Expanding the Range of Tractable Scope-Underspecified Semantic Representations

Mehdi Manshadi and James Allen
Department of Computer Science
University of Rochester
Rochester, NY 14627
{mehdih, james}@cs.rochester.edu

Abstract

Over the past decade, several underspecification frameworks have been proposed that efficiently solve a big subset of scope-underspecified semantic representations within the realm of the most popular constraint-based formalisms. However, there exists a family of coherent natural language sentences whose underspecified representation does not belong to this subset. It has remained an open question whether there exists a tractable superset of these frameworks, covering this family. In this paper, we show that the answer to this question is yes. We define a superset of the previous frameworks, which is solvable by similar algorithms with the same time and space complexity.

1 Introduction

Scope ambiguity is a major source of ambiguity in semantic representation. For example, the sentence

1. *Every politician has a website.*

has at least two possible interpretations, one in which each *politician* may have a different *website* (i.e., *Every* has **wide scope**) and one in which there is a unique *website* for all the *politicians* (i.e., *Every* has **narrow scope**). Since finding the most preferred reading automatically is very hard, the most widely adopted solution is to use an **Underspecified Representation** (**UR**), that is to encode the ambiguity in the semantic representation and leave scoping **underspecified**.

In an early effort, Woods (1986) developed an unscoped logical form where the above sentence is represented (roughly) as the formula:

2. $Has(\langle Every\ x\ Politician\rangle, \langle A\ y\ Website\rangle)$

To obtain a fully scoped formula, the quantifiers are pulled out one by one and wrapped around the formula. If we pull out *Every* first, we produce the fully-scoped formula:

3. $A(y, Website(y),$
   $Every(x, Politician(x), Has(x, y))$

If we had pulled out *A* first, we would have had the other reading, with *Every* having wide scope.

Hobbs and Shieber (1987) extend this formalism to support operators (such as *not*) and present an enumeration algorithm that is more efficient than the naive wrapping approach.

Since the introduction of Quasi Logical Form (Alshawi and Crouch, 1992), there has been a lot of work on designing constraint-based underspecification formalisms where the readings of a UR are not defined in a constructive fashion as shown above, but rather by a set of constraints. A fully-scoped structure is a reading iff it satisfies all the constraints. The advantage of these frameworks is that as the processing goes deeper, new (say pragmatically-driven) constraints can be added to the representation in order to filter out unwanted readings. **Hole Semantics** (Bos, 1996; Bos, 2002), Constraint Language for Lambda Structures (**CLLS**) (Egg et al., 2001), and Minimal Recursion Semantics (**MRS**) (Copestake et al., 2001) are among these frameworks.

In an effort to bridge the gap between the above formalisms, a graph theoretic model of scope underspecification was defined by Bodirsky et al. (2004), called **Weakly Normal Dominance Graphs**. This

*First Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 142–150,
Montréal, Canada, June 7-8, 2012. ©2012 Association for Computational Linguistics

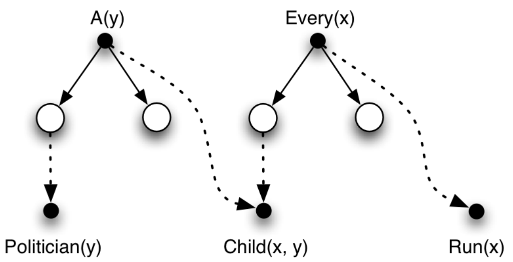

Figure 1: UG for *Every child of a politician runs.*

framework and its ancestor, **Dominance Constraints** (Althaus et al., 2003), are broad frameworks for solving constrained tree structures in general. When it comes to scope underspecification, some of the terminology becomes counter-intuitive. Therefore, here we first define (scope) **Underspecification Graphs** (**UG**), a notational variant of weakly normal dominance graphs, solely defined to model scope underspecification.[1] Figure 1 shows a UG for the following sentence.

4. *Every child of a politician runs.*

The big circles and the dot nodes are usually referred to as the **hole** nodes (or simply holes) and the **label** nodes (or simply labels) respectively. The left and the right holes of each quantifier are placeholders for the **restriction** and the **body** of the quantifier. A fully scoped structure is built by **plugging** labels into holes, as shown in Figure 2(a). The dotted edges represent the **constraints**. For example, the constraint from the restriction hole of *Every(x)* to the node *Politician(x)* states that this label node must be within the scope of the restriction of *Every(x)* in every reading of the sentence. The constraint edge from *Every(x)* to *Run(x)* forces the binding constraint for variable *x*; that is variable *x* in *Run(x)* must be within the scope of its quantifier. Figure 2(b) represents the other possible reading of the sentence. Now consider the sentence:

