Dynamic control of scope underspecification

Markus Egg and Michael Kohlhase
Universität des Saarlandes, Computerlinguistik/Informatik

1 Introduction

We present an approach to a subject matter that has been discussed extensively in the literature, viz., the question of how to represent semantic structures with scope ambiguities in an underspecified way. In the following, we will focus our attention on examples like (1), which have two readings, as either quantifier may have scope over the other one:

(1) Every researcher visited a company

Many solutions have been proposed for an appropriate treatment of such examples. Some of them, e.g., UDRT (Reyle 1992), Cooper (1983), or Pereira (1990), invoke additional machinery to control the relation between the underspecified representation of sentences like (1) and its possible disambiguations. USDL (Pinkal 1996) can do without such additions to the representation formalism. However, USDL must take recourse to a metalanguage level to construct underspecified representations of structurally ambiguous sentences. In addition, the strict control over variable binding processes, which is guaranteed by other formalisms (e.g. type theory), is lost in the USDL metalanguage.

We tackle these problems by reformulating the USDL analysis of examples like (1) in λ-DRT (Kohlhase, Kuschert, and Pinkal 1996), a higher-order extension of DRT. Instead of λ-DRT itself, we use the dynamic λ-calculus $\mathcal{DLC}$ (Kohlhase and Kuschert 1997), which provides a uniform algebraic basis for λ-DRT and other compositional discourse logics. $\mathcal{DLC}$ provides the crucial type inference mechanism for strict variable control in our derivations. A dynamic base formalism allows us to express underspecified representations of scopally ambiguous sentences directly without recourse to a metalanguage.

Let us first review the original formulation. USDL is a semantic metalanguage. Its expressions denote object language expressions or operations. E.g., the constant $\mathbf{company}'$ denotes an expression in a suitable object language, say, '$\lambda x\ \mathbf{company}'(x)$' in a language of predicate logic. '@' stands for functional application, and 'L_x' models object-level λ-abstraction over a variable x. In contrast to λx the metalevel L_x may capture free occurrences of x. For scopally ambiguous sentences S, USDL yields a set of constraints on the meaning of S, which describes the set of unambiguous readings of S. These constraints involve context variables C_n which are instantiated during the calculation of the unambiguous readings from such a constraint set. For (1), USDL deducts the set (2) of constraints over its meaning X_0:

(2) $\{X_0 = C_5(\mathbf{visit}'@y@x),\ X_0 = C_3(\mathbf{a}'@\mathbf{company}'@L_y\ C_4(\mathbf{visit}'@y@x)),$
$X_0 = C_1(\mathbf{every}'@\mathbf{researcher}'@L_x\ C_2(\mathbf{visit}'@y@x))\}$

(2) means in prose that the sentence meaning X_0 is equal to some material (C_5) applied to the skeleton sentence meaning $\mathbf{visit}'@y@x$. In addition, for each of the two quantifiers, X_0 can also be described as involving the quantifier meaning applied to a λ-abstraction of a

constituent that contains the skeleton sentence meaning. This entails that each quantifier has scope over the verb that syntactically subcategorizes for it. The syntax-semantic interface alone ensures that the subscripts of the metalanguage lambda operators L_n match the variables x_n in the skeleton sentence meaning.

Applying a linear second-order unification algorithm (e.g. Niehren, Ruhrberg, and Pinkal 1997) to the constraint set (2) yields the substitution

$$\{C_1 = C_4 = \lambda X.X, C_2 = \lambda X.\mathbf{a}'@\mathbf{company}'@L_y(X), C_3 = \lambda X.\mathbf{every}'@\mathbf{researcher}'@L_x(X)\}$$

Applying this to the original set (2) yields the sentence semantics which denotes the reading of (1) with wide scope for the universal quantifier.

$$X_0 = \mathbf{every}'@\mathbf{researcher}'@L_x\ (\mathbf{a}'@\mathbf{company}'@L_y\ (\mathbf{visit}'@y@x))$$

2 The dynamic approach

A simple rewriting of the USDL analysis to act directly on standard predicate logic fails, as solutions for contexts cannot be substituted in the same way: The free variables of the skeleton sentence must not be captured in the process of variable instantiation. E.g., short of renaming the free x_j in the equation (3), the solution $C_2 = \lambda p \exists x_j(\mathbf{company}'(x_j) \wedge p)$ (with p of type t) could not be inserted into (3). In the USDL metalevel, this 'binding' was unproblematic due to the lack of binding constraints imposed by L_x in (2). The USDL analysis of examples like (1) relies crucially on this feature of the metalanguage.

