7

Toward a Universal Underspecified Semantic Representation

MEHDI HAFEZI MANSHADI, JAMES F. ALLEN, MARY SWIFT

Abstract

We define Canonical Form Minimal Recursion Semantics (CF-MRS) and prove that all the well-formed MRS structures generated by the MRS semantic composition algorithm are in this form. We prove that the qeq relationships are equivalent to outscoping relations when MRS structures are in this form. This result fills the gap between some underspecification formalisms and motivates defining a Canonical Form Underspecified Representation (CF-UR) which brings those underspecification formalisms together.

Keywords UNDERSPECIFICATION, SEMANTIC FORMALISMS, MINIMAL RECURSION SEMANTICS

7.1 Introduction

Several underspecification formalisms in semantic representation have been proposed during the last two decades, such as Quasi Logical Form (Alshawi and Crouch 1992), Hole Semantics (Bos 1996 and 2002), Minimal Recursion Semantics (Copestake et al. 2001), and Dominance Constraints (Egg et al. 2001). Recently there have been some efforts to bring these formalisms under a unified theory of underspecification.

Koller et al. (2003) define a back-and-forth translation between Hole Semantics and Dominance Constraints and show that under some specific restrictions (chain-connectedness and leaf-labeledness) the two formalisms generate the same number of solutions. Under these restrictions, however, the encoding is exact. By giving an example of a grammar, they also claim that all linguistically useful structures satisfy these restrictions.

Niehren and Thater (2003) give a translation from Minimal Recursion

FG-2008.
Philippe de Groote (Ed.).

Semantics (MRS) to Dominance Constraints. They define the concept of *nets* and show that when an MRS is in this form, there is a one-to-one correspondence between the (bounded) scope-resolved structures of an MRS and minimal solved forms of its corresponding dominance net. In their approach, however, they treat MRS's qeq relationships, which are a restricted version of outscoping relations (see section 7.2 for details), as simple dominance (i.e. outscoping) relations. Fuchss et al. (2004) claim that in MRS nets, the additional power of qeq relationships is not necessary and replacing them by simple outscoping relationships does not affect the number of scope-resolved structures. Although experimental data (the output of English Resource Grammar on Redwood corpus) supports this claim, there has been no theorem to show this equivalence rigorously. Furthermore, there are examples of coherent English sentences for which the MRS structure is not a net (Thater 2007). Therefore even if we accept that this equivalence holds for all MRS nets, the notion of net is not broad enough to cover all linguistically well-formed MRS structures.

In this paper we seek to prove the equivalence of qeq and outscoping relations for a class of MRS structures which we call Canonical Form MRS or CF-MRS. In a recent paper, Copestake et al. (2005) give an algorithm which applies to the syntactic tree of a sentence to build its MRS structure. Here, we define the notion of canonical form and show that every well-formed MRS structure which is generated by this algorithm is in this form. Then, we show a very useful property of canonical form: we prove that when an MRS structure is in this form, Fuchss et al. (2004)'s claim about the equivalence of qeq and outscoping relations holds. However, our approach has the following two advantages:

- The assumption, that all the well-formed MRS structures occurring in practice are in canonical form, is well-justified.
- We rigorously prove that the equivalence of qeq and outscoping relationships hold for every MRS structure which is in this form.

The notion of CF-MRS and the equivalence of qeq and dominance relations for this class of MRS structures motivate the definition of a universal underspecified semantic representation which we call Canonical Form Underspecified Representation or CF-UR. We define the notion of CF-UR and show the back and forth translation between CF-MRS and CF-UR. We leave the details of the translation between CF-UR and the other two formalisms (Hole Semantics and Dominance Constraints) for future work.

The rest of this paper is organized as follows. We review the definition of MRS (7.2). We define CF-MRS and prove that every MRS which is generated by the semantic composition algorithm is in canonical form (7.3). We give a formal definition of CF-MRS (7.4) and show that qeq and outscoping relationships are equivalent when MRS structures are in this form (7.5). The notion of CF-UR is defined in section (7.6).

7.2 Minimal Recursion Semantics

Elementary Predications (EP) are the basic blocks of MRS. An EP is a labeled relation of the form

l: P(x1, x2, ..., h1, h2, ...)

where l is the *label* of the EP, P is the *relation*, x1, x2 ... are variables of the object language called *non-scopal arguments* (also referred to as ordinary variables) and h1, h2... are variables over the set of labels, called *handle-taking arguments* or *holes* of the EP. We use the term *handle* to include both holes and labels.

MRS recognizes three different types of EP. *Non-scopal* EPs are EPs with no hole. They model first order predicates in the object language. *Floating-scopal* or quantifier EPs are in the form l:Q(x, hr, hb) where Q is the actual generalized quantifier; x is the variable quantified by Q; and hr and hb are holes for the restriction and the body of the quantifier and are referred to as *restriction hole* and *body hole* respectively. All other EPs are called *fixed-scopal* EPs. They model modal operators in the object language. The term *scopal* is used for both fixed and floating scopal EPs.

