Intensionality and Quantifier Scope

Yoad Winter

March 23, 2012

Topics: a toy extensional lexicon, object-quantifier composition, quantifier scope, intensional contexts, *de dicto*/*de re* ambiguities, possible world semantics, the Montague/Quine hypothesis, intensionalization, a toy intensional lexicon

Main claims:

1. Using the techniques that we have learned so far we can develop substantial semantic accounts of many phenomena.
2. Some cases of ambiguity, called *scope ambiguity*, pose serious problems that must be treated using some non-trivial extensions of the syntax, the semantics, and/or the syntax-semantics interface.
3. After doing that, we can use *possible-world semantics* to address the old problem of *de dicto*/*de re* ambiguities.

1 A toy extensional lexicon

Word	**Type**	**Meaning**	**Definition**
Donald	$(et)t$	$I_{\mathbf{d}'}$	$I_{\mathbf{d}'}(B) = 1 \Leftrightarrow \mathbf{d}' \in B$
every	$(et)((et)t)$	$\mathbf{every}'$	$\mathbf{every}'(A)(B) = 1 \Leftrightarrow A \subseteq B$
a	$(et)((et)t)$	$\mathbf{some}'$	$\mathbf{some}'(A)(B) = 1 \Leftrightarrow A \cap B \neq \emptyset$
duck	et	$\mathbf{duck}'$	–
cat	et	$\mathbf{cat}'$	–
swam	et	$\mathbf{swim}'$	–
flew	et	$\mathbf{fly}'$	–
found	$e(et)$	$\mathbf{find}'$	–
and	$(et)((et)(et))$	$\mathbf{and}'$	$\mathbf{and}'(A)(B)(x) = 1 \Leftrightarrow x \in A \cap B$

Simple sentences covered: *Donald swam; Some duck swam; Every duck swam; Every duck swam and flew*.

Easily extendable for covering entailments with:

1. Copulas: *Donald is a duck.*
2. Other coordinations: *Every duck (n)either swam (n)or flew*.
3. Restrictive modifiers: *Every fat duck swam*; *Every duck swam quickly*.
4. Relative clauses: *A cat that flew swam.*
5. Exceptive constructions: *Every duck but Donald swam.*

   ...and many other phenomena.

The problem of quantifiers in object position

(1) Donald [found every cat].

How can the $((et)t)$-type denotation of the object compose with the (et)-type denotation of the transitive verb?

Proposed answers:

1. Syntactic/Semantic – Montague (1973), Partee & Rooth (1983), May (1977), Heim & Kratzer (1997), Carpenter (1997), de Groote (2001), Muskens (2003).
2. Purely semantic – Cooper (1975), Partee and Rooth (1983!), Van Benthem (1991), Hendriks (1993).

For our purposes, a simple version of Hendriks' semantic answer is sufficient.

Notation: For a $R \in D_{e(et)}$ and $x \in D_e$, R^x is the characteristic function in D_{et} of $\{y \in D_e : R(y)(x) = 1\}$ – the *left-image* of x under R.

Object narrow scope operator:

$$\text{ONS}(R_{e(et)})(Q_{(et)t})(x_e) = 1 \quad \text{iff} \quad R^x \in Q.$$

In words: ONS is the operator of type $((e(et))(((et)t)(et))$ that sends any binary relation R between entities to the binary relation between quantifiers Q and entities x, s.t. the left-image of x under R is in Q.

Object-narrow-scope TV-modifier:

(2) Donald [[ϵ_{ONS} found] every cat].

a. $\text{ONS}(\mathbf{find'})(\mathbf{every'}(\mathbf{cat'}))$

$= \{x \in E : \{y \in E : \mathbf{find'}(y)(x)\} \in \mathbf{every'}(\mathbf{cat'})\}$

b. $\text{ONS}(\mathbf{find'})(\mathbf{every'}(\mathbf{cat'})) \in I_{\mathbf{d'}}$

$\Leftrightarrow \{y \in E : \mathbf{find'}(y)(\mathbf{d'})\} \in \mathbf{every'}(\mathbf{cat'})$

$\Leftrightarrow \forall y[\mathbf{cat'}(y) \to \mathbf{find'}(y)(\mathbf{d'})]$

2 Quantifier scope ambiguity

But this solution is not enough for transitive constructions. Consider the following example:

(3) A duck [found every cat].

Two readings:

1. *Object narrow scope* (ONS): There was a duck that found all the cats.
2. *Object wide scope* (OWS): For each cat there was a duck that found it.

(4) A duck [[ϵ_{ONS} found] every cat].

$\textsc{ons}(\mathbf{find}')(\mathbf{every}'(\mathbf{cat}')) \in \mathbf{some}'(\mathbf{duck}')$

$\Leftrightarrow\ \exists x[\mathbf{duck}'(x) \wedge \forall y[\mathbf{cat}'(y) \rightarrow \mathbf{find}'(y)(x)]]$

This is the ONS reading. We derive the OWS reading using an **object wide scope operator**:

$$\textsc{ows}(