(3) $X_0 = C_1(\forall x_i(\mathbf{researcher}'(x_i) \longrightarrow C_2(\mathbf{visit}'(x_i, x_j))))$

What we need is a dynamic formalism that allows the capturing of free variables. Such a formalism would allow us to take over the USDL analyses of scope underspecification directly. It must employ a concept of dynamic binding that differs from the one employed e.g. in Dynamic Predicate Logic (DPL, Groenendijk and Stokhof 1991) in two respects:

- it assigns a certain amount of dynamicity to universal quantifiers
- its notion of dynamicity is fine-grained, to distinguish dynamicity of existential and universal quantifiers

We recast the USDL account of scope ambiguity in λ-DRT (Kohlhase, Kuschert, and Pinkal 1996), a higher-order extension of DRT. As in USDL, we render the meaning of a sentence by a constraint set over this meaning, which can describe more than one unambiguous reading. Deriving unambiguous readings from such constraint sets, in particular, the problematic case of quantifier insertion (as in (3)), is easy in $\mathcal{DLC}$.

Underspecification enters by giving the meaning of constituents (especially, of sentences) in terms of equations in λ-DRT. This formalism combines the Montagovian type theoretic framework (Montague 1974) with dynamic approaches, such as DRT: it uses β-reduction to compute a first-order DRT representation of sentences from higher-order lexical items.

The capturing of free variables, *the* thing impossible in pure λ-calculus, is the driving force of the establishment of anaphoric binding in λ-DRT. We will use this very possibility of capturing variables in a controlled way to formalize quantifier underspecification.

Next, we will sketch $\mathcal{DLC}$ (the implementation of λ-DRT in this paper), and give an example of using higher-order unification in $\mathcal{DLC}$ to represent quantifier underspecification.

3 Dynamic λ-Calculus

The central idea of $\mathcal{DLC}$ is extending the type system to incorporate information on variables. Types of $\mathcal{DLC}$ expressions have a 'mode' part that lists their free variables (that can be captured) and their binders (that can dynamically capture free variables). Modes annotate variables with '-', and binders, with '+'. E.g., the type $X^-, U^+ \# t$ describes propositions with a free variable X that dynamically bind a variable (and introduce a discourse referent) U. Since $\mathcal{DLC}$ is a λ-calculus, the set of types is closed under function types.

$\mathcal{DLC}$-formulae are λ-calculus formulae together with dynamic abstractions of the form $\delta U.\mathbf{A}$. $\mathcal{DLC}$-well-typedness is verified (from type assumptions in a signature Σ and a variable context $\mathcal{A}$) using a variant of the classical λ-calculus inference system: $\mathbf{A}$ is a well-formed $\mathcal{DLC}$ formula of type α, iff the judgement $\mathcal{A} \vdash \mathbf{A}:\alpha$ is provable in the λ-calculus inference system with an added rule for δ-abstractions and where the λ-abstraction rule has been modified to manipulate the mode information in the types as indicated below.

$$\frac{\mathcal{A}, [U:U^- \# \overline{\overline{\beta}}] \vdash \mathbf{A}:\alpha}{\mathcal{A}, [U:U^- \# \overline{\overline{\beta}}] \vdash \delta U_\beta.\mathbf{A}:U^+ \# \alpha} \qquad \frac{\mathcal{A}, [X:\Gamma^- \# \overline{\overline{\beta}}] \vdash \mathbf{A}:\alpha \quad X \notin \mathbf{Dom}(\Gamma)}{\mathcal{A} \vdash (\lambda X_\beta.\mathbf{A}):(\Gamma^- \# \overline{\overline{\beta}}) \to \alpha}$$

$\overline{\overline{\alpha}} = \alpha$, and $\overline{\overline{\alpha \to \beta}} = \Gamma \# \alpha \to \Delta \# \beta$ for new modes Γ and Δ. In the merging of modes, positively signed variables overwrite negatively signed ones of the same type and name. This overwriting is important for the last rule for *dynamic abstraction*, where U^- may have been present in Γ, indicating the presence of a free variable U in $\mathbf{A}$. It is this overwriting effect that models the variable capturing at function application on the syntactic side.