5. *Every politician, whom I know a child of, probably runs.*

with its UG shown in Figure 3. This sentence contains a scopal adverbial (a.k.a. **fixed-scopal**; cf. Copestake et al. (2005)), the word *Probably*. Since in general, quantifiers can move inside or outside a

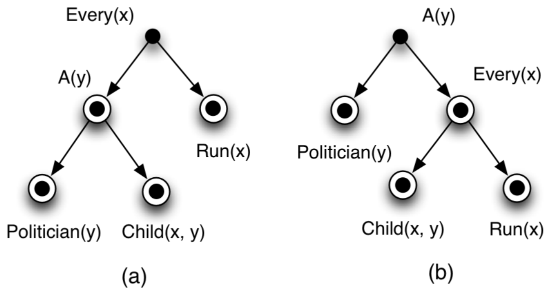

Figure 2: Solutions of the UG in Figure 1.

scopal operator, the scope of *Probably* is left underspecified, and hence represented by a hole. It is easy to verify that the corresponding UG has five possible readings, two of which are shown in Figure 4.

There are at least two major algorithmic problems that need to be solved for any given UG U: the **satisfiability** problem; that is whether there exists any reading satisfying all the constraints in U, and the **enumeration** problem; that is enumerating all the possible readings of a satisfiable U. Unfortunately, both problems are NP-complete for UG in its general form (Althaus et al., 2003). This proves that Hole Semantics and Minimal Recursion Semantics are also intractable in their general form (Thater, 2007). In the last decade, there has been a series of interesting work on finding a tractable subset of those frameworks, broad enough to cover most structures occurring in practice. Those efforts resulted in two closely related **tractable** frameworks: (dominance) **net** and **weak** (dominance) **net**. Intuitively, the net condition requires the following property. Given a UG U, for every label node in U with n holes, if the node together with all its holes is removed from U, the remaining part is composed of at most n (weakly) connected components. A difference between net and weak net is that in nets, label-

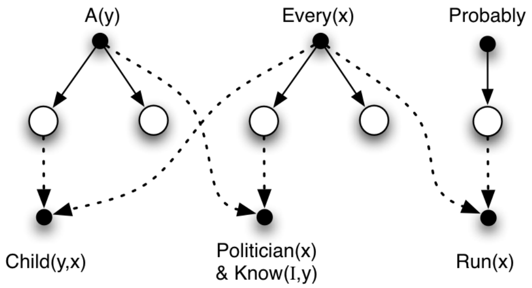

Figure 3: UG for the sentence in (5).

[1] The main difference is in the concept of solution in the two frameworks. See Section 4.3 for details.

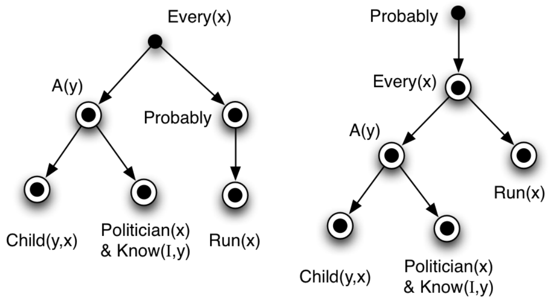

Figure 4: Two of the solutions to the UG in Figure 3.

to-label constraints (e.g. the constraint between *Every(x)* and *Run(x)* in Figure 1) are not allowed.

Using a sample grammar for CLLS, Koller et al. (2003) conjecture that the syntax/semantics interface of CLLS only generates underspecified representations that follow the definition of net and hence can be solved in polynomial time. They also prove that the same efficient algorithms can be used to solve the underspecification structures of Hole Semantics which satisfy the net condition.

Unlike Hole Semantics and CLLS, MRS implicitly carries label-to-label constraints; hence the concept of net could not be applied to MRS. In order to address this, Niehren and Thater (2003) define the notion of weak net and conjecture that it covers all semantically complete MRS structures occurring in practice. Fuchss et al. (2004) supported the claim by investigating MRS structures in the Redwoods corpus (Oepen et al., 2002). Later coherent sentences were found in other corpora or suggested by other researchers (see Section 6.2.2 in Thater (2007)), whose UR violates the net condition, invalidating the conjecture. However, violating the net condition occurs in a similar way in those examples, suggesting a family of non-net structures, characterized in Section 4.2. Since then, it has been an open question whether there exists a tractable superset of weak nets, covering this family of non-net UGs.