Consider the following bag of EPs for the example *Every hungry dog probably chases a cat*:

(1) {h1:Every(x,h7,h8), h2:Hungry(x) h2:Dog(x), h3:Probably(h9),
h4:Chase(x,y), h5:A(y,h10,h11), h6:Cat(y)}

This example shows the three kinds of EP: non-scopal: Hungry(x), Dog(x), Chase(x,y), Cat(y); quantifier EPs: Every(x,h2,h3), A(y,h9, h10); and fixed-scopal EP Probably(h6). In this example, h1...h6 are labels and h7...h11 are holes.

A group of EPs with the same label are called an EP *conjunction*. For example the EPs Dog(x) and Hungry(x) form an EP conjunction which can be thought as the semantic fragment Dog(x)∧Hungry(x) if interpreted in the first order logic. Every MRS has a unique hole called *(global) top handle* to mark the highest EP (or EP conjunction). There is also a set of *(handle) constraints* associated with every MRS that restrict how holes are equated with labels. Every handle constraint (or simply constraint) relates one hole to one label and is shown as h =q l, where h is a hole and l is a label. This handle constraint is satisfied iff either h = l or h = l1, where l1 is the label of a quantifier EP Q(x, h1, h2) and h2 =q l recursively holds. Handle constraints are also called *qeq* (equality modulo quantifier) relationships. In summary, an MRS structure (or simply an MRS) is a triple ⟨GT, R, C⟩ where GT is the global top handle, R is a bag of EPs and C is a set of handle constraints. As

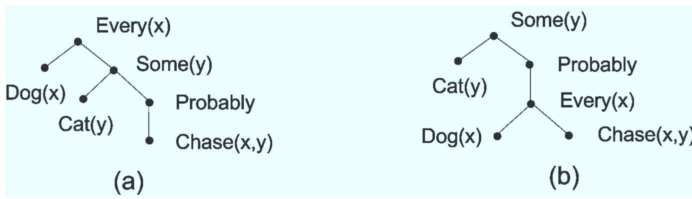

FIGURE 1 Two scope-resolved MRSs

an example, the complete MRS structure for the above sentence is:

(2) ⟨ h0, {h1:Every(x,h7,h8), h2:Hungry(x) h2:Dog(x), h3:Probably(h9), h4:Chase(x,y), h5:A(y,h10,h11), h6:Cat(y)},{h0 =q h3, h7 =q h2, h9 =q h4, h10 =q h6}⟩

Every MRS corresponds to a set of *scope-resolved* MRSs in which every hole is equated with some label and no label is equated with more than one hole. A scope-resolved MRS must form a tree of EPs (or EP conjunctions), in which dominance is determined by *outscoping*[1] relation, and must satisfy all the qeq relationships. For example the scope-resolved MRS in (3) can be obtained from the above MRS using the equalities h0=h1, h7=h2, h8=h5, h10=h6, h11=h3 and h9=h4.

(3) {h0:Every(x,h2,h5), h2:Hungry(x) h2:Dog(x), h3:Probably(h4), h4:Chase(x,y), h5:A(y,h6, h3), h6:Cat(y)}

Scope-resolved MRSs are usually represented as tree structures. For example the scoped-resolved structure in (3) is represented as the tree shown in figure (1a). An MRS is called *well-formed* if it corresponds to at least one scope-resolved structure. A scope-resolved MRS is called *bounded* if every non-scopal argument (i.e. x, y, ...) is in the scope of its quantifier. It can be easily verified that the above MRS has six bounded scope-resolved structures, two of which shown in figure (1a, b).

7.3 Semantic composition algorithm

Copestake et al. (2005) give a semantic composition algorithm which converts a syntactic tree to an MRS. In this section we introduce the notion of Canonical Form MRS (CF-MRS) and prove that every well-formed MRS structure that is generated by this algorithm is in this form.

A Canonical Form MRS (CF-MRS) is an MRS which satisfies following conditions:

[1] A label l (or its corresponding EP or EP conjunction) *immediately outscopes* a label l′ (or its corresponding EP or EP conjunction) iff l is the label of some EP P(...h...) and h=l′. *Outscopes* is the reflexive transitive closure of *immediately outscopes*.

- No quantifier EP is involved in an EP conjunction.
- The body hole of no quantifier EP is involved in any constraint;
- The label of no quantifier EP is involved in any constraint.
- Every other hole and label occurs in exactly one constraint.

Theorem 1 *Every well-formed MRS structure which is generated by MRS semantic composition algorithm is in canonical form.*

To prove this theorem, we need to describe the semantic composition algorithm. In order to do this, we define *partial* MRS to be a 4-tuple ⟨GT, LT, R, C⟩, where LT is a new handle called *local top*. As an initialization step, for every leaf of the syntactic tree (i.e. every word in the sentence), an MRS of the form ⟨h0, h1, {h2:P(...)}, {}⟩ is created; where the EP P comes from the lexicon and label h2 is a new distinct label. If P is floating scopal, h1 is a new distinct handle; otherwise, h1 = h2. Note that h0, the global top handle, would be the same for all the partial MRSs which are built during the semantic composition process. For example consider the sentence *Every hungry dog frequently barks*. (4) shows the partial MRS built for the floating scopal *Every*, fixed-scopal *Probably* and non-scopal EP *Bark*.

