Minimal Recursion Semantics as Dominance Constraints: Translation, Evaluation, and Analysis

Ruth Fuchss, Alexander Koller, Joachim Niehren, Stefan Thater

▶ **To cite this version:**

Ruth Fuchss, Alexander Koller, Joachim Niehren, Stefan Thater. Minimal Recursion Semantics as Dominance Constraints: Translation, Evaluation, and Analysis. 42th Meeting of the Association for Computational Linguistics, 2004, Barcelona, Spain. pp.247-254. inria-00536532

HAL Id: inria-00536532
https://hal.inria.fr/inria-00536532

Submitted on 16 Nov 2010

Minimal Recursion Semantics as Dominance Constraints: Translation, Evaluation, and Analysis

Ruth Fuchss,[1] **Alexander Koller,**[1] **Joachim Niehren,**[2] **and Stefan Thater**[1]

[1] Dept. of Computational Linguistics, Saarland University, Saarbrücken, Germany [*]

[2] INRIA Futurs, Lille, France

{fuchss,koller,stth}@coli.uni-sb.de

Abstract

We show that a practical translation of MRS descriptions into normal dominance constraints is feasible. We start from a recent theoretical translation and verify its assumptions on the outputs of the English Resource Grammar (ERG) on the Redwoods corpus. The main assumption of the translation—that all relevant underspecified descriptions are *nets*—is validated for a large majority of cases; all non-nets computed by the ERG seem to be systematically incomplete.

1 Introduction

Underspecification is the standard approach to dealing with scope ambiguity (Alshawi and Crouch, 1992; Pinkal, 1996). The readings of underspecified expressions are represented by compact and concise descriptions, instead of being enumerated explicitly. Underspecified descriptions are easier to derive in syntax-semantics interfaces (Egg et al., 2001; Copestake et al., 2001), useful in applications such as machine translation (Copestake et al., 1995), and can be resolved by need.

Two important underspecification formalisms in the recent literature are Minimal Recursion Semantics (MRS) (Copestake et al., 2004) and dominance constraints (Egg et al., 2001). MRS is the underspecification language which is used in large-scale HPSG grammars, such as the English Resource Grammar (ERG) (Copestake and Flickinger, 2000). The main advantage of dominance constraints is that they can be solved very efficiently (Althaus et al., 2003; Bodirsky et al., 2004).

Niehren and Thater (2003) defined, in a theoretical paper, a translation from MRS into normal dominance constraints. This translation clarified the precise relationship between these two related formalisms, and made the powerful meta-theory of dominance constraints accessible to MRS. Their goal was to also make the large grammars for MRS

and the efficient constraint solvers for dominance constraints available to the other formalism.

However, Niehren and Thater made three technical assumptions:

1. that EP-conjunction can be resolved in a preprocessing step;

2. that the qeq relation in MRS is simply dominance;

3. and (most importantly) that all linguistically correct and relevant MRS expressions belong to a certain class of constraints called *nets*.

This means that it is not obvious whether their result can be immediately applied to the output of practical grammars like the ERG.

In this paper, we evaluate the truth of these assumptions on the MRS expressions which the ERG computes for the sentences in the Redwoods Treebank (Oepen et al., 2002). The main result of our evaluation is that 83% of the Redwoods sentences are indeed nets, and 17% aren't. A closer analysis of the non-nets reveals that they seem to be systematically incomplete, *i. e.* they predict more readings than the sentence actually has. This supports the claim that all linguistically correct MRS expressions are indeed nets. We also verify the other two assumptions, one empirically and one by proof.

Our results are practically relevant because dominance constraint solvers are much faster and have more predictable runtimes when solving nets than the LKB solver for MRS (Copestake, 2002), as we also show here. In addition, nets might be useful as a debugging tool to identify potentially problematic semantic outputs when designing a grammar.

Plan of the Paper. We first recall the definitions of MRS (§2) and dominance constraints (§3). We present the translation from MRS-nets to dominance constraints (§4) and prove that it can be extended to MRS-nets with EP-conjunction (§5). Finally we evaluate the net hypothesis and the qeq assumption on the Redwoods corpus, and compare runtimes (§6).

---

2 Minimal Recursion Semantics

This section presents a definition of Minimal Recursion Semantics (MRS) (Copestake et al., 2004) including EP-conjunctions with a merging semantics. Full MRS with qeq-semantics, top handles, and event variables will be discussed in the last paragraph.

