(X_1) = \alpha(X).1$, $\alpha(X_2) = \alpha(X).2$ and $L_\tau(\alpha(X)) = f$. This is equivalent to $\tau, \alpha \models X{:}f(X_1, X_2)$.
- Case φ is $X{:}a$. Since $\sigma \models [\![\varphi]\!]$, it holds that $\tau = \tau^{\epsilon}_{\alpha(X)}(a)$. This is equivalent to $L_\tau(X) = a$, and thus to $\tau, \alpha \models X{:}a$.

□

6 Conclusion and Future Work

We have shown that dominance constraints can be expressed in the once-only SO-nesting fragment of stratified CU. This classifies dominance constraints within the decidable fragments of CU, and it yields a new small fragment of stratified CU that is NP-hard, the once-only SO-nesting fragment. In addition, it shows that stratified CU is sufficiently expressive for many applications to natural language semantics, including the modeling of scope underspecification.

So far, however, it remains unclear whether the additional expressiveness of stratified CU can help to improve modeling natural language phenomena. An answer exists in the case of parallelism [5], as stratified CU lacks sufficient expressiveness for modeling VP-ellipsis in natural language semantics. This is in contrast to well-nested CU [12], another NP-complete fragment of CU, which however subsumes only a fragment of dominance constraints (where minimal solutions can be built without guessing new material absent in the constraint).

A drawback of stratified CU so far is that it is lacking efficient polynomial time algorithms for solving constraints in relevant fragments. This is in contrast to *normal dominance constraints*, a fragment of dominance constraints for which satisfiability is in linear time [1]. So an interesting question for future work is to distinguish efficient fragments of stratified CU which allow for polynomial time algorithms. A more general open question is whether there exist polynomial time fragments of CU which are sufficient for modeling relevant classes of VP-ellipsis.

Acknowledgements. Many thanks to Manfred Schmidt-Schauß for helpful discussion, and for inducing us to this result.

References

[1] Manuel Bodirsky, Denys Duchier, Sebastian Miele, and Joachim Niehren. A new algorithm for normal dominance constraints. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 54–78, 2004.

[2] H. Comon. Completion of rewrite systems with membership constraints. In *International Colloquium on Automata, Languages and Programming*, pages 392–403, 1992.

[3] H. Comon, M. Dauchet, R. Gilleron, F. Jacquemard, D. Lugiez, S. Tison, and M. Tommasi. Tree automata techniques and applications. Available online at `www.grappa.univ-lille3.fr/tata`, 1997.

[4] D. Duchier and C. Gardent. Tree descriptions, constraints and incrementality. In *Computing Meaning 2*, vol. 77 of *Linguistics and Philosophy*, pages 205–227. 2001.

[5] M. Egg, A. Koller, and J. Niehren. The constraint language for lambda structures. *Journal of Logic, Language and Information*, 10:457–485, 2001.

[6] E. Filiot, J. Niehren, J.-M. Talbot, and S. Tison. On XPath dialects with variables. 2006.

[7] G. Gottlob and C. Koch. Monadic queries over tree-structured data. In *17th Annual IEEE Symposium on Logic in Computer Science*, pages 189–202, 2002.

[8] G. Gottlob, C. Koch, and K. U. Schulz. Conjunctive queries over trees. In *23rd ACM Symposium on Principles of Database Systems*, pages 189–200, 2004.

[9] A. Koller, J. Niehren, and R. Treinen. Dominance constraints: Algorithms and complexity. In *3rd International Conference on Logical Aspects of Computational Linguistics (1998)*, volume 2014 of *LNCS*, pages 106–125, 2001.

[10] J. Levy. Linear second order unification. In *7th Int. Conference on Rewriting Techniques and Applications*, volume 1103 of *LNCS*, pages 332–346, 1996.

[11] L. Libkin. Logics over unranked trees: an overview. *Logical Methods in Computer Science*, 3(2), pages 1-31, 2006.

[12] J. Levy, J. Niehren, and M. Villaret. Well-nested context unification. In *20th Intern. Conference on Automated Deduction*, volume 3632 of *LNAI*, pages 149–163. 2005.

[13] J. Levy, M. Schmidt-Schauss, and M. Villaret. Stratified context unification is NP-complete. In *3rd International Joint Conference on Automated Reasoning*, volume 4130 of *LNCS*, pages 82–96. 2006.

[14] G. Makanin. The problem of solvability of equations in a free semigroup. *Soviet Akad. Nauk SSSR*, 223(2), 1977.

[15] M. P. Marcus, D. Hindle, and M. M. Fleck. D-theory: Talking about talking about trees. In *21st Meeting of the Ass. of Computational Linguistics*, pages 129–136, 1983.

[16] M. Marx. Conditional XPath. *ACM Transactions on Database Systems*, 30(4):929–959, 2005.

[17] J. Niehren and A. Koller. Dominance constraints in context unification. In *3rd International Conferernce on Logical Aspects of Computational Linguistics (1998)*, volume 2014 of *LNCS*, pages 199–218, 2001.

[18] J. Niehren, R. Treinen, and S. Tison. On rewrite constraints and context unification. *Information Processing Letters*, 74(1-2):35–40, Apr. 2000.

[19] J. Rogers and K. Vijay-Shanker. Extracting trees from their descriptions: an application to tree-adjoining grammars. *Computational Intelligence*, 10(4):401–421, 1994.

[20] M. Schmidt-Schauss. A decision algorithm for stratified context unification. *Journal of Logic and Computation*, 12(6):929–953, 2002.

[21] M. Schmidt-Schauss and K. U. Schulz. Solvability of context equations with two context variables is decidable. *Journal of Symbolic Computation*, 33(1):77–122, 2002.

[22] M. Schmidt-Schauß and K. U. Schulz. On the exponent of periodicity of minimal solutions of context equation. In *9th RTA*, volume 1379 of *LNCS*, pages 61–75, 1998.

[23] J. W. Thatcher and J. B. Wright. Generalized finite automata with an application to a decision problem of second-order logic. *Mathematical System Theory*, 2:57–82, 1968.