(4) ⟨h0,h1,{h2:Every(x,h3,h4)},{}⟩
⟨h0, h5, {h5: Frequently(h6)}, {}⟩
⟨h0, h7, {h7:Bark(x)}, {}⟩

Once a partial MRS is created for every leaf, the semantic composition algorithm moves up in the syntactic tree and for every interior node assigns the combination of its children's partial MRS to that node. There are two kinds of MRS combinations: scopal and intersective. Consider the two partial MRSs m1=⟨h0, lt1, R1, C1⟩ and m2=⟨h0, lt2, R2, C2⟩ and let m = ⟨h0, lt, R, C⟩ be their combination. If m1 has a scopal EP P(..., h, ...) which scopes over an EP in m2, the combination of m1 and m2 is a *scopal combination*, defined as:

(5) lt = lt1
R = R1 + R2
C = C1 ∪ C2 ∪ {h =q lt2}

Otherwise it is an *intersective combination*:

(6) lt = lt1 = lt2
R = R1 + R2
C = C1 ∪ C2

where h is the hole of the scopal EP in m1 and + means append. The definitions can easily be extended to the case where more than two MRSs are combined. It should be noted that the body hole of the quantifier EPs is ignored during semantic composition; that is no handle constraint for the body

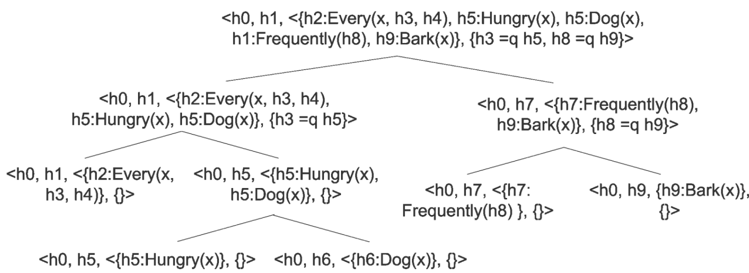

FIGURE 2 Semantic composition process for the sentence *Every hungry dog frequently barks.*

of the quantifier EPs is created. Figure (2) shows how the partial MRS for each interior node is built using scopal and intersective combinations.

Once the algorithm gets to the root, if the partial MRS for the root is ⟨h0, h1, R, C⟩, it outputs ⟨h0, R, C ∪ {h0 =q h1}⟩ as the final MRS for the whole sentence. (7) shows the final MRS built for the above example.

(7) ⟨h0, {h2: Every(x,h3,h4), h5: Hungry(x), h5: Dog(x), h1: Frequently(h8), h9:Bark(x)}, {h0 =q h1, h3 =q h5, h8 =q h9}⟩

Let M be the output of the semantic composition process for an arbitrary sentence with the syntactic tree T. Here we sketch the proof of theorem 1 by stating following propositions. We leave the detailed proof of this theorem to the longer version of this paper.

Proposition 2 *No quantifier EP in M is involved in an EP conjunction.*

Proposition 3 *Every hole in M is involved in at most one handle constraint. Furthermore, there is no handle constraint for the body hole of the quantifier EPs.*

Proposition 4 *Quantifier EP labels in M are not involved in any handle constraint. Every other label is involved in exactly one constraint.*

Proposition 2 and 3 directly result from the definition of the algorithm. To prove proposition 4, first we show the following lemma.

Lemma 5 *The local top of every node in the tree is involved in exactly one handle constraint in M.*

Proof. Consider a node v in the tree with the local top lt_v and the parent u. From the definition of combination, either lt_v is involved in a handle constraint in M_u (the MRS of the parent) or it is the local top of u as well. The same argument can be applied to the node u. If lt_v is not involved in any handle constraint until we hit the root, then the termination step of the algorithm adds the constraint h0=q lt_v. This proves that every local top is involved in at least one constraint. On the other hand, every local top can occur in at most one constraint because once a local top gets into a handle constraint, it cannot be the local top of the parent or any of its ancestors. □

From the above lemma and the fact that the labels of all EPs except quantifier EPs are a local top at the leaf level (refer to the initialization step of the algorithm) and that the quantifier EP labels are never a local top, proposition 4 is proved.

From propositions 2, 3 and 4, we can see that, the number of holes is greater than or equal to the number of labels in M (note that we count the global top handle as a hole). On the other hand, in every well-formed MRS structure number of labels is always greater than or equal to the number of holes.[2] These two facts lead us to the following proposition:

Proposition 6 *The number of holes is equal to number of labels in M or M is not well-formed.*

In conjunction with proposition 4 and 6, proposition 3 results in the following corollary:

Corollary 7 *If M is well-formed, every hole which is not the body hole of a quantifier is involved in exactly one constraint in M.*

Theorem 1 directly results from propositions 2, 3, and 4 and corollary 7.[3]

Although not mentioned in Copestake et al. (2005) it seems that in practice a slightly modified version of this algorithm is used, where in the scopal combination an equality constraint can be added instead of a normal handle constraint. This allows labels to occur as an argument of a scopal EP in an MRS. That is, an MRS can have two EPs of the form l1:P1(..h..) and l2:P2(...) with an equality h = l2. In this case, we can collapse the two EPs into one EP whose arguments are the union of the arguments of the two EPs excluding the hole h. For example the EPs l1:P1(x, h1, l2) and l2:P2(y, h3, h4) can be transformed into a single EP l1:P1-2(x, y, h1, h3, h4). While this transformation does not affect the number of interpretations, it conforms to theorem 1 and lets us keep the same definition for CF-MRS.