MRS Syntax. MRS constraints are conjunctive formulas over the following vocabulary:

1. An infinite set of *variables* ranged over by h. Variables are also called *handles*.

2. An infinite set of *constants* x, y, z denoting indivual variables of the object language.

3. A set of *function symbols* ranged over by P, and a set of *quantifier symbols* ranged over by Q. Pairs Q_x are further function symbols.

4. The binary predicate symbol '$=_q$'.

MRS constraints have three kinds of literals, two kinds of *elementary predications* (EPs) in the first two lines and *handle constraints* in the third line:

1. $h : P(x_1, \ldots, x_n, h_1, \ldots, h_m)$, where $n, m \geq 0$

2. $h : Q_x(h_1, h_2)$

3. $h_1 =_q h_2$

In EPs, *label positions* are on the left of ':' and *argument positions* on the right. Let M be a set of literals. The *label* set $\mathsf{lab}(M)$ contains all handles of M that occur in label but not in argument position, and the *argument handle* set $\mathsf{arg}(M)$ contains all handles of M that occur in argument but not in label position.

Definition 1 (MRS constraints). An *MRS constraint* (*MRS* for short) is a finite set M of MRS-literals such that:

M1 every handle occurs at most once in argument position in M,

M2 handle constraints $h =_q h'$ always relate argument handles h to labels h', and

M3 for every constant (individual variable) x in argument position in M there is a unique literal of the form $h : Q_x(h_1, h_2)$ in M.

We say that an MRS M is *compact* if every handle h in M is either a label or an argument handle. Compactness simplifies the following proofs, but it is no serious restriction in practice.

We usually represent MRSs as directed graphs: the nodes of the graph are the handles of the MRS, EPs are represented as solid lines, and handle constraints are represented as dotted lines. For instance, the following MRS is represented by the graph on the left of Fig. 1.

$\{h_5 : \mathrm{some}_y(h_6, h_8), h_7 : \mathrm{book}(y), h_1 : \mathrm{every}_x(h_2, h_4),$
$\quad h_3 : \mathrm{student}(x), h_9 : \mathrm{read}(x, y), h_2 =_q h_3, h_6 =_q h_7\}$
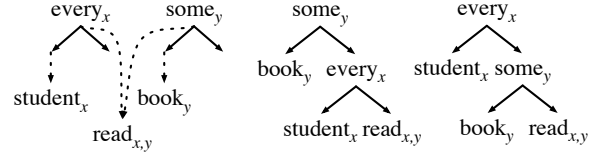

Figure 1: An MRS and its two configurations.

Note that the relation between bound variables and their binders is made explicit by *binding edges* drawn as dotted lines (cf. C2 below); transitively redundand binding edges (*e. g.*, from some_y to book_y) however are omitted.

MRS Semantics. Readings of underspecified representations correspond to *configurations* of MRS constraints. Intuitively, a configuration is an MRS where all handle constraints have been resolved by plugging the "tree fragments" into each other.

Let M be an MRS and h, h' be handles in M. We say that h *immediately outscopes* h' in M if there is an EP in M with label h and argument handle h', and we say that h *outscopes* h' in M if the pair (h, h') belongs to the reflexive transitive closure of the immediate outscope relation of M.

Definition 2 (MRS configurations). An MRS M is a *configuration* if it satisfies conditions C1 and C2:

C1 The graph of M is a tree of solid edges: (i) all handles are labels *i. e.*, $\mathsf{arg}(M) = \emptyset$ and M contains no handle constraints, (ii) handles don't properly outscope themselves, and (iii) all handles are pairwise connected by EPs in M.

C2 If $h : Q_x(h_1, h_2)$ and $h' : P(\ldots, x, \ldots)$ belong to M, then h outscopes h' in M *i. e.*, binding edges in the graph of M are transitively redundant.

We say that a configuration M is *configuration of an MRS* M' if there exists a partial substitution $\sigma : \mathsf{lab}(M') \rightsquigarrow \mathsf{arg}(M')$ that states how to identify labels with argument handles of M' so that:

C3 $M = \{\sigma(E) \mid E \text{ is an EP in } M'\}$, and

C4 for all $h =_q h'$ in M', h outscopes $\sigma(h')$ in M.

The value $\sigma(E)$ is obtained by substituting all labels in $\mathsf{dom}(\sigma)$ in E while leaving all other handels unchanged.

