ACHIEVING EXPRESSIVE COMPLETENESS AND COMPUTATIONAL EFFICIENCY FOR UNDERSPECIFIED SCOPE REPRESENTATIONS

CHRIS FOX AND SHALOM LAPPIN

Department of Computer Science
University of Essex
foxcj@essex.ac.uk

Department of Philosophy
King's College London
shalom.lappin@kcl.ac.uk

The tension between expressive power and computational tractability poses an acute problem for theories of underspecified semantic representation. In previous work we have presented an account of underspecified scope representations within Property Theory with Curry Typing (PTCT), an intensional first-order theory for natural language semantics. Here we show how filters applied to the underspecified-scope terms of PTCT permit both expressive completeness and the reduction of computational complexity in a significant class of non-worst case scenarios.

1. Introduction

In Fox and Lappin (2005a) we propose Property Theory with Curry Typing (PTCT) as a formal framework for the semantics of natural language. PTCT allows fine-grained distinctions of meaning without recourse to modal notions like (im)possible worlds. It also supports a unified dynamic treatment of pronominal anaphora and VP ellipsis, as well as related phenomena such as gapping and pseudo-gapping.

PTCT consists of three sublanguage components. The first component encodes a property theory within a language of terms (an untyped λ-calculus). The second adds dynamic Curry typing (Curry and Feys, 1958) to provide a system for expressing type judgements for terms. The third uses a first-order logic to specify the truth-conditions of the propositional subpart of the term language. Our semantic representation language is first-order in character, rather than higher-order. We achieve the sort of expressive power previously limited to higher-order theories within a formally more constrained system. This provides an effective procedure for modelling inference in natural language.

In Fox and Lappin (2005a,b) product types are used to generate underspecified semantic representations within PTCT, the representation language, rather than through meta-language devices, which are invoked in most current treatments of underspecification (Reyle, 1993; Bos, 1995; Blackburn and Bos, 2005; Copestake et al., 1997). The expressive power of the language permits the formulation of filters on scope readings that cannot be captured in other theories of underspecification which rely on special purpose extra-linguistic operations and a weak system for constraint specification.

These filters on underspecified scope terms can solve the problem of expressive incompleteness that Ebert (2005) raises for other theories of underspecification. They can also be used to reduce the complexity involved in computing the set of possible scope readings that an underspecified term generates.

2. PTCT

PTCT is a first-order sysem in which types and propositions are terms over which we can quantify. This allows rich expressiveness whilst restricting the system to first order resources (Fox and Lappin, 2005a, Chapter 9).

The language of terms is the untyped λ-calculus, enriched with logical constants. It is used to *represent* the interpretations of natural language expressions. It has no internal logic, but when we add a proof theory, the simple language of types together with the language of terms can be combined to produce a Curry-typed λ-calculus.

The syntactic rules of PTCT are flexible. They allow the generation of syntactic expressions that have no intuitively meaningful interpretation. This does not undermine the system. The rules give a minimal characterisation of the syntax while our proof theory and our model theory characterise the proper subset of well-formed PTCT terms that constitute meaningful expressions.

In the first-order language of wffs we formulate type judgements for terms, and truth conditions for those terms judged to be in Prop.

3. Underspecified Representations in PTCT

Generalised quantifiers (GQs) represent noun phrases. We follow Keenan (1992) and van Eijck (2003) in taking a GQ to be an arity reduction operator that applies to a relation r to yield either a proposition or a relation r' that is produced by effectively saturating one of r's argument with the GQ.

We specify a family of functions $perms\_scope_k$ (where $k > 1$) that generate all $k!$ indexed permutation products of a k-ary indexed product term $\langle t_1, \ldots, t_k \rangle$ as part of the procedure for generating the set of possible scope readings of a sentence.

For our treatment of underspecification, $perms\_scope_k$ needs to take a k-ary product of scope taking elements (by default, in the order in which they appear in the surface syntax) and a k-ary relation representing the core proposition as its arguments. The scope taking elements and the core representation can be combined into a single product, e.g. as a pair consisting of the k-tuples of quantifiers as its first element and the core relation as its second. The permutation function $perms\_scope_k$ produces the $k!$-ary product of scoped readings. When a k-tuple of quantifiers is permuted, the λ-operators that bind the quantified argument positions in the core relation are effectively permuted in the same order as the quantifiers in the k-tuple. This correspondence is necessary to preserve the connection between each GQ and its argument position in the core relation across scope permutations.