---

[2]this is true because in order to build a scope-resolved structure, every hole must be equated with some label and no label can be equated with more than one hole.

[3]We have made an implicit assumption in proving theorem 1 which needs to be clarified here. We have assumed that in the scopal combination (equation 5) lt2 is always a label; however this is not the case if the grammar is not linguistically meaningful (for example consider the case where we have a quantifier EP *Every* as a leaf node and a quantifier EP *Some* as its parent which scopes over *Every*). However in this case, in the final MRS M, there is a handle constraint between two handles where none of them is a label in M. This contradicts the definition of handle constraints which are required to relate one hole to one label; hence such an MRS can be considered an ill-formed MRS structure. As a result, theorem 1 remains valid even when the grammar is not linguistically meaningful.

7.4 Canonical Form MRS

In section 7.3, we defined the notion of canonical form MRS as a subset of MRS structures; however the definition of MRS given in section 7.2, following the strategy of standard MRS formalism, was not formal and precise. Here, in order to give a rigorous proof for the equivalence of qeq and outscoping relationship in CF-MRS, we need a mathematically clean definition of all the concepts. Therefore, in this section we define the notion of CF-MRS more formally as an independent concept. Note that most of the concepts, which were already defined in section 7.2, are redefined in this section in a more formal fashion.

Definition 1 A *CF-MRS* is a triple ⟨GT, R, C⟩ where R is a *set* of EPs as defined in section 7.2; GT is a unique hole which does not occur in any argument position in R[4] ; and C, the set of *(handle) constraints*, is a bijection from H-H_b to L-L_q, in which H and L are the set of all the holes and labels in R respectively; H_b is the set of body holes of the quantifiers; and L_q is the set of labels of the quantifier EPs. We require that every label and hole (except GT) occurs exactly once in R.[5]

In order to get a more intuitive representation, we introduce a graph representation for CF-MRS. The graph of a CF-MRS is a directed graph with two types of node and two types of edge. Every label/hole in the CF-MRS is represented as a single label/hole node in the graph. Solid edges connect the label of EPs to their holes and dotted edges represent handle constraints; that is every handle constraint (h, l) in C is represented as a dotted edge from hole node h to label node l. As an example, figure (3) represents the graph representation of the CF-MRS given in (8) for the sentence *Every dog probably chases some cat*.[6]

(8) ⟨h0, {l1: Every(x, h1, h2), l2: Dog(x), l3: Probably(h3), l4: Chase(x, y), l5: Some(y, h4, h5), l6: Cat(y)}, {(h0,l3), (h1,l2), (h3,l4), (h4,l6)}⟩

As shown in this figure, every label node in the graph is labeled with its corresponding EP, dropping the label and the handle-taking arguments when there is no ambiguity. Note that in the graphical representation, we order the solid outgoing edges of every node from left to right based on the position

[4]Note that by *hole* we mean a variable over the set of labels; therefore the set of holes of a CF-MRS is the set of handle-taking arguments plus the global top GT.

[5]This is not a limitation in CF-MRS as it is always possible to collapse the set of all the EPs which share the same label (i.e. an EP conjunction) to a single EP whose arguments is the union of all the arguments of all the EPs in the conjunction. For example the EPs l1:P1(x1, h1) and l2:P2(x2, h2) can be collapsed into one EP of the form P1-2(x1, x2, h1, h2) where P1-2 is a new relation corresponding to the conjunction of the two relations P1 and P2.

[6]We have removed the arrows from the edges in the graphical representation of CF-MRS and scope-resolved structures throughout this paper, since the direction is clear from the context.

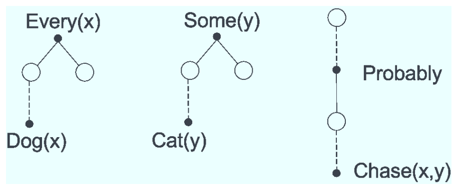

FIGURE 3 An MRS graph

of the corresponding handle-taking argument in the EP; for example the restriction hole of a quantifier always lies on the left side of its body hole in the graphical representation.

Definition 2 Every bijection from H, the set of holes, to L, the set of labels, is called a *label assignment*.

Definition 3 Given a CF-MRS M, a *scope-resolved* structure for M is the pair $\langle M, I \rangle$ where I is a label assignment which satisfies all the constraints in M.