The MRS on the left of Fig. 1, for instance, has two configurations given to the right.

EP-conjunctions. Definitions 1 and 2 generalize the idealized definition of MRS of Niehren and Thater (2003) by EP-conjunctions with a merging semantics. An MRS M contains an *EP-conjunction* if it contains different EPs with the same label h. The intuition is that EP-conjunctions are interpreted by object language conjunctions.
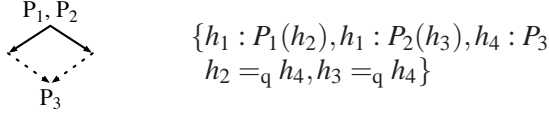
Figure 2: An unsolvable MRS with EP-conjunction

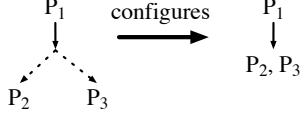$$\{h_1 : P_1(h_2), h_1 : P_2(h_3), h_4 : P_3 \\ h_2 =_q h_4, h_3 =_q h_4\}$$

Figure 3: A solvable MRS without merging-free configuration

Fig. 2 shows an MRS with an EP-conjunction and its graph. The function symbols of both EPs are conjoined and their arguments are merged into a set. The MRS does not have configurations since the argument handles of the merged EPs cannot jointly outscope the node P_4.

We call a configuration *merging* if it contains EP-conjunctions, and *merging-free* otherwise. Merging configurations are needed to solve EP-conjuctions such as $\{h : P_1, h : P_2\}$. Unfortunately, they can also solve MRSs without EP-conjunctions, such as the MRS in Fig. 3. The unique configuration of this MRS is a merging configuration: the labels of P_1 and P_2 must be identified with the only available argument handle. The admission of merging configurations may thus have important consequences for the solution space of arbitrary MRSs.

Standard MRS. Standard MRS requires three further extensions: (i) qeq-semantics, (ii) top-handles, and (iii) event variables. These extensions are less relevant for our comparision.

The qeq-semantics restricts the interpretation of handle constraints beyond dominance. Let M be an MRS with handles h, h'. We say that h *is qeq* h' *in* M if either $h = h'$, or there is an EP $h : Q_x(h_0, h_1)$ in M and h_1 is qeq h' in M. Every qeq-configuration is a configuration as defined above, but not necessarily *vice versa.* The qeq-restriction is relevant in theory but will turn out unproblematic in practice (see §6).

Standard MRS requires the existence of top handles in all MRS constraints. This condition doesn't matter for MRSs with connected graphs (see (Bodirsky et al., 2004) for the proof idea). MRSs with unconnected graphs clearly do not play any role in practical underspecified semantics.

Finally, MRSs permit events variables e, e' as a second form of constants. They are treated equally to individual variables except that they cannot be bound by quantifiers.

3 Dominance Constraints

Dominance constraints are a general framework for describing trees. For scope underspecification, they are used to describe the syntax trees of object language formulas. Dominance constraints are the core language underlying CLLS (Egg et al., 2001) which adds parallelism and binding constraints.

Syntax and semantics. We assume a possibly infinite signature $\Sigma = \{f, g, \ldots\}$ of function symbols with fixed arities (written $\mathrm{ar}(f)$) and an infinite set of variables ranged over by X, Y, Z.

A dominance constraint φ is a conjunction of dominance, inequality, and labeling literals of the following form, where $\mathrm{ar}(f) = n$:

$$\varphi ::= X \lhd^* Y \mid X \neq Y \mid X : f(X_1, \ldots, X_n) \mid \varphi \wedge \varphi'$$

Dominance constraints are interpreted over finite constructor trees *i. e.*, ground terms constructed from the function symbols in Σ. We identify ground terms with trees that are rooted, ranked, edge-ordered and labeled. A solution for a dominance constraint φ consists of a tree τ and an assignment α that maps the variables in φ to nodes of τ such that all constraints are satisfied: labeling literals $X : f(X_1, \ldots, X_n)$ are satisfied iff $\alpha(X)$ is labeled with f and its daughters are $\alpha(X_1), \ldots, \alpha(X_n)$ in this order; dominance literals $X \lhd^* Y$ are satisfied iff $\alpha(X)$ dominates $\alpha(Y)$ in τ; and inequality literals $X \neq Y$ are satisfied iff $\alpha(X)$ and $\alpha(Y)$ are distinct nodes.