A scope reading is generated by applying the elements of the k-tuple of quantifiers in sequence to the core relation, reducing its arity with each such operation until a proposition results. The ith scope reading is identified by projecting the ith element of the indexed product of propositions that is the output of our $perms\_scope_k$ function. Therefore, the PTCT term consisting of the application of $perms\_scope_k$ to an input pair of a k-tuple of GQs and a core relation provides an underspecified representation of the sentence corresponding to this term.

4. Filters and Expressive Completeness

Scope constraints can be formulated as filters on the $k!$-tuple of permutations of the form $\langle\langle Qtuple_1, Rel_1\rangle, \ldots, \langle Qtuple_{k!}, Rel_{k!}\rangle\rangle$ that $perms\_scope_k$ generates for an argument pair $\langle Qtuple_1, Rel_1\rangle$. Each such filter is a Boolean property function that imposes a condition on the elements of the $k!$-tuple.

Underspecified representations can be disambiguated by information acquired through subsequent discourse. So, for example, resolving anaphoric expressions like pronouns and definite descriptions in sentences following a statement that exhibits scope ambiguity may eliminate certain readings of the antecedent.

(1) A: Every student wrote a program for some professor.
(2) B: Yes, I know the professor. She taught the Haskell course.
(3) C: I saw the programs, and they were all list-sorting procedures.

Identifying "*some professor*" in (1) as the antecedent for "*the professor*" and "*she*" in (2) gives "*some professor*" scope over "*every student*" in (1). Interpreting "*a program*" in (1) as the antecedent for "*the programs*" and "*they*" in (3) causes "*a program*" to have narrow scope relative to "*every student*" in (1). Therefore, taken conjointly (2) and (3) forces on (1) the fully resolved scope order

$$\langle\text{“\textit{some professor}”}, \text{“\textit{every student}”}, \text{“\textit{a program}”}\rangle$$

Assume that "*every student*" $= Q_1$, "*a program*" $= Q_2$, and "*some professor*" $= Q_3$. We can formulate the filters contributed by (2) and (3) as (4) and (5), respectively (where GQ in $\hat{=}_{GQ}$ abbreviates the appropriate type of Q_i). In these filters we take $\langle Quants, Rel\rangle$ to be a variable ranging over pairs in which $Quants$ is a k-tuple and Rel is a k-ary relation. As the k-tuples are indexed, there is a one-to-one correspondence between the elements of a k-tuple and their respective indices. Let $tuple\_elem(i, Quants) = Q_i$ if Q_i is the ith member of $Quants$, and the distinguished term ω otherwise.

(4) $\lambda\langle Quants, Rel\rangle[\hat{\forall} i \epsilon \mathsf{Num} \hat{\forall} j \epsilon \mathsf{Num}((tuple\_elem(i, Quants) \hat{=}_{GQ} Q_3 \mathbin{\hat{\wedge}} tuple\_elem(j, Quants) \hat{=}_{GQ} Q_1) \mathbin{\hat{\rightarrow}} i \mathbin{\hat{<}} j)]$

(5) $\lambda\langle Quants, Rel\rangle[\hat{\forall} i \epsilon \mathsf{Num} \hat{\forall} j \epsilon \mathsf{Num}((tuple\_elem(i, Quants) \hat{=}_{GQ} Q_1 \mathbin{\hat{\wedge}} tuple\_elem(j, Quants) \hat{=}_{GQ} Q_2) \mathbin{\hat{\rightarrow}} i \mathbin{\hat{<}} j)]$

We specify the function $filter\_tuple(\langle F, T\rangle)$ which maps a pair consisting of a j-tuple F of filters and a k-tuple T to a k'-tuple (possibly the empty tuple) of all the elements of T that satisfy each filter in F. We construct a PTCT term of the form (6) to represent the k'-tuple obtained by applying the elements of F to the $k!$-tuple that is the value of $perms\_scope_k(\langle Quants_k, Rel\rangle)$.

(6) $filter\_tuple(\langle F, perms\_scope_k(\langle Quants_k, Rel\rangle)\rangle)$

Ebert (2005) shows that most current theories of underspecification are expressively incomplete to the extent that they cannot identify the proper subset of possible scope readings specified by Boolean operations other than conjunction, and in particular by negation. He cites the following example to illustrate the problem.

(7) Every market manager showed five sales representatives a sample.

Ebert stipulates that, in his example, real world knowledge allows all scope permutations except the one corresponding to $\langle \exists, 5, \forall \rangle$, where *a sample* takes wide scope, *five sales representatives* intermediary position, and *every market manager* narrow scope. He demonstrates that storage (Cooper, 1983; Pereira, 1990), hole semantics (Bos, 1995; Blackburn and Bos, 2005), Minimal Recursion Semantics (Copestake et al., 1997), and Normal Dominance Conditions (Koller et al., 2003) cannot formulate underspecified representations that express the set containing only the five remaining scope readings.