Here, we give two different interpretations for a handle constraint. The first interpretation is the standard definition of handle constraints in MRS i.e. qeq relationships. As before, when interpreted as qeq relationship, a handle constrain (h, l) in C is represented as h =q l. A label assignment I satisfies this qeq relationship iff either I(h)=l or I(h)=l′, where l′ is the label of a quantifier EP Q(x, hr, hb) and I recursively satisfies hb =q l. The second interpretation of a handle constraint is *outscoping* relation, that is I satisfies the constraint (h, l) (shown as $h \leq l$ for this case) iff either I(h)=l or I(h)=l′ where l′ is the label of some EP P(...h′...) and I recursively satisfies $h' \leq l$. To differentiate between the two possible definitions of a scope-resolved structure, we define the two following versions of a scope-resolved structure.

Definition 4 We call a scope-resolved structure *standard* when handle constraints are treated as qeq relationships and call it *simple* when they are considered outscoping relations.

The graph of a scope-resolved structure is built by removing all the dotted edges in the original graph and merging every hole node h with the label node I(h). More precisely, the graph of a scope-resolved structure $\langle M, I \rangle$ is G=(V, E) where there is exactly one node v in V corresponding to every hole h_v in H and its corresponding label $l_v=I(h_v)$. Here, for the benefit of the section 7.5 and to emphasize that every node v in V corresponds to exactly one hole and one label of M, we represent the vertices of G as big circles with a dot in center. For example, figure (4) gives two scope-resolved structures for the CF-MRS given in (8). From the above definitions, it is easy to see that the

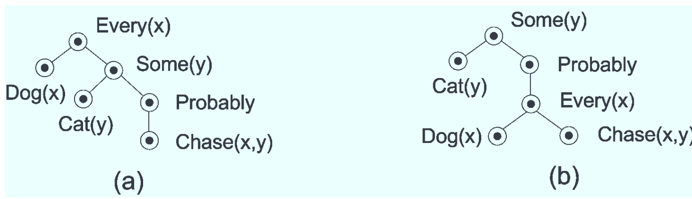

FIGURE 4 Two scope-resolved structures

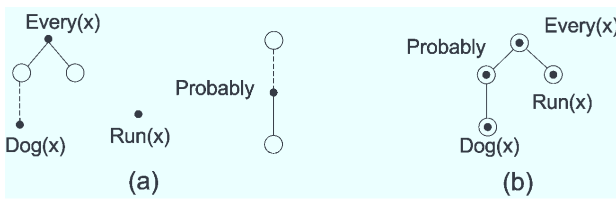

FIGURE 5 A simple but *not* standard scope-resolved structure

graph of a scope-resolved structure for every CF-MRS is always a tree.[7] In this paper, whenever we refer to the holes and labels of a scope-resolved tree or its subtrees we mean the holes and labels in M which correspond to the nodes of that tree/subtree. For example, the holes and the labels of the subtree rooted at the node *Probably* in figure (4a) are h5, h3 and l3, l4 respectively (refer to the CF-MRS given in (8)).

Note that both of the trees shown in figure (4) are both standard and simple scope-resolved structures. In section 7.5, we show that this is not a coincidence but for every CF-MRS structure this property holds; that is every simple scope-resolved structure is also a standard one and vice versa. Figure (5a,b) on the other hand shows the graph representation of an MRS (which is not a CF-MRS) and one of its simple scope-resolved structures that is not a standard one.[8]

Definition 5 A non-quantifier EP l1:P(...) is said to be *dependent* on the quantifier EP l2:Q(x, hr, hb) iff x is an argument of P. We say that a scope-resolved MRS M satisfies this *dependency constraint* iff l2 outscopes l1 in

[7]Since we order the outgoing solid edges of every label node in the CF-MRS graph, the graph of a scope-resolved structure is actually an *ordered* tree.

[8]Note that the graph representation and the concepts of simple and standard scope-resolved structures for a general MRS can be defined exactly in the same way as they were defined for CF-MRS.

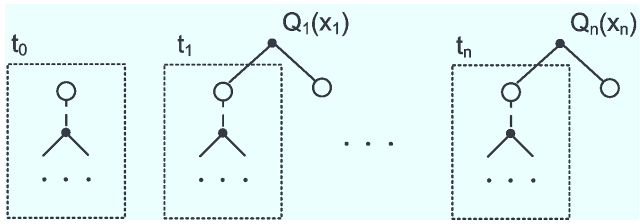

FIGURE 6 General structure of CF-MRS

M.[9]

Definition 6 A scope-resolved structure is called *bounded* if it satisfies all the dependency constraints carried by the non-scopal arguments of the EPs.

As the final point in this section, note that from the definition of CF-MRS every CF-MRS is a forest of exactly n+1 trees (n is the number of quantifiers) whose roots are the global top handle and the quantifier labels, as shown in figure (6).

7.5 Equivalence of qeq and outscoping relationships in CF-MRS

To prove this equivalence we need to prove the following theorem:

Theorem 8 *Given an arbitrary CF-MRS M=⟨h0, R, C⟩, T=⟨M, I⟩ is a simple scope-resolved structure if and only if T is a standard scope-resolved structure.*

The if direction is trivial as qeq relation always implies outscoping. In order to prove the only if direction, we use the following lemma:

Lemma 9 *Let T=⟨M, I⟩ be a simple scope-resolved structure and T′ be a subtree* [10] *of T with no quantifier's body hole. For every hole h in T′ we have I(h) = C(h)*[11] .