Solved forms. Satisfiable dominance constraints have infinitely many solutions. Constraint solvers for dominance constraints therefore do not enumerate solutions but *solved forms i. e.*, "tree shaped" constraints. To this end, we consider *(weakly) normal* dominance constraints (Bodirsky et al., 2004).

We call a variable a *hole* of φ if it occurs in argument position in φ and a *root* of φ otherwise.

Definition 3. A dominance constraint φ is *normal* if it satisfies the following conditions.

N1 (a) each variable of φ occurs at most once in the labeling literals of φ.

   (b) each variable of φ occurs at least once in the labeling literals of φ.

N2 for distinct roots X and Y of φ, $X \neq Y$ is in φ.

N3 (a) if $X \lhd^* Y$ occurs in φ, Y is a root in φ.

   (b) if $X \lhd^* Y$ occurs in φ, X is a hole in φ.

We call φ *weakly normal* if it satisfies the above properties except for N1 (b) and N3 (b).

Note that Definition 3 imposes *compactness*: the height of tree fragments is always one. This is not
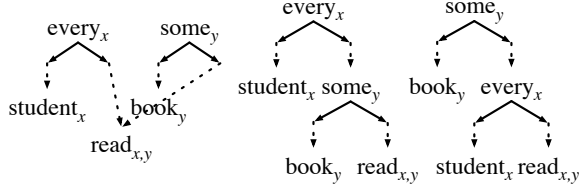
Figure 4: A normal dominance constraint (left) and its two solved forms (right).

(a) strong   (b) weak   (c) island

Figure 5: Fragment Schemata of Nets

a serious restriction, as weakly normal dominance constraints can be compactified, provided that dominance links relate either roots or holes with roots.

Weakly normal dominance constraints φ can be represented by *dominance graphs*. The dominance graph of φ is a directed graph $G = (V, E_T \uplus E_D)$ defined as follows. The nodes of G are the variables of φ. Labeling literals $X : f(X_1, \ldots, X_k)$ are represented by *tree edges* $(X, X_i) \in E_T$, for $1 \le i \le k$, and dominance literals $X \lhd^* X'$ are represented by *dominance edges* $(X, X') \in E_D$. Inequality literals are not represented in the graph. In pictures, labeling literals are drawn with solid lines and dominance edges with dotted lines.

We say that a constraint φ is in *solved form* if its graph is in solved form. A graph G is in solved form iff it is a forest. The *solved forms of G* are solved forms G' which are more specific than G *i.e.*, they differ only in their dominance edges and the reachability relation of G extends the reachability of G'. A *minimal solved form* is a solved form which is minimal with respect to specificity. *Simple solved forms* are solved forms where every hole has exactly one outgoing dominance edge. Fig. 4 shows as a concrete example the translation of the MRS description in Fig. 1 together with its two minimal solved forms. Both solved forms are simple.

4 Translating Merging-Free MRS-Nets

This section defines MRS-nets without EP-conjunctions, and sketches their translation to normal dominance constraints. We define nets equally for MRSs and dominance constraints. The key semantic property of nets is that different notions of solutions coincide. In this section, we show that merging-free configurations coincides to minimal solved forms. §5 generalizes the translation by adding EP-conjunctions and permitting merging semantics.

Pre-translation. An MRS constraint M can be represented as a corresponding dominance constraint φ_M as follows: The variables of φ_M are the handles of M, and the literals of φ_M correspond
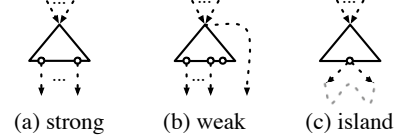
those of M in the following sence:

$$h : P(x_1, \ldots, x_n, h_1, \ldots, h_k) \mapsto h : P_{x_1, \ldots, x_n}(h_1, \ldots, h_k)$$
$$h : Q_x(h_1, h_2) \mapsto h : Q_x(h_1, h_2)$$
$$h =_q h' \mapsto h \lhd^* h'$$

Additionally, dominance literals $h \lhd^* h'$ are added to φ_M for all h, h' s. t. $h : Q_x(h_1, h_2)$ and $h' : P(\ldots, x, \ldots)$ belong to M (cf. C2), and literals $h \ne h'$ are added to φ_M for all h, h' in distinct label position in M.