The types of λ-abstractions explicitly pass on mode information from arguments of applications. Thus, the rule discharges the type assumption about the variable X from the variable context, since it is no longer free in the abstraction. Read backwards, this move augments the variable context with type information for X. The set of possible types of X_β in $\lambda X_\beta.\mathbf{A}$ is $\Gamma^- \# \overline{\overline{\beta}}$. The Γ^- part stands for the (as yet unknown) set of free variables of the argument. If the argument is itself functional, $\overline{\overline{\beta}} \neq \beta$, which reflects the fact that we do not know yet how the argument affects the free variables of its own argument within $\mathbf{A}$. (For details, see Kohlhase and Kuschert 1997.) The rule also enforces a kind of occurs-check for X in Γ^-. Finally, we assume the type identity $\Gamma \#(\Delta \# \alpha) = \Delta, \Gamma \# \alpha$ for all Γ, Δ, α.

Given this, we attain λ-DRT by fixing the set of base types to $\mathcal{BT} = \{e, t\}$ (individuals and truth values) and defining a set of connectives as conjunction, disjunction, negation and implication. In standard DRT, the binding properties of discourse referents in the context must be defined in an accessibility relation. With the extended type system, we can now express this relation directly in the syntax. E.g., we define a collection of dynamic implication operators of type $\Rightarrow_{\Gamma\Delta}:(\Gamma \# t) \to (\Delta \# t) \to ((\Gamma^-, (\Delta^- \setminus \Gamma^+)) \# t)$. where Γ^- is the set of negatively signed variables in the mode Γ. The mode of the type that results from applying the two arguments to the operator, $\Gamma^-, (\Delta^- / \Gamma^+)$, captures the accessibility relation between the two arguments: apart from the free variables of the first argument of the quantifier, only those free variables of the second argument are available to the outside which are not bound by the discourse referents of the first argument.

4 Example

We will now apply our analysis to sentence (1). Just like USDL, it generates the constraints

$$X_0 = C_1(\mathsf{er}(C_3(\mathsf{vis}UV))) = C_2(\mathsf{ac}(C_4(\mathsf{vis}UV)))$$

where variables C_i have types $\overline{\overline{t \to t}} = (\Gamma_i \# t) \to (\Xi_i \# t)$; 'er' and 'ac' abbreviate the semantics of *every researcher* and *a company*. We assume the following lexicon:[1]

word	formula	type
visited	vis	$\Gamma^- \# e \to \Delta^- \# e \to \Gamma^- \cup \Delta^- \# t$
sleeps	sleeps	$\Gamma^- \# e \to \Gamma^- \# t$
researcher	res	$\Gamma^- \# e \to \Gamma^- \# t$
company	comp	$\Gamma^- \# e \to \Gamma^- \# t$
every	$\lambda R_{e\to t}.\lambda Q_t.(\delta U_e.RU) \underline{\Rightarrow} Q$	$(\Gamma \# e \to t) \to (\Delta, U^- \# t) \to (\Gamma^- \cup (\Delta^- \setminus \Gamma^+)) \# t)$
a	$\lambda R_{e\to t}.\lambda Q_t.\delta V_e.RV \otimes Q$	$(\Gamma \# e \to t) \to (\Delta, V^- \# t) \to (\Gamma \cup \Delta, V^+) \# t)$

By the added expressivity of $\mathcal{DLLC}$ we can express the behaviour of binders and variables in the types of $\mathcal{DLLC}$ expressions. As functional application and instantiation of mode variables must be well-typed, the linear higher-order unification and the simplification of type constraints induced by type inference mutually constrain each other, giving us a powerful mechanism for propagating information about variables and quantifiers binding them.