In the rest of this paper, we answer this question. We modify the definition of weak net to define a superset of it, which we call **super net**. Super net covers the above mentioned family of non-net structures, yet is solvable by (almost) the same algorithms as those solving weak nets with the same time and space complexity.

The structure of the paper is as follows. We define our framework in Section 2 and present the polynomial-time algorithms for its satisfiability and enumeration problems in Section 3. In Section 4, we compare our framework with nets and weak nets. Section 5 discusses the related work, and Section 6 summarizes this work and discusses future work.

2 Super net

We first give a formal definition of underspecification graph (UG). We then define super net as a subset of UG. In the following definitions, we openly borrow the terminology from Hole Semantics, Dominance Constraints, and MRS, in order to avoid inventing new terms to name old concepts.

Definition 1 (Fragments)**.** *Consider L a set of labels, H a set of holes, and S a set of directed* solid *edges from labels to holes, such that $\mathcal{F} = (L \uplus H, S)$ is a forest of* ordered *trees of depth at most 1, whose root and only the root is a label node. Each of these trees is called a* ***fragment****.*

Following this definition, the number of trees in $\mathcal{F}$ (including single-node trees) equals the number of labels. For example, if we remove all the dotted edges in Figure 1, we obtain a forest of 5 fragments.

Definition 2 (Underspecification Graph)**.** *Let $\mathcal{F} = (L \uplus H, S)$ be a forest of fragments and C be a set of directed* dotted *edges from $L \uplus H$ to L, called the set of* ***constraints****.*[2] *$U = (L \uplus H, S \uplus C)$ is called an* ***underspecification graph*** *or* ***UG****.*

Figures 1 and 3 each represent a UG.

Definition 3 (Plugging)**.** (Bos, 1996) *Given a UG $U = (L \uplus H, S \uplus C)$, a* ***plugging*** *$P$ is a total one-to-one function from H to L.*

In Figure 1, if l_A, l_E, l_P, l_C, and l_R represent the nodes labeled by *A(y)*, *Every(x)*, *Politician(y)*, *Child(x,y)*, and *Run(x)* respectively and h^r_A (h^b_A) and h^r_E (h^b_E) represent the restriction (body) hole of *A* and *Every* respectively, then P in (6) is a plugging.

6. $P = \{(h^r_A, l_P), (h^b_A, l_C), (h^r_E, l_A), (h^b_E, l_R)\}$

We use $\boldsymbol{T_{U,P}}$ to refer to the graph, formed from U by removing all the constraints and plugging $P(h)$ into h for every hole h. For example if U is the UG in Figure 1 and P is the plugging in (6), then $T_{U,P}$ is the graph shown in Figure 2(a).

[2] We assume that there is no constraint edge between two nodes of the same fragment.

Definition 4 (Permissibility/Solution). $T_{U,P}$ ***satisfies*** *the constraint (u,v) in U, iff u* dominates[3] *v in $T_{U,P}$.*[4] *A plugging P is* ***permissible****, iff $T_{U,P}$ is a* forest *satisfying all the constraints in U. $T_{U,P}$ is called a* ***solution*** *of U iff P is a permissible plugging. In informal contexts, solutions are sometimes referred to as* ***readings****.*

It is easy to see that the plugging in (6) is a permissible plugging for the UG in Figure 1, and hence Figure 2(a) is a solution of this UG. Similarly, Figures 4(a,b) represent two solutions of the UG in Figure 3. The solutions in Figures 2 and 4 are all *tree* structures. This is because UGs in Figures 1 and 3 are *weakly connected*.[5] Lemma 2 proves that this holds in general, that is:

Proposition 1. *Every solution of a weakly connected UG is a tree.*

Throughout the rest of this paper, unless otherwise specified, UGs are assumed to be weakly connected, hence solutions are tree structures.[6]

Lemma 2. (Bodirsky et al., 2004) *Given a UG U and a solution T of U, if the nodes u and v in U are connected using an undirected path p, there exists a node w on p such that w dominates both u and v in T.*

This Lemma is proved using induction on the length of p. As mentioned before, *satisfiability* and *enumeration* are two fundamental problems to be solved for a UG. A straightforward approach is depicted in Figure 5. We pick a label l; remove it from U; recursively solve each of the resulting *weakly connected components* (*WCC*s; cf. footnote 2) and plug the root of the returned trees into the corresponding holes of l. A problem to be addressed though is whether there exists any solution rooted at l. This leads us to the following definition.