Proof. We prove this using induction on the depth of T′, d. If d = 0, T′ is a single leaf node u of T which corresponds to some hole h in M. Because h is not the body hole of some quantifier, there is some label l such that C(h)=l or equivalently h$\leq$l. Since u is a leaf node, this constraint is satisfied in T only if I(h)=l, which implies I(h)=C(h).

Now suppose that d>0, let u be the root of T′; u_1, u_2, ... u_k be the children of u and T_1, T_2, ...T_k be the subtrees of T rooted at u_1 ... u_k (figure 7). Based on the induction assumption, for every hole h′ in T_1 ... T_k, we have I(h′)=

[9] Refer to footnote 1 for the definition of outscoping.

[10] In this paper, by subtree of a tree T we mean a node with *all* of its descendants in T.

[11] Since C, the set of constraints, is a function; if (h, l) $\in$ C, we can refer to l as C(h).

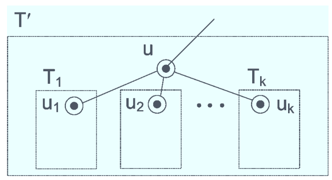

FIGURE 7 Inductive proof of lemma 9

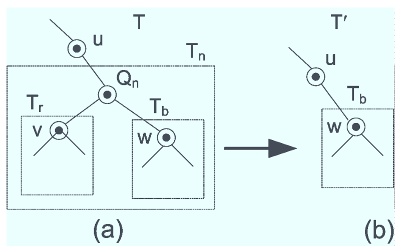

FIGURE 8 Inductive proof of theorem 8

C(h′). Therefore, all we need to show is that $I(h_u)=C(h_u)$, where h_u is the hole corresponding to the node u. Assume to the contrary that $I(h_u)\neq C(h_u)$. Because h_u is not the body hole of some quantifier, there is some label l, such that $C(h_u)$=l. In order to satisfy the constraint $h_u \leq l$, there must be some hole h′ in one of the subtrees T_1 ... T_k such that I(h′)=l. But we already saw that for every h′ in these subtrees I(h′)=C(h′). This implies $C(h')=I(h')=l=C(h_u)$ which is a contradiction because C is a one-to-one function. □

Proof of the main theorem: Let T be a simple scope-resolved structure T=⟨M, I⟩; using induction on n, number of quantifiers, we show that T is a standard scope-resolved structure as well. For n = 0, there is no quantifier in T, therefore according to lemma 9 for every hole h in T, I(h)=C(h) which means if we treat all the constraints as qeq relationship, I satisfies all these constraints, hence T is also a standard scope-resolved structure.

Now let n>0, and consider an arbitrary CF-MRS M with n quantifiers as it is shown in figure (6). There is a quantifier node in T=⟨M, I⟩ that does not outscope any other quantifiers (for example the deepest quantifier in T). Without loss of generality let's assume that Q_n has this property and call the tree rooted at Q_n in T T_n. We call the tree rooted at the left child of Q_n in T the restriction tree and the tree rooted at the right child of Q_n the body tree of Q_n and represent them by T_r and T_b respectively (figure 8).

There is no quantifier in T_r, intuitively it means that T_r is the tree t_n (figure 6) in which every hole is merged with its paired label. More precisely, according to lemma 9, for every hole h in T_r, I(h)=C(h). Therefore, if we treat all the constraints in t_n (refer to figure 6) as qeq relationships, these constraints are satisfied in T_r. Let's detach the tree T_n from T, replace it with T_b and call the new tree T′ (figure 8). It is easy to see that T′ is a simple scope-resolved structure for the CF-MRS M′ with n-1 quantifier (that is the CF-MRS shown in figure (6) without the whole tree rooted at Q_n). To see why, first note that there is a one-to-one correspondence between the nodes of T′ and the holes/labels in M′. Second, every handle constraint in M′ is satisfied in T, and hence is satisfied in T′ as well (because the transformation in figure 8 does not violate any *outscoping* relation).

T′ is a simple scope-resolved structure for M′; therefore based on the induction assumption, T′ is also a standard scope-resolved structure. It means that if we treat all the handle constraints in M′ as qeq relationship, they are all satisfied in T′. But moving from T′ to T (by replacing back the node Q_n and the subtree T_r, see figure 8) does not violate any qeq relationship which already holds in T′. On the other hand we already saw that if we treat all the handle constraints in t_n as qeq relationship they are satisfied in T_r. As a result if we treat all the handle constraints in M as qeq relationships, they are all satisfied in T. Hence T is also a standard scope-resolved structure.