Lemma 1. If a compact MRS M does not contain EP-conjunctions then φ_M is weakly normal, and the graph of M is the transitive reduction of the graph of φ_M.

Nets. A *hypernormal path* (Althaus et al., 2003) in a constraint graph is a path in the undirected graph that contains for every leaf X at most one incident dominance edge.

Let φ be a weakly normal dominance constraint and let G be the constraint graph of φ. We say that φ is a *dominance net* if the transitive reduction G of G is a *net*. G' is a net if every tree fragment F of G' satisfies one of the following three conditions, illustrated in Fig. 5:

Strong. Every hole of F has exactly one outgoing dominance edge, and there is no weak root-to-root dominance edge.

Weak. Every hole except for the last one has exactly one outgoing dominance edge; the last hole has no outgoing dominance edge, and there is exactly one weak root-to-root dominance edge.

Island. The fragment has one hole X, and all variables which are connected to X by dominance edges are connected by a hypernormal path in the graph where F has been removed.

We say that an MRS M is an *MRS-net* if the pre-translation of its literals results in a dominance net φ_M. We say that an MRS-net M is connected if φ_M is connected; φ_M is connected if the graph of φ_M is connected.

Note that this notion of MRS-nets implies that MRS-nets cannot contain EP-conjunctions as otherwise the resulting dominance constraint would not be weakly normal. §5 shows that EP-conjunctions

can be resolved *i. e.*, MRSs with EP-conjunctions can be mapped to corresponding MRSs without EP-conjunctions.

If M is an MRS-net (without EP-conjunctions), then M can be translated into a corresponding dominance constraint φ by first pre-translating M into a φ_M and then *normalizing* φ_M by replacing weak root-to-root dominance edges in weak fragments by dominance edges which start from the open last hole.

Theorem 1 (Niehren and Thater, 2003). Let M be an MRS and φ_M be the translation of M. If M is a connected MRS-net, then the merging-free configurations of M bijectively correspond to the minimal solved forms of the φ_M.

The following section generalizes this result to MRS-nets with a merging semantics.

5 Merging and EP-Conjunctions

We now show that if an MRS is a net, then all its configurations are merging-free, which in particular means that the translation can be applied to the more general version of MRS with a merging semantics.

Lemma 2 (Niehren and Thater, 2003). All minimal solved forms of a connected dominance net are simple.

Lemma 3. If all solved forms of a normal dominance constraint are simple, then all of its solved forms are minimal.

Theorem 2. The configurations of an MRS-net M are merging-free.

Proof. Let M' be a configuration of M and let σ be the underlying substitution. We construct a solved form $\varphi_{M'}$ as follows: the labeling literals of $\varphi_{M'}$ are the pre-translations of the EPs in M, and $\varphi_{M'}$ has a dominance literal $h' \vartriangleleft^* h$ iff $(h, h') \in \sigma$, and inequality literals $X \neq Y$ for all distinct roots in $\varphi_{M'}$.

By condition C1 in Def. 2, the graph of M' is a tree, hence the graph of $\varphi_{M'}$ must also be a tree *i. e.*, $\varphi_{M'}$ is a solved form. $\varphi_{M'}$ must also be more specific than the graph of φ_M because the graph of M' satisfies all dominance requirements of the handle constraints in M, hence $\varphi_{M'}$ is a solved form of φ_M.

M clearly solved $\varphi_{M'}$. By Lemmata 2 and 3, $\varphi_{M'}$ must be simple and minimal because φ_M is a net. But then M' cannot contain EP-conjunctions *i. e.*, M' is merging-free. $\square$

The merging semantics of MRS is needed to solve EP-conjunctions. As we have seen, the merging semantics is not relevant for MRS constraints which are nets. This also verifies Niehren and Thater's (2003) assumption that EP-conjunctions are "syntactic sugar" which can be resolved in a pre-processing step: EP-conjunctions can be resolved by exhaustively applying the following rule which adds new literals to make the implicit conjunction explicit:

$$h : E_1(h_1, \ldots, h_n), h : E_2(h'_1, \ldots, h'_m) \Rightarrow$$
$$h : \text{'}E_1 \& E_2\text{'}(h_1, \ldots, h_n, h'_1, \ldots, h'_m),$$

where $E(h_1, \ldots, h_n)$ stands for an EP with argument handles $h_1, \ldots, h_n$, and where '$E_1 \& E_2$' is a complex function symbol. If this rule is applied exhaustively to an MRS M, we obtain an MRS M' without EP-conjunctions. It should be intuitively clear that the configurations of M and M' correspond; Therefore, the configurations of M also correspond to the minimal solved forms of the translation of M'.