In addition to the constraints taken over from USDL, we can use $\mathcal{DLLC}$ types to encode the general observation that, at text level, there may be no free variables except those introduced by deictic pronouns. (Single sentences qualify as a text, too, if they occur in isolation.) This entails that if the type of the sentence (1) is $\Theta \# t$ then $\Theta^- = \emptyset$, and, in particular, the variables U and V are no longer free at the sentence level.

Let us now calculate the type constaints Γ_i and Ξ_i imposed by the equational constraints: $\mathsf{vis}UV$ is of type $U^-, V^- \# t$, hence, $\Gamma_3 = \Gamma_4 = U^-, V^-$. The domain type of er is Δ, U^- and that of ac is Δ', V^-, so $\Xi_3 = \Delta, U^-$ and $\Xi_4 = \Delta', V^-$. Analogously we obtain values for the type of C_1 and C_2, which entails the following type judgements.

$$\begin{array}{|ll|}\hline C_1\colon (\Delta^- \# t) \to (\Theta \# t) & C_2\colon (\Delta', V^+ \# t) \to (\Theta \# t) \\ C_3\colon (U^-, V^- \# t) \to (\Delta, U^- \# t) & C_4\colon (U^-, V^- \# t) \to (\Delta', V^- \# t) \\ \hline \end{array}$$

Now we start linear unification, calculating and simplifying the type constraints after each step. First, we project the variable C_1, i.e., we guess that $C_1 = \lambda Z_t.Z$. This yields the wide scope reading of *every researcher* for (1). Linear imitation for C_1 is also possible and yields the narrow scope reading. Since $\lambda Z_t.Z$ has type $(\Psi \# t) \to (\Psi \# t)$, we have

$$\boxed{\Delta^- = \Theta \qquad C_1\colon (\Theta \# t) \to (\Theta \# t)}$$

[1] Since we omit modification here, we assume simplified types of nouns and verbs. We enforce non-degenerate binding for quantifiers that dynamically bind some variable U, by adding the mode U^- to the type specification of the mode Δ in the above representations of quantifiers. This expresses that they require a free variable U in their second argument.

Thus even at this early stage of the computation, we can see that

- Θ is empty, because by $\Delta^- = \Theta$ it contains no positive variables and by the status of (1) as a text it contains no negative variables. I.e., X_0 has neither free variables nor binding potential. This was to be expected, as the reading of (1) we are looking at is a text in which the universal quantifier *every researcher* has widest scope.
- the value of C_3 binds V: V^- is in the argument but not the target type mode of C_3.

In the next unification step, we imitate C_2, since the projection would lead to immediate failure, i.e. we bind C_2 to $\lambda Z_t.\mathsf{er}(C_5 Z)$, where C_5 is a new variable of characteristic type (i.e., the type without mode) $t \to t$. Type inference reveals that the type of C_5 is

$$\boxed{C_5\colon (\Delta', V^+\#t) \to (\Delta, U^-\#t)}$$

After decomposition the problem has the form $C_3(\mathsf{vis}UV) = C_5(\mathsf{ac}(C_4(\mathsf{vis}UV)))$. We continue by projecting ($C_5 = \lambda Z_t.Z$) and get the type constraint $\Delta', V^+\#t = (\Delta, U^-\#t)$ (argument and target types of $\lambda Z.Z$ must be identical) or equivalently for a new mode Φ.

$$\boxed{\Delta' = \Phi, U^- \qquad \Delta = \Phi, V^+}$$

Now we imitate C_3 by binding it to $\lambda Z_t.\mathsf{ac}(C_6 Z)$. C_6 is a new variable of type

$$\boxed{C_6\colon U^- V^-\#t \to (\Phi, U^-, V^-\#t)}$$

Decomposition yields $C_6(\mathsf{vis}UV) = C_4(\mathsf{vis}UV)$. If we guess that $C_4 = C_6 = \lambda Z_t.Z$, the completion of the linear unification is trivial. Type inference gives us the constraints

$$\boxed{\Phi = \emptyset \qquad \Delta = V^+ \qquad \Delta' = U^-}$$

Collecting all constraints makes all the dynamic potentials explicit:

variable	value	type
C_1	$\lambda Z_t.Z$	$t \to t$
C_2	er	$(U^-, V^+\#t) \to t$
C_3	ac	$(U^-, V^-\#t) \to (U^-, V^+\#t)$
C_4	$\lambda Z_t.Z$	$(U^-, V^-\#t) \to (U^-, V^-\#t)$
X_0	(er(ac(vis U V)))	t