7.6 Canonical Form Underspecified Representation

This result motivates a universal *Canonical Form Underspecified Representation* (CF-UR) similar to the CF-MRS structure in figure (6). In CF-MRS, however, the dependency constraints are encoded in the non-scopal arguments while in Hole Semantics and Dominance Constraints, all the constraints are explicitly expressed in the underspecified representation using outscoping constraints. The equivalence of qeq and outscoping relationship in CF-MRS allows us to do the same thing in CF-UR. We can remove all the non-scopal arguments and represent dependency constraints using outscoping constraints between the label of every quantifier EP and the label of the non-quantifier EPs which are dependent on that quantifier. Figure (9) shows the graph representation of the CF-MRS given in (8), in which both handle and dependency constraints are shown using dotted edges.

In addition, we label the dependency edges (i.e. the edges which correspond to the dependency constraints) with integers. This is necessary in order to keep the information that states which argument position in an EP is filled by which variable. In this example, the edge between *Every* and *Chase* is labeled 1 which shows that the variable quantified by *Every*, say x, fills the first non-scopal argument position of the EP *Chase*. In general, instead of num-

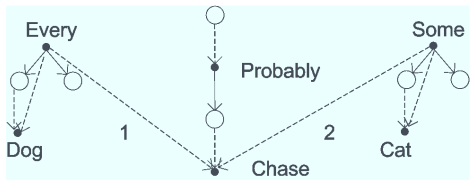

FIGURE 9 CF-MRS with explicit dependency constraints

bers we can label the dependency edges using a set of predefined roles. For example, the edge between *Every* and *Chase* can be labeled by the role *agent* and the edge between *Some* and *Chase* can be labeled by the role *theme*. For similar reasons, the edges from *Every* to *Dog* and from *Some* to *Cat* are necessary as they encode the non-scopal arguments of each predicate. However, since these predicates have only one argument we haven't shown the integer label for these two edges.

More formally we define a CF-UR as a 6-tuple $\langle$L, H, F, T, C, A$\rangle$ where L is a set of *labels*; H is a set of variables over labels called *holes*; F is a set of *labeled formulas* consisting of two types: the *predications* of form l_i:P_i(h_1, h_2, ... h_k) and the *quantifications* of form l'_i:Q_i(h'_1, h'_2), where l_i, $l'_i \in$ L and h_1, h_2, ...h_k, h'_1, $h'_2 \in$ H; and T is a unique hole in H called *top* which does not occur in any argument position in F. We require that every label and every hole (except T) occurs exactly once in F. Therefore in a CF-UR, no two formulas can be labeled by the same label and no two argument positions can be filled by the same hole. We define L_Q as the set of all the labels which label some quantification in F and L_P as the set of all other labels. We also define H_b as the set of all the holes which occur as the second argument position of some quantification (called *body holes*) and H_C as the set of all other holes. C is a relation over H U L and L, called the set of constraints; More precisely, C=C_HUC_L where C_H is a bijection from H_C to L_P and C_L is a relation over L_Q and L_P. Intuitively C_H and C_L are equivalent to the set of handle and dependency constraints in a CF-MRS respectively. Finally, A is a total function from C_L to ROLES, where ROLES is a set of predefined roles (such as {agent, theme, ...}). Intuitively, A specifies the role of the argument position which is encoded by every dependency constraint. ROLES can also be defined as the set of positive integers. In this case, A represents the argument position that every dependency constraint encodes and we need to force the condition that ((l1, l2), i) $\in$ A, only if for every 0<j<i, there is some label l such that ((l, l2), j) $\in$ A.

In order to have a more intuitive representation, we usually represent a CF-UR U=$\langle$L, H, F, T, C, A$\rangle$ as a directed graph G_U=(L$\cup$H, S$\cup$C) where S is the

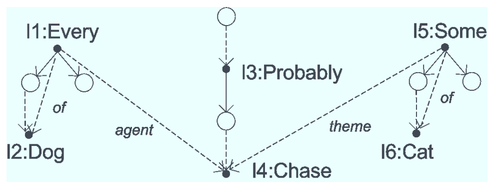

FIGURE 10 Graphical representation of CF-UR

set of all the pairs (l, h) such that h is an argument of the formula labeled by l. The nodes in L and H are represented as dots and holes respectively and the edges in S and C are represented using solid and dotted edges respectively. We label the dependency edges (i.e. label to label dotted edges) by their corresponding role (or argument position) specified by the function A. We also order the outgoing solid edges of every label node from left to right based on the position of the hole (in the corresponding labeled formula), to which the edge is connected. For example, the graph shown in figure (10) is the graphical representation of the CF-UR in (9) in which for the purpose of clarity we labeled every label node with its corresponding labeled formula.