6 Evaluation

The two remaining assumptions underlying the translation are the "net-hypothesis" that all linguistically relevant MRS expressions are nets, and the "qeq-hypothesis" that handle constraints can be given a dominance semantics practice. In this section, we empirically show that both assumptions are met in practice.

As an interesting side effect, we also compare the run-times of the constraint-solvers we used, and we find that the dominance constraint solver typically outperforms the MRS solver, often by significant margins.

Grammar and Resources. We use the English Resource Grammar (ERG), a large-scale HPSG grammar, in connection with the LKB system, a grammar development environment for typed feature grammars (Copestake and Flickinger, 2000). We use the system to parse sentences and output MRS constraints which we then translate into dominance constraints. As a test corpus, we use the Redwoods Treebank (Oepen et al., 2002) which contains 6612 sentences. We exclude the sentences that cannot be parsed due to memory capacities or words and grammatical structures that are not included in the ERG, or which produce ill-formed MRS expressions (typically violating M1) and thus base our evaluation on a corpus containing 6242 sentences. In case of syntactic ambiguity, we only use the first reading output by the LKB system.

To enumerate the solutions of MRS constraints and their translations, we use the MRS solver built into the LKB system and a solver for weakly normal dominance constraints (Bodirsky et al., 2004),
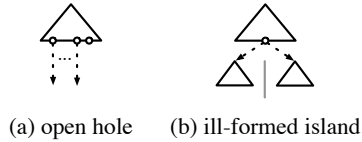
(a) open hole   (b) ill-formed island

Figure 6: Two classes of non-nets

which is implemented in C++ and uses LEDA, a class library for efficient data types and algorithms (Mehlhorn and Näher, 1999).

6.1 Relevant Constraints are Nets

We check for 6242 constraints whether they constitute nets. It turns out that 5200 (83.31%) constitute nets while 1042 (16.69%) violate one or more net-conditions.

Non-nets. The evaluation shows that the hypothesis that all relevant constraints are nets seems to be falsified: there are constraints that are not nets. However, a closer analysis suggests that these constraints are incomplete and predict more readings than the sentence actually has. This can also be illustrated with the average number of solutions: For the Redwoods corpus in combination with the ERG, nets have 1836 solutions on average, while non-nets have 14039 solutions, which is a factor of 7.7. The large number of solutions for non-nets is due to the "structural weakness" of non-nets; often, non-nets have only merging configurations.

Non-nets can be classified into two categories (see Fig. 6): The first class are violated "strong" fragments which have holes without outgoing dominance edge and without a corresponding root-to-root dominance edge. The second class are violated "island" fragments where several outgoing dominance edges from one hole lead to nodes which are not hypernormally connected. There are two more possibilities for violated "weak" fragments—having more than one weak dominance edge or having a weak dominance edge without empty hole—, but they occur infrequently (4.4%). If those weak fragments were normalized, they would constitute violated island fragments, so we count them as such.

124 (11.9%) of the non-nets contain empty holes, 762 (73.13%) contain violated island fragments, and 156 (14.97%) contain both. Those constraints that contain only empty holes and no violated island fragments cannot be configured, as in configurations, all holes must be filled.

Fragments with open holes occur frequently, but not in all contexts, for constraints representing for example time specifications (*e. g.*, *"from nine to twelve"* or *"a three o'clock flight"*) or intensional expressions (*e. g.*, *"Is it?"* or *"I suppose"*). Ill-
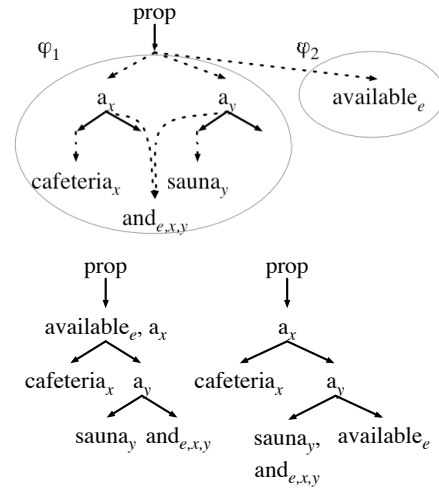