By contrast it is straightforward to formulate a filter in PTCT that rules out the problematic scope sequence in Ebert's case while permitting the five other readings.

(8) $\lambda \langle Quants, Rel \rangle [\hat{\forall} i \epsilon \mathsf{Num} \hat{\forall} j \epsilon \mathsf{Num} \hat{\forall} k \epsilon \mathsf{Num}((tuple\_elem(i, Quants) \hat{=}_{GQ} Q_{\exists}$
$\hat{\wedge}\ tuple\_elem(j, Quants) \hat{=}_{GQ} Q_5$
$\hat{\wedge}\ tuple\_elem(k, Quants) \hat{=}_{GQ} Q_{\forall}) \hat{\rightarrow} \hat{\sim}(i \hat{<} j \hat{\wedge} j \hat{<} k))]$

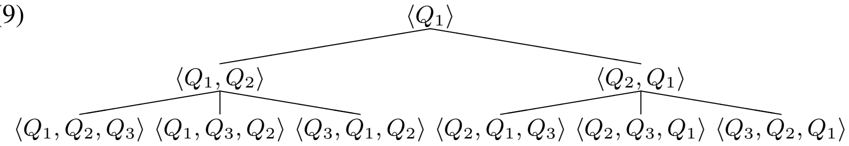
PTCT is, in principle, able to achieve expressive completeness in Ebert's (2005) sense.

5. Efficient Computation of Possible Scope Readings

At first glance it might seem that it is, in general, necessary to generate the full $k!$-tuple that is the value of $perms\_scope_k(\langle Quants_k, Rel \rangle)$ before applying the filters of F to the elements of this $k!$-tuple in order to compute the value of (6). Fortunately, this is not the case.

In Fox and Lappin (2005c) we present a tree construction algorithm for generating all possible permutations of a k-tuple. If this algorithm takes as its input the triple $\langle Q_1, Q_2, Q_3 \rangle$, then it generates the following tree.

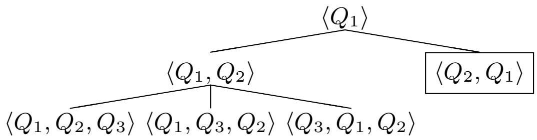
(9)



Filters can apply as constraints to nodes in the tree as the algorithm produces them. If a node violates a filter, then it is deleted, and the subtree that it dominates is not generated. In this way filters can reduce the size of the tree, and so limit the search space of possible scope readings that are explored for underspecified-scope terms $perms\_scope_k(\langle Quants_k, Rel \rangle)$ to a proper subset of the elements of the $k!$-tuple that is its value.

So, for example, the filter $Q_1 < Q_2$ prunes the tree in (9) to give the one in (10).

(10)



Identifying the size of a tree with the number of its nodes, we can compute the size of a tree T, $|T|$, through the formula

(11) $|T| = \Sigma i!$, where i is the index of the ith element of the initial k-tuple which the algorithm takes as its input.

Therefore, the size of the tree in (9) is $1! + 2! + 3! = 9$. The size of the tree in (10) is 6, which is a reduction of 30%.

The size of a subtree ST dominated by a node n at level i, but not including n, is given by the formula

(12) $|ST| = \Pi j \ (i < j \leq k) + \Sigma j' \ (i < j' < k)$.

Consider the quadruple $\langle Q_1, Q_2, Q_3, Q_4 \rangle$. The tree algorithm produces an indexed $k!$-tuple of 24 k-tuples as the leaves of a tree T_4 with 4 levels and 33 nodes. If a filter like $Q_1 < Q_2$ applies at level 2, the first branching node of T_4, it prunes the right-half of T_4 under $\langle Q_2, Q_1 \rangle$, and so it eliminates a subtree of 15 nodes, reducing T_4 by 15/33 = 45.4%. The remaining left side of T_4 has the three nodes $\langle Q_1, Q_2, Q_3 \rangle$, $\langle Q_1, Q_3, Q_2 \rangle$, $\langle Q_3, Q_1, Q_2 \rangle$ at level 3. If the filter $Q_2 < Q_3$ applies at this level, the 8 leaf nodes under $\langle Q_1, Q_3, Q_2 \rangle$ and $\langle Q_3, Q_1, Q_2 \rangle$ are pruned. Therefore, the conjunction of the filters $Q_1 < Q_2$ and $Q_2 < Q_3$ reduces T_4 by 15 + 8 = 23 nodes, which is (approximately) 70% of the full tree.