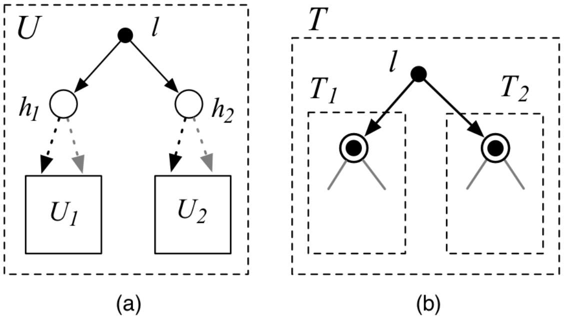

Figure 5: Recursively solving UGs.

Definition 5 (Freeness). (Bodirsky et al., 2004) *A label l in U is called a* ***free node****, iff there exists some solution of U rooted at l. The fragment rooted at l is called a* ***free fragment****.*

The following proposition states the necessary conditions for a label (or fragment) to be free.[7]

Proposition 3. *Let l in U be the root of a fragment F with m holes. l is a free node of U, only if*

P3a. l has no incoming (constraint) edge;

P3b. Every distinct hole of F lies in a distinct WCC in $U-l$;

P3c. $U-F$ consists of at least m WCCs.

Proof. The first condition is trivial. To see why the second condition must hold, let T be a solution rooted at l, and assume to the contrary that h_1 and h_2 lie in the same WCC in $U-l$. From Lemma 2, all the nodes in this WCC must be in the scope of both h_1 and h_2. But this is not possible, because T is a tree. The third condition is proved similarly. Assume to the contrary that $U-F$ has $m-1$ WCCs. From Lemma 2, all the nodes in a WCC must be in the scope of a single hole of F. But there are m holes and only $m-1$ WCCs. It means that one of the holes in T is left unplugged. Contradiction! □

The motivation behind defining super nets is to find a subset of UG for which these conditions are also sufficient. The following concept from Althaus et al. (2003) plays an important role.

[3] u dominates v in the directed graph G, iff u reaches v in G by a $directed$ path.

[4] Here, we are referring to the nodes in $T_{U,P}$ by calling the nodes u and v in U. This is a sound strategy, as every node in U is mapped into a unique node in $T_{U,P}$. The inverse is not true though, as every node (except the root) in $T_{U,P}$ corresponds to one hole and one label in U. Addressing $T_{U,P}$'s nodes in this way is convenient, so we practice that throughout the paper.

[5] Given a directed graph G and the nodes u and v in G, u is said to be *weakly connected* to v (and vice versa), iff u and v are connected in the underlying undirected graph of G. A *weakly connected graph* is a graph in which every two nodes are weakly connected. Since weak connectedness is an equivalence relation, it partitions a directed graph into equivalent classes each of which is called a *weakly connected component* or *WCC*.

[6] Since fragments are ordered trees, solutions are ordered trees as well.

[7] Necessary conditions of freeness in a UG are not exactly the same as the ones in a weakly normal dominance graph, as depicted in Bodirsky et al. (2004), because the definition of solution is different for the two frameworks (c.f. Section 4.3).

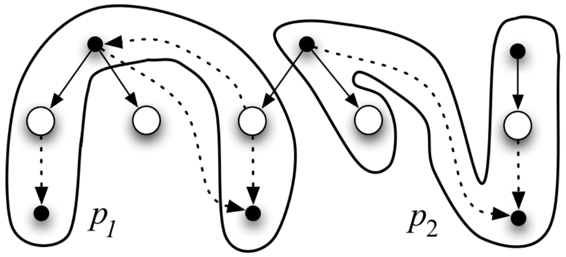

Figure 6: UG for *Illustration of hypernormal path.*

Definition 6 (Hypernormal Connectedness)**.** *Given a UG U, a* ***hypernormal path*** *is an undirected path*[8] *with no two consecutive constraint edges emanating from the same node. Node u is* ***hypernormally connected*** *to node v iff there is at least one hypernormal path between the two. U is called* ***hypernormally connected*** *iff every pair of nodes in U are hypernormally connected.*

For example, in Figure 2, p_2 is a hypernormal path, but p_1 is not. In spite of that, the whole graph is hypernormally connected.[9] The following simple notion will also come handy.

Definition 7 (Openness)**.** (Thater, 2007) *A node u of a fragment F is called an* ***open node*** *iff it has no outgoing constraint edge.*

For example, l in Figure 5(a) is an open label node. In Figure 2(b), h_2 is an open hole. We are finally ready to define super net.

Definition 8 (Super net)**.** *A UG U is called a* ***super net*** *if for every fragment F rooted at l:*

D8a. F has at most one open node.

D8b. If l_1 and l_2 are two dominance children of a hole h of F, then l_1 and l_2 are hypernormally connected in $U-h$.