5 Conclusion and further work

To sum up, in this paper we have presented an approach to the underspecified representation of scopally ambiguous sentences that accounts for them directly in terms of a dynamic lambda calculus, without having to rely on additional machinery. Deriving fully specified readings from underspecified representations involves only simple substitutions of typed context variables. The binding potential of expressions is highlighted throughout, in the derivation and in the resolution of the constraints on sentence meanings.

So far, we used the strict record on binders and variables merely to make the treatment of binders and variables explicit. But we expect the proposed approach to prove useful in restricting potential ambiguities within texts. Consider e.g. forcing wide scope of the existential quantifier in an example like (4) by picking up the referent of the quantifier anaphorically in the next sentence. This can easily be described in our formalism:

(4) Every researcher visited a company. It belonged to a tycoon.

Omitting deictic use of pronouns, we request (complete) texts to contain no free variables, that is to say, all free variables introduced in the text must be bound off within it as well. Hence, if the type of the second sentence in (4) is $\Delta^+, U^-\#t$ (the free variable U is introduced by the pronoun *it*), there has to be a preceding sentence in the text whose mode comprises a binder for U, i.e., is of type $\Gamma, U^+\#t$, for any Γ. In (4), this restricts the ambiguity of the first sentence in the desired way: It must provide a binder for the variable as introduced by the pronoun, which excludes the reading in which the indefinite quantifier has narrow scope w.r.t. the universal quantifier. Scoping a universal quantifier over an existential would have 'shadowed off' the binding potential of the existential quantifier.

The proposed approach will also be of use in the domain of nested quantifiers: It provides strict control over variables and their binders, which are needed to rule out overgeneration (as noted by e.g. Nerbonne 1992). Our approach provides this control and therefore blocks overgeneration even in examples like (5):

(5) Everyone with a picture of himself arrived

The unwanted solution with wide scope of the existential quantifier is ruled out: Its mode would still contain a free variable, which clashes with the presupposed type for texts.

References

Cooper, R. (1983). *Quantification and syntactic theory.* Dordrecht: Reidel.

Groenendijk, J. and M. Stokhof (1991). Dynamic predicate logic. *Linguistics & Philosophy 14*, 39–100.

Kohlhase, M. and S. Kuschert (1997). *Dynamic λ-Calculus.* CLAUS-Report 92, Uni. d. Saarl.

Kohlhase, M., S. Kuschert, and M. Pinkal (1996). A type-theoretic semantics for λ-DRT. In P. Dekker and M. Stokhof (eds), *Proc. 10th Amsterdam Colloquium*, 479–498. ILLC.

Montague, R. (1974). The proper treatment of quantification in ordinary English. In R. Montague (ed.), *Formal Philosophy. Selected Papers.* New Haven: Yale University Press.

Nerbonne, J. (1992). A feature-based syntax/semantics interface. DFKI Report RR-92-42.

Niehren, J., P. Ruhrberg, and M. Pinkal (1997). On equality up-to constraints over finite trees, context unification and one-step rewriting. In *Proceedings of CADE'97.*

Pereira, F. (1990). Categorial semantics and scoping. *Computational Linguistics 16*, 1–10.

Pinkal, M. (1996). Radical underspecification. In P. Dekker and M. Stokhof (eds), *Proceedings of the 10th Amsterdam Colloquium*, 587–606. ILLC, Amsterdam.

Reyle, U. (1992). Dealing with ambiguities by underspecification: a first order calculus for unscoped representations. In *Proceedings of the Eighth Amsterdam Colloquium.*