(9) U=⟨{l1, l2, l3, l4, l5, l6}, {h0, h1, h2, h3, h4, h5}, {l1:Every(h1, h2), l2:Dog, l3:Probably(h3), l4:Chase, l5:Some(h4, h5), l6:Cat}, h0, {h0≤l3, h1≤l2, h3≤l4, h4≤l6, l1≤l2, l1≤l4, l5≤l4, l5≤l6 }, {((l1, l2), of), ((l1, l4), agent), ((l5, l4), theme), ((l5, l6), of)}⟩

Note that in this example, we have assumed ROLES is a set of predefined roles which includes the three roles *agent*, *theme* and *of*. As shown in (9) we usually represent the ordered pairs (x, y) in C as x≤y. From the above definitions, it is easy to see that the labels in L_Q and the hole T are the only roots (i.e. nodes with no incoming edge) of a CF-UR graph.
Given a CF-UR, any bijection from H to L is called a label assignment. A label assignment I satisfies a constraint x≤y iff

- when x is a hole and y is a label: either I(x)=y or I(x)=z and I recursively satisfies z≤y;
- when x and y are both labels: either x=y or x:F_i(...z..) is in F and I recursively satisfies z≤y;

A tuple ⟨U, I⟩ where U is a CF-UR and I is a label assignment which satisfies all the constraints in C is called a *solution* of U. As with the CF-MRS sometimes we use a graphical representation to represent a solution. The graph of a solution ⟨U, I⟩ is built by taking the graph of the CF-UR and merging every hole node h with the label node I(h).

For example, the CF-UR in (9) has 6 possible solutions, two of which are

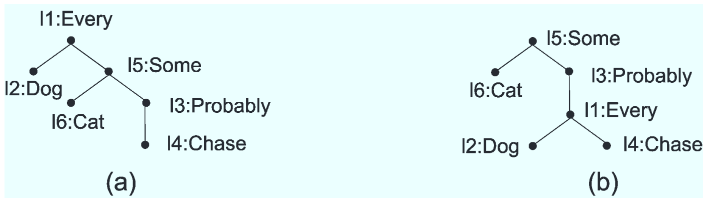

FIGURE 11 Two solutions for the CF-UR given in figure (9)

shown in figure (11a,b)[12] , one with I = {(h0, l1), (h1, l2), (h2, l5), (h3, l4), (h4, l6), (h5, l3)} and one with I = {(h0, l5), (h1, l2), (h2, l3), (h3, l1), (h4, l6), (h5, l4)}.

We have shown how a CF-MRS can be converted to a CF-UR. The inverse is straightforward. The corresponding CF-MRS of a CF-UR U=⟨L, H, F, T, C, A⟩ is a tuple M_U= ⟨T, R, C′⟩ in which R is the set of all the EPs of the form l_i:Q_i(x_i, hr_i, hb_i) where l_i:Q_i(hr_i, hb_i) is a quantification in F and EPs of the form l_k:P_k(x_i, x_j, ..., h1, h2, ...) where l_k:P_k(h1, h2, ...) is a predication in F; and x_i is an argument of the EP l_k:P_k(...) if and only if $l_i \leq l_k$ is in C. Finally C′ is a subset of C which includes all the hole to label (but not any label to label) constraints in C. Every (h, l) (or equivalently h≤l) in C is represented as h =q l in C′. Trivially M is a CF-MRS; hence the qeq relationships are equivalent to outscoping constraints in M. Using this fact, it can be easily seen that every bounded scope-resolved structure of M corresponds to exactly one distinct solution of U and vice versa. As a result, there is a one-to-one correspondence between the bounded scope-resolved structures of M and the solutions of U.

In a forthcoming paper we show the back and forth translation between CF-UR and the two other formalisms, Hole Semantics and Dominance Constraints.

7.7 Conclusion

We defined Canonical Form MRS and showed that the every well-formed MRS which is generated by this algorithm is in this form. We have shown for CF- MRS, qeq relationship is equivalent to dominance relationship. Based on this result we have proposed a universal underspecified representation, called Canonical Form Underspecified Representation or CF-UR. In a forthcoming paper, we will show how this single representation can be translated back and forth between the other semantic formalisms such as Dominance Constraints and Hole Semantics.

[12]Although dependency edges are part of the graph, for the purpose of clarity, we have only shown the solid edges in these figures.

Acknowledgment

This work was supported in part by a grant from National Science Foundation (#0748942), by the Defense Advanced Research Projects Agency (DARPA) under Contract No. FA8750-07-D-0185, and the Office of Naval Research grant No. N000140510314.

References

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

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

Bos, J. 2002. *Underspecification and resolution in discourse semantics.* PhD Thesis. Saarland University.

Copestake, A., D. Flickinger, C. Pollard, and I. Sag. 2005. Minimal Recursion Semantics: An Introduction. *Research,on Language and Computation* 3:281–332.

Copestake, A., A. Lascarides, and D. Flickinger. 2001. An Algebra for Semantic Construction in Constraint-Based Grammars. In *ACL-01 Toulouse, France*.

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

Fuchss, R., A. Koller, J. Niehren, and S. Thater. 2004. Minimal Recursion Semantics as Dominance Constraints: Translation, Evaluation, and Analysis. In *Proc. ACL-04 Barcelona, Spain*, pages 247–254.

Koller, A., J. Niehren, and S. Thater. 2003. Bridging the gap between underspecification formalisms: Hole semantics as dominance constraints. In *EACL-03*.

Niehren, J. and S. Thater. 2003. Bridging the Gap Between Underspecification Formalisms: Minimal Recursion Semantics as Dominance Constraints. In *ACL-03*.

Thater, S. 2007. *Bridging the Gap Between Underspecification Formalisms: Minimal Recursion Semantics as Dominance Constraints.* PhD Thesis. Universität des Saarlandes.