D8c.
- *Case 1: F has no open hole.*
  Every dominance child[10] *of l is hypernormally connected to some hole of F in U–l.*
- *Case 2: F has an open hole.*
  All dominance children of l, not hypernormally connected to a hole of F in U–l, are hypernormally connected together.

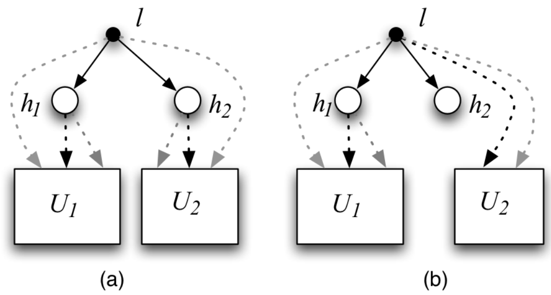

Figure 7: Illustration of super net conditions.

Definition 9 (Types of fragment)**.** *Following Definition 8, super net allows for three possible types of fragment:*

D9a. ***Open-root****: Only the root is open (Figure 5a)*

D9b. ***Open-hole****: Only a hole is open (Figure 2b)*

D9c. ***Closed****: F There is no open node. (Figure 2a)*

Definition 8 guarantees the following property.

Lemma 4. *For a super net U and a fragment F of U with m holes, which satisfies the conditions in Proposition 3, $U-F$ consists of exactly m WCCs, each of which is a super net.*

Proof sketch. The detailed proof of this Lemma is long. Therefore, we sketch the proof here and leave the details for a longer paper. First, we show that $U-F$ consists of exactly m WCCs. Following conditions (D8b) and (D8c), no matter what structure F has, U–F consists of at most m WCCs. On the other hand, based on condition (P3c), $U-F$ has at least m WCCs. Therefore, $U-F$ has exactly m WCCs. To prove that each WCC in $U-F$ is a super net, all we need to prove is that if two nodes u and v, which do not belong to F, are hypernormally connected in U, they are also hypernormally connected in U–F. This is proved by showing that there is no hypernormal path between u and v in U that visits some node of F. Suppose that F is an open-hole fragment rooted at l, as in Figure 2(b) (the two other cases are proved similarly) and assume to the contrary that there is a hypernormal path p between u and v that visits some node of F. One of the following three cases holds.

i. p visits exactly one node of F.
ii. p visits (at least) two holes of F.
iii. p visits l and exactly one hole of F.

All the three cases results in a contradiction: (i) proves that p is not hypernormal; (ii) proves that F

[8] Throughout this paper, by path we always mean a simple path, that is no node may be visited more than once on a path.

[9] Note that even though p_1 is not a hypernormal path, there is another hypernormal path connecting the same two nodes

[10] v is a dominance child of u in a UG U, if (u, v) is a constraint edge in U.

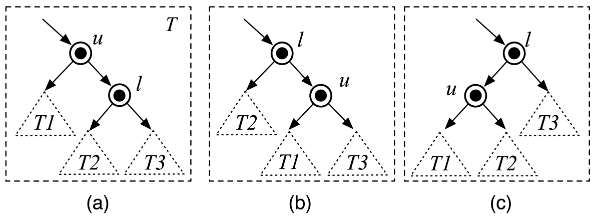

Figure 8: Proof of Proposition 5

is not a free fragment because it violates condition (P3b); and (iii) proves that U is not a super net because F violates condition (D8c).

Proposition 5. *If U is a satisfiable super net, the necessary freeness conditions in Proposition 3 are also sufficient.*

Proof sketch. Let F rooted at l be a fragment satisfying the three conditions in Proposition 3. Among all the solutions of U, we pick a solution T in which the depth d of l is minimal. Using proof by contradiction, we show that $d = 0$, which proves l is the root of T. If $d > 0$, there is some node u that outscopes l (Figure 8(a)). Lemma 2 and 4 guarantee that at least one of the trees in Figures 8(b,c) is a solution of U. So U has a solution in which, the depth of l is smaller than d. Contradiction!

3 SAT and ENUM algorithms

Following Lemma 4 and Proposition 5, Table 1 gives the algorithms for the satisfiability (**SAT**), and the enumeration (**ENUM**) of super nets.

Theorem 6. *ENUM and SAT are correct.*

Proof sketch. Using Lemma 4 and induction on the depth of the recursion, it is easy to see that if ENUM or SAT returns a tree T, T is a solution of U. This proves that ENUM and SAT are sound. An inductive proof is used to prove the completeness as well. Consider a solution T of depth n of U (Figure 5). It can be shown that T_1 and T_2 must be the solutions to U_1 and U_2. Therefore based on the induction assumption they are generated by $Solve(G)$, hence T is also generated by $Solve(G)$.