Figure 7: An MRS for "A sauna and a cafeteria are available" (top) and two of sixteen merging configurations (below).
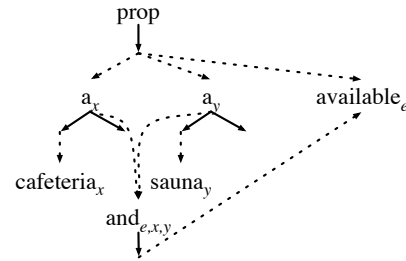

Figure 8: The "repaired" MRS from Fig. 7

formed island fragments are often triggered by some kind of coordination, like *"a restaurant and/or a sauna"* or *"a hundred and thirty Marks"*, also implicit ones like *"one hour thirty minutes"* or *"one thirty"*. Constraints with both kinds of violated fragments emerge when there is some input that yields an open hole and another part of the input yields a violated island fragment (for example in constructions like *"from nine to eleven thirty"* or *"the ten o'clock flight Friday or Thursday"*, but not necessarily as obviously as in those examples).

The constraint on the left in Fig. 7 gives a concrete example for violated island fragments. The topmost fragment has outgoing dominance edges to otherwise unconnected subconstraints φ_1 and φ_2. Under the merging-free semantics of the MRS dialect used in (Niehren and Thater, 2003) where every hole has to be filled exactly once, this constraint cannot be configured: there is no hole into which "available" could be plugged. However, standard MRS has merging configuration where holes can be filled more than once. For the constraint in Fig. 7 this means that "available" can be merged in almost

everywhere, only restricted by the "qeq-semantics" which forbids for instance "available" to be merged with "sauna." In fact, the MRS constraint solver derives sixteen configurations for the constraint, two of which are given in Fig. 7, although the sentence has only two scope readings.

We conjecture that non-nets are semantically "incomplete" in the sense that certain constraints are missing. For instance, an alternative analysis for the above constraint is given in Fig. 8. The constraint adds an additional argument handle to "and" and places a dominance edge from this handle to "available." In fact, the constraint is a net; it has exactly two readings.

6.2 Qeq is dominance

For all nets, the dominance constraint solver calculates the same number of solutions as the MRS solver does, with 3 exceptions that hint at problems in the syntax-semantics interface. As every configuration that satisfies proper qeq-constraints is also a configuration if handle constraints are interpreted under the weaker notion of dominance, the solutions computed by the dominance constraint solver and the MRS solver must be identical for every constraint. This means that the additional expressivity of proper qeq-constraints is not used in practice, which in turn means that in practice, the translation is sound and correct even for the standard MRS notion of solution, given the constraint is a net.

6.3 Comparison of Runtimes

The availability of a large body of underspecified descriptions both in MRS and in dominance constraint format makes it possible to compare the solvers for the two underspecification formalisms. We measured the runtimes on all nets using a Pentium III CPU at 1.3 GHz. The tests were run in a multi-user environment, but as the MRS and dominance measurements were conducted pairwise, conditions were equal for every MRS constraint and corresponding dominance constraint.

The measurements for all MRS-nets with less than thirty dominance edges are plotted in Fig. 9. Inputs are grouped according to the constraint size. The filled circles indicate average runtimes within each size group for enumerating all solutions using the dominance solver, and the empty circles indicate the same for the LKB solver. The brackets around each point indicate maximum and minimum runtimes in that group. Note that the vertical axis is logarithmic.

We excluded cases in which one or both of the solvers did not return any results: There were 173 sentences (3.33% of all nets) on which the LKB

solver ran out of memory, and 1 sentence (0.02%) that took the dominance solver more than two minutes to solve.

The graph shows that the dominance constraint solver is generally much faster than the LKB solver: The average runtime is less by a factor of 50 for constraints of size 10, and this grows to a factor of 500 for constraints of size 25. Our experiments show that the dominance solver outperforms the LKB solver on 98% the cases. In addition, its runtimes are much more predictable, as the brackets in the graph are also shorter by two or three orders of magnitude, and the standard deviation is much smaller (not shown).

7 Conclusion

We developed Niehren and Thater's (2003) theoretical translation into a practical system for translating MRS into dominance constraints, applied it systematically to MRSs produced by English Resource Grammar for the Redwoods treebank, and evaluated the results. We showed that:

1. most "real life" MRS expressions are MRS-nets, which means that the translation is correct in these cases;

2. for nets, merging is not necessary (or even possible);

3. the practical translation works perfectly for all MRS-nets from the corpus; in particular, the $=_q$ relation can be taken as synonymous with dominance in practice.

