GENERALIZED QUANTIFIERS WITH UNDERSPECIFIED SCOPE RELATIONS IN A FIRST-ORDER REPRESENTATION LANGUAGE

In this paper we show that by adding Curry typing to a first-order property theory it is possible to represent the full range of generalized quantifiers (GQs) corresponding to natural language determiners. We characterize GQs as property terms that specify cardinality relations between properties (or separation types). We also generate underspecified quantifier scope representations within the representation language, rather than through meta-language devices, as in most current treatments of underspecification (Reyle, 1993; Bos, 1995; Blackburn & Bos, 2003; Copestake, Flickinger, & Sag, 1997).

Our representation language, Property Theory with Curry Typing (PTCT), encodes a property theory within a language of terms (an untyped lambda calculus). We add dynamic Curry typing and use a first-order logic to specify the truth-conditions of the propositional subpart of the term language. The resulting system gives us a language of semantic representation which has the expressive power of higher- order type theories, like Monatague's IL, while remaining first-order in its formal resources.

We extend PTCT in a straightforward way to express underspecified quantifier scope representations as terms within the representation language. The expressive power of the language permits the formulation of restrictions 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.

We follow Keenan (1992) and van Eijck (2003) in taking GQs to be arity reduction operators that apply 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. (In Keenan's presentation, "non-Fregean" generalised quantifiers can bind more that one of R's arguments, and so reduce its arity by more than 1.)

GQs are of type $(X \Longrightarrow \mathsf{Prop}) \Longrightarrow \mathsf{Prop}$. Core propositional relations, such as verbs, are of type $X_1 \Longrightarrow \ldots \Longrightarrow X_n \Longrightarrow \mathsf{Prop}$. Slightly modifying van Eijck (2003)'s Haskell-based treatment of GQs, we define an operator R to "lift" quantifiers to the appropriate level to combine with a relation. We then compose representations of n quantifiers with a relation r using $RQ_1(RQ_2 \ldots (RQ_n r) \ldots)$.

For our treatment of underspecification, we use a product permutation function *perms_scope* that takes a k-ary product of scope taking elements (in the order in which they appear in the surface syntax) and a k-ary relation representing the core proposition as arguments. *perms_scope* returns 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 proposition, 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 by our *perms_scope* function. In an implementation of PTCT it is not necessary to compute the full

Date: February 2004.

k!-ary product of permutations that provides (one of the) arguments for this function. We can adopt lazy evaluation of our scope reading function to identify the ith permutation, as required. Therefore, the PTCT term consisting of the application of *perms_scope* 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.

Consider the example *Every man loves a woman*, with the GQs interpreting the subject and object NPs, the core relation, and PTCT term expressing the underspecified representation of the sentence given, respectively, as follows.

(1) $Q_1 = \lambda P \hat{\forall} x \in B(\mathsf{man}'(x) \hat{\rightarrow} P(x))$
(2) $Q_2 = \lambda Q \hat{\exists} y \in B(\mathsf{woman}'(y) \hat{\wedge} Q(y))$
(3) $\lambda uv.\mathsf{loves}'uv$
(4) $perms\_scope(\langle\langle Q_1, Q_2\rangle, \lambda uv.\mathsf{loves}'uv\rangle)$

The permutations of the quantifiers and the core representation that we produce are

(5) $\langle\langle\langle Q_1, Q_2\rangle, \lambda uv.\mathsf{loves}'uv\rangle\langle\langle Q_2, Q_1\rangle, \lambda vu.\mathsf{loves}'uv\rangle\rangle$

Applying relation reduction to computing the final propositions gives us a product containing the two readings.

(6) $perms\_scope(\langle\langle Q_1, Q_2\rangle, \lambda uv.\mathsf{loves}'uv\rangle)) =$
$\langle \forall x(\mathsf{man}'(x) \rightarrow \exists y(\mathsf{woman}'(y) \wedge \mathsf{loves}'(x, y))),$
$\exists y(\mathsf{woman}'(y) \wedge \forall x(\mathsf{man}'(x) \wedge \mathsf{loves}'(x, y)))\rangle$

To obtain resolved scope readings from an underspecified representation, we define a function $project\_scope^i_k$ that computes the ith permutation of a k-ary product of propositions.

There are various kinds of constraints that limit the set of possible scope readings for a particular sentence to a proper subset of the set of k! orderings of the k scope taking elements which appear in it. A common condition on relative scope is the strong preference for wide scope assignment to certain quantifiers by virtue of their lexical semantic properties, like *a certain N'*. A second kind of condition depends upon the syntactic domain in which a GQ appears. So, for example, a quantified NP within a relative clause cannot take scope over a quantified NP in which the relative clause is embedded. Scope constraints of these kinds can be formulated as filters on the k!-tuple of permutations $\langle\langle Qtuple_1, Rel_1\rangle, \ldots, \langle Qtuple_{k!}, Rel_{k!}\rangle\rangle$ that *perms_scope* generates for an argument pair $\langle Qtuple_1, Rel_1\rangle$. Each such filter is Boolean property function that imposes a condition on the elements of the k!-tuple.

In conclusion, underspecified scope representations, the projection of a particular scope reading, 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, as in current theories of underspecified semantic representation.

References

Blackburn, P., & Bos, J. (2003). *Representation and inference for natural language.* Stanford: CSLI.

Bos, J. (1995). Predicate logic unplugged. In *Proceedings of the tenth amsterdam colloquium.* Amsterdam, Holland.

Copestake, A., Flickinger, D., & Sag, I. A. (1997). *Minimal recursion semantics* (Tech. Rep.). Stanford, CA: Stanford University. (unpublished ms.)

Keenan, E. (1992). Beyond the fregean boundary. *Linguistics and Philosophy*, *15*, 199–221.

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

van Eijck, J. (2003). *Computational semantics and type theory.* CWI, Amsterdam: unpublished ms.