It is not difficult to construct a plausible case in which the interpretation of a sentence containing four quantified NPs is disambiguated by a conjunction of two filters of this kind through anaphora resolution in subsequent discourse, as in **A:** "It's amazing. A critic recently reviewed two plays for every newspaper in a major city." **B:** "Yes, he published the same reviews of *A Midsummer Night's Dream* and *New-Found-Land* in every major paper in New York last week."

Clearly, the earlier in the tree construction process (the higher up in the tree) that a filter applies, the greater the reduction in search space of possible scope readings that it achieves. It is also possible to optimise the interaction of filters and the tree construction algorithm by specifying a procedure that reorders the elements of the input k-tuple to permit the filters to apply at the earliest point in the generation of the tree. So, for example, if the algorithm takes as its input the triple $\langle Q_1, Q_2, Q_3 \rangle$ and one of the filters that apply to this triple is $Q_2 < Q_3$, then the reordering operation will map the triple into $\langle Q_2, Q_3, Q_1 \rangle$. We will leave the formulation of this operation for future work.

Ebert (2005) proves a theorem that entails that if a theory is expressively complete, then it will, in the worst case, produce a combinatorial explosion equivalent to generating all $k!$ scope readings for a sentence. This result holds for PTCT in the limit case, where no filters have been applied to a $perms\_scope_k(\langle Quants_k, Rel \rangle)$ term, or they do not operate early enough in the tree construction algorithm to restrict the scope permutation tree. However, as we have seen, there is a large class of cases in which filters significantly reduce the search space through tree pruning, and so they offer a mechanism for rendering scope disambiguation computationally efficient.

6. Conclusion

We have formulated constraints on scope readings as filters on the $k!$-tuples that $perms\_scope_k$ produces. These filters are PTCT terms which encode Boolean conditions and quantification over the integers of indexed k-tuples. In principle, they permit PTCT to achieve expressive completeness in the sense of Ebert (2005).

We have also invoked a tree generation algorithm to characterise (the permutation part of) the computable function that $perms\_scope_k$ denotes. When filters are applied as constraints on nodes in the tree that the algorithm generates, they can significantly reduce the search space of possible scope readings given by an underspecified representation.

Underspecified representations, the projection of a particular scope interpretation, and constraints on possible scope readings are all specified by appropriately typed λ-terms within the semantic representation language, PTCT, rather than through operations on schematic metalinguistic objects. Our proposed treatment of underspecified representations within PTCT achieves both significant expressive power and efficient computation of possible scope interpretations.

Acknowledgements

We are grateful to Christian Ebert for helpful comments on an earlier draft of this paper. His recent Ph.D. dissertation has stimulated and influenced much of the work we report here. We would also like to thank Dick Crouch and Ron Kaplan for useful discussion of the complexity issues we address. The second author's research was supported by research grant number RES–000–23–0065 from the Economic and Social Research Council of the United Kingdom.

References

Blackburn, P. and Bos, J.: 2005, *Representation and Inference for Natural Language*, CSLI, Stanford

Bos, J.: 1995, Predicate logic unplugged, in *Proceedings of the Tenth Amsterdam Colloquium*, Amsterdam, Holland

Cooper, R.: 1983, *Quantification and Syntactic Theory*, Synthese Language Library, D. Reidel, Dordrecht

Copestake, A., Flickinger, D., and Sag, I. A.: 1997, *Minimal Recursion Semantics*, Tech. rep., Stanford University, Stanford, CA, unpublished ms.

Curry, H. B. and Feys, R.: 1958, *Combinatory Logic*, vol. 1 of *Studies in Logic*, North Holland

Ebert, C.: 2005, *Formal Investigation of Underspecified Representations*, Ph.D. thesis, Department of Computer Science, King's College London, unpublished

van Eijck, J.: 2003, *Computational Semantics and Type Theory*, unpublished ms., CWI, Amsterdam

Fox, C. and Lappin, S.: 2005a, *Formal Foundations of Intensional Semantics*, Blackwell, Oxford

Fox, C. and Lappin, S.: 2005b, Underspecified interpretations in a Curry-typed representation language, *The Journal of Logic and Computation* 15, 131–143

Fox, C. and Lappin, S.: 2005c, *Expressive Completeness and Computational Efficiency for Underspecified Representations*, unpublished ms., University of Essex and King's College London

Keenan, E.: 1992, Beyond the Fregean boundary, *Linguistics and Philosophy* 15, 199–221

Koller, A., Niehren, J., and Thater, S.: 2003, Bridging the gap between underspecified formalisms: Hole semantics as dominance constraints, in *Proceedings of 11th EACL*, Budapest

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

Reyle, U.: 1993, Dealing with ambiguities by underspecification: Construction, representation and deduction, *Journal of Semantics* 10, 123–179