Let $U = (L \uplus H, S \uplus C)$. The running time of the algorithms depends on the depth of the recursion which is equal to the number of fragments/labels, $|L|$. At each depth it takes $O(|U|)$ to find the set of free fragments (Bodirsky et al., 2004) and also to compute $U - F$ for some free fragment F. ($|U| =_{def} |V| + |E|$, where $|V| =_{def} |L| + |H|$, and $|E| =_{def} |S| + |C|$). Therefore SAT (and each branch of ENUM) run(s) in $O(|L|.|U|)$ step. Therefore the worst-case time complexity of SAT and each branch of ENUM is quadratic in the size of U.

```
Solve(U)
1. If $U$ contains a single (label) node, return $U$.
2. Pick a free fragment $F$ with m holes
                        rooted at l, otherwise fail.
   // For SAT: pick arbitrarily.
   // For ENUM: pick non-deterministically.
3. Let $U_1, U_2, \cdots, U_m$ be WCCs of $U - F$.
4. Let $T_i = Solve(U_i)$ for $i = 1 \cdots m$.
5. Let $h_i$ be the hole of $F$ connected to
                  $U_i$ in $U - l$, for $i = 1 \cdots m$.
  (If for some $k$, $U_k$ is not connected to any hole
   of $F$ in $U - l$, let $h_k$ be the open hole of $F$.)
6. Build $T$ by plugging the root of $T_i$ into $h_i$,
                                  for $i = 1 \cdots m$.
7. Return $T$.
```

Table 1: ENUM and SAT algorithms

4 Super net versus weak net

Although net is a subset of weak net, to better understand the three frameworks, we first define net.

4.1 Net

Net was first defined by Koller et al. (2003), in order to find a subset of Hole Semantics that can be solved in polynomial-time. Nets do not contain any label-to-label constraints. In fact, out of the three possible structures that super net allows for a fragment F (Definition 9), net only allows for the first one, that is open-root.

Definition 10 (Net). (Thater, 2007)
Let U be a UG with no label-to-label constraints. U is called a ***net*** *iff for every fragment F:*

D10a. *F has no open hole.*

D10b. *If l_1, l_2 are two dominance children of a hole h of F, then l_1 and l_2 are hypernormally connected in $U - h$.*

The root of F is open, therefore (D8a) subsumes (D10a). Condition (D10b) is exactly the same as (D8b). Therefore, super net is a superset of net. Strictness of the superset relationship is trivial.

4.2 Weak net

Weak net was first introduced by Niehren and Thater (2003), in order to find a tractable subset of MRS. In order to model MRS, weak net allows for label-to-label constraints, but to stay a tractable framework it forces the following restrictions.

Definition 11 (Weak net). (Thater, 2007)
A UG U is a weak net iff for every fragment F:

D11a. *F has exactly one open node.*

D11b. *If l_1, l_2 are two dominance children of a node u of F, then l_1 and l_2 are hypernormally connected in $U-u$.*

Weak nets suffer from two limitations with respect to super nets.

First, out of the three possible types of fragment allowed by super net (Definition 9), weak net only allows for the first two; open-root and open-hole. In practice this becomes an issue only if new constraints are to be added to a UG after syntax/semantic interface. Since weak net requires one node of every fragment to be open, a constraint cannot be added if it violates this condition.[11]

Second, open-hole fragments in weak nets are more restricted than open-hole fragments in super nets. This is the Achilles' heel of weak nets (D11b). To see why, consider the UG in Figure 3 for the sentence *Every politician, whom I know a child of, runs* which we presented in Section 1. If F is the fragment for the quantifier *Every* and l is the root of F, the two dominance children of l are not (hypernormally) connected in $U-l$. Therefore, U is not a weak net. All the non-net examples we have found so far behave similarly. That is, there is a quantifier with more than one outgoing dominance edge. Once you remove the quantifier node, the dominance children are no longer weakly (and hence hypernormally) connected, violating condition (D11b). In super net, however, we define case 2 of condition (D8c) such that it does not force dominance children of l to be (hypernormally) connected, allowing for non-net structures such as the one in Figure 3.[12]

Proposition 7. *Weak net is a strict subset of super net.*

Proof. Consider an arbitrary weak net U, and let F be an arbitrary fragment of U rooted at l.

(i). F has exactly one open node, so it satisfies condition (D8a).