Because the translation works so well in practice, we were able to compare the runtimes of MRS and dominance constraint solvers on the same inputs. This evaluation shows that the dominance constraint solver outperforms the MRS solver and displays more predictable runtimes. A researcher working with MRS can now solve MRS nets using the efficient dominance constraint solvers.

A small but significant number of the MRS constraints derived by the ERG are not nets. We have argued that these constraints seem to be systematically incomplete, and their correct completions are indeed nets. A more detailed evaluation is an important task for future research, but if our "net hypothesis" is true, a system that tests whether all outputs of a grammar are nets (or a formal "safety criterion" that would prove this theoretically) could be a useful tool for developing and debugging grammars.

From a more abstract point of view, our evaluation contributes to the fundamental question of what expressive power an underspecification formalism needs. It turned out that the distinction between qeq
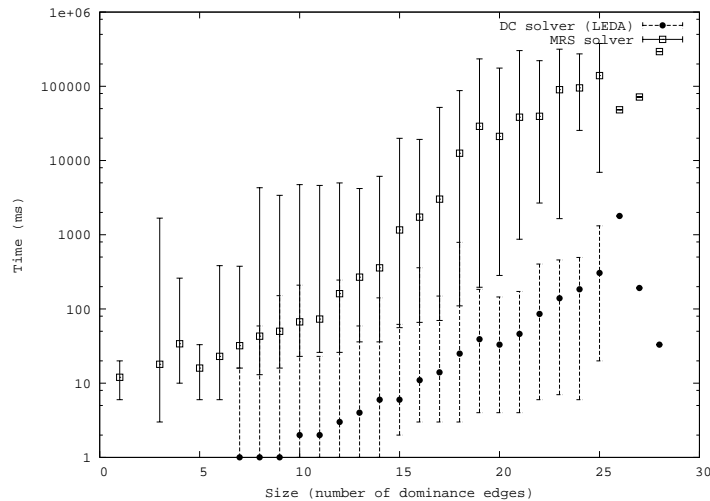
Figure 9: Comparison of runtimes for the MRS and dominance constraint solvers.

and dominance hardly plays a role in practice. If the net hypothesis is true, it also follows that merging is not necessary because EP-conjunctions can be converted into ordinary conjunctions. More research along these lines could help unify different underspecification formalisms and the resources that are available for them.

Acknowledgments We are grateful to Ann Copestake for many fruitful discussions, and to our reviewers for helpful comments.

References

H. Alshawi and R. Crouch. 1992. Monotonic semantic interpretation. In *Proc. 30th ACL*, pages 32–39.

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

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

Ann Copestake and Dan Flickinger. 2000. An open-source grammar development environment and broad-coverage english grammar using HPSG. In *Conference on Language Resources and Evaluation*.

Ann Copestake, Dan Flickinger, Rob Malouf, Susanne Riehemann, and Ivan Sag. 1995. Translation using Minimal Recursion Semantics. Leuven.

Ann Copestake, Alex Lascarides, and Dan Flickinger. 2001. An algebra for semantic construction in constraint-based grammars. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 132–139, Toulouse, France.

Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan Sag. 2004. Minimal recursion semantics: An introduction. *Journal of Language and Computation*. To appear.

Ann Copestake. 2002. *Implementing Typed Feature Structure Grammars*. CSLI Publications, Stanford, CA.

Markus Egg, Alexander Koller, and Joachim Niehren. 2001. The Constraint Language for Lambda Structures. *Logic, Language, and Information*, 10:457–485.

K. Mehlhorn and S. Näher. 1999. *The LEDA Platform of Combinatorial and Geometric Computing*. Cambridge University Press, Cambridge. See also http://www.mpi-sb.mpg.de/LEDA/.

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 of the Association for Computational Linguistics*.

Stephan Oepen, Kristina Toutanova, Stuart Shieber, Christopher Manning, Dan Flickinger, and Thorsten Brants. 2002. The LinGO Redwoods treebank: Motivation and preliminary applications. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING'02)*, pages 1253–1257.

Manfred Pinkal. 1996. Radical underspecification. In *10th Amsterdam Colloquium*, pages 587–606.