(ii). For every two holes of F, condition (D11b) guarantees that condition (D8b) holds.

(iii).
- Case 1) F has no open hole:
  Based on condition (D11a) the root of F is open, hence it has no dominance children. (D8c) trivially holds in this case.
- Case 2) F has an open hole:
  Based on condition (D11b) every two dominance children of l are hypernormally connected, so (D8c) holds in this case too.

Therefore, every fragment F satisfies all the conditions in Definition 8, hence U is a super net. This and the fact that Figure 3 is a super net but not a weak net complete the proof. □

4.3 Underspecification graph vs. weakly normal dominance graph

Dominance graphs and their ancestor, dominance constraints, are designed for solving constrained tree structures in general. Therefore, some of the terminology of dominance graph may seem counterintuitive when dealing with scope underspecification. For example the notion of **solution** in that formalism is broader than what is known as solution in scope underspecification formalisms. As defined there (but translated into our terminology), a solution may contain unplugged holes, or holes plugged with more than one label. This broad notion of solution is computationally less expensive such that an algorithm very similar to the one in Table 1 can be used to solve every weakly normal dominance graph (Bodirsky et al., 2004). Solution, as defined in this paper (Definition 4), corresponds to the notion of **simple leaf-labeled solved forms** (a.k.a. **configuration**) in dominance graphs. Although solutions of a weakly normal dominance graph can be found in polynomial time, finding configurations is NP-complete. Solvability of underspecification graphs is equivalent to configurability of weakly normal dominance graphs, and hence NP-complete.

[11] As discussed in Section 5, by defining the notion of *downward connectedness*, Koller and Thater (2007) address this issue of weak nets, at the expense of cubic time complexity.

[12] For simplicity, throughout this paper we have used the term non-net to refer to non-(weak net) UGs.

5 Related work

We already compared our model with nets and weak nets. Koller and Thater (2007) present another extension of weak nets, downward connected nets. They show that if a *dominance graph* has a subgraph which is a weak net, it can be solved in polynomial time. This addresses the first limitation of weak nets, discussed in Section 4.2, but it does not solve the second one, because the graph in Figure 3 neither is a weak net, nor has a weak-net subgraph.

Downward connected dominance graph, in its general form, goes beyond weakly normal dominance graph (and hence UG), incorporating label-to-hole constraints. It remains for future work to investigate whether allowing for label-to-hole constraints adds any value to the framework within the context of scope underspecified semantics, or whether it is possible to model the same effect using hole-to-label and label-to-label constraints. In any case, the same extension can be applied to super nets as well, defining downward connected super nets, a strict super set of downward connected nets, solvable using similar algorithms with the same time/space complexity.

Another tractable framework presented in the past is our own framework, Canonical Form Underspecified Representation (**CF-UR**) (Manshadi et al., 2009), motivated by Minimal Recursion Semantics. CF-UR is defined to characterize the set of all MRS structures generated by the MRS semantic composition process (Manshadi et al., 2008). CF-UR in its general form is not tractable. Therefore, we define a notion of coherence called **heart-connectedness** and show that all heart-connected CF-UR structures can be solved efficiently. We also show that heart-connected CF-UR covers the family of non-net structures, so CF-UR is in fact the first framework to address the non-net structures. In spite of that, CF-UR is quite restricted and does not allow for adding new constraints after semantic composition.

In recent work, Koller et al. (2008) suggest using Regular Tree Grammars for scope underspecification, a probabilistic version of which could be used to find the best reading. The framework goes beyond the formalisms discussed in this paper and is expressively complete in Ebert (2005)'s sense of completeness, i.e. it is able to describe any subset of the readings of a UR. However, this power comes at the cost of exponential complexity. In practice, RTG is built on top of weak nets, benefiting from the compactness of this framework to remain tractable. Being a super set of weak net, super net provides a more powerful core for RTG.

Koller and Thater (2010) address the problem of finding the weakest readings of a UR, which are those entailed by some reading(s), but not entailing any other reading of the UR. By only considering the weakest readings, the space of solutions will be dramatically reduced. Note that entailment using the weakest readings is sound but not complete.

6 Summary and Future work

Weakly normal dominance graph brings many current constraint-based formalisms under a uniform framework, but its configurability is intractable in its general form. In this paper, we present a tractable subset of this framework. We prove that this subset, called super net, is a strict superset of weak net, a previously known tractable subset of the framework, and that it covers a family of coherent natural language sentences whose underspecified representation are known not to belong to weak nets.

As mentioned in Section 5, another extension of weak nets, downward connected nets, has been proposed by Koller and Thater (2007), which addresses some of the limitations of weak nets, yet is unable to solve the known family of non-net structures. A thorough comparison between super nets and downward connected nets remains for future work.

Another interesting property of super nets to be explored is how they compare to heart-connected graphs. Heart-connectedness has been introduced as a mathematical criterion for verifying the coherence of an underspecified representation within the framework of underspecification graph (Manshadi et al., 2009). Our early investigation shows that super nets may contain all heart-connected UGs. If this conjecture is true, super net would be broad enough to cover every coherent natural language sentence (under this notion of coherence). We leave a detailed investigation of this conjecture for the future.

Acknowledgments

This work was support in part by NSF grant 1012205, and ONR grant N000141110417.

References

Hiyan Alshawi and Richard Crouch. 1992. Monotonic semantic interpretation. In *Proceedings of ACL '92*, pages 32–39.

Ernst Althaus, Denys Duchier, Alexander Koller, Kurt Mehlhorn, Joachim Niehren, and Sven Thiel. 2003. An efficient graph algorithm for dominance constraints. *J. Algorithms*, 48(1):194–219, August.

Manuel Bodirsky, Denys Duchier, Joachim Niehren, and Sebastian Miele. 2004. An efficient algorithm for weakly normal dominance constraints. In *In ACM-SIAM Symposium on Discrete Algorithms*. The ACM Press.

J. Bos. 1996. Predicate logic unplugged. In *In Proceedings of the 10th Amsterdam Colloquium*, pages 133–143.

J. Bos. 2002. *Underspecification and Resolution in Discourse Semantics*. Saarbrücken dissertations in computational linguistics and language technology. DFKI.

Ann Copestake, Alex Lascarides, and Dan Flickinger. 2001. An algebra for semantic construction in constraint-based grammars. In *Proceedings of ACL '01*, pages 140–147.

Christian Ebert. 2005. Formal investigations of underspecified representations. Technical report, King's College, London, UK.

M. Egg, A. Koller, and J. Niehren. 2001. The constraint language for lambda structures. *J. of Logic, Lang. and Inf.*, 10(4):457–485, September.

Ruth Fuchss, Alexander Koller, Joachim Niehren, and Stefan Thater. 2004. Minimal recursion semantics as dominance constraints: Translation, evaluation, and analysis. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 247–254, Barcelona, Spain, July.

Jerry R. Hobbs and Stuart M. Shieber. 1987. An algorithm for generating quantifier scopings. *Comput. Linguist.*, 13(1-2):47–63, January.

Alexander Koller and Stefan Thater. 2007. Solving unrestricted dominance graphs. In *Proceedings of the 12th Conference on Formal Grammar*, Dublin.

Alexander Koller and Stefan Thater. 2010. Computing weakest readings. In *Proceedings of the 48th ACL*, Uppsala.

Alexander Koller, Joachim Niehren, and Stefan Thater. 2003. Bridging the gap between underspecification formalisms: Hole semantics as dominance constraints. In *Proceedings of the 11th EACL*, Budapest.

Alexander Koller, Michaela Regneri, and Stefan Thater. 2008. Regular tree grammars as a formalism for scope underspecification. In *Proceedings of ACL-08: HLT*, Columbus, Ohio.

Mehdi H. Manshadi, James F. Allen, and Mary Swift. 2008. Toward a universal underspecified semantic representation. In *Proceedings of the 13th Conference on Formal Grammar*, Hamburg, Germany, August.

Mehdi H. Manshadi, James F. Allen, and Mary Swift. 2009. An efficient enumeration algorithm for canonical form underspecified semantic representations. In *Proceedings of the 14th Conference on Formal Grammar*, Bordeaux, France, July.

Joachim Niehren and Stefan Thater. 2003. Bridging the gap between underspecification formalisms: minimal recursion semantics as dominance constraints. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 367–374, Stroudsburg, PA, USA. Association for Computational Linguistics.

S. Oepen, K. Toutanova, S. Shieber, C. Manning, D. Flickinger, and T. Brants. 2002. The lingo redwoods treebank motivation and preliminary applications. In *Proceedings of COLING '02*, pages 1–5.

S. Thater. 2007. *Minimal Recursion Semantics as Dominance Constraints: Graph-theoretic Foundation and Application to Grammar Engineering*. Saarbrücken dissertations in computational linguistics and language technology. Universität des Saarlandes.

W A Woods. 1986. Semantics and quantification in natural language question answering. In Barbara J. Grosz, Karen Sparck-Jones, and Bonnie Lynn Webber, editors, *Readings in natural language processing*, pages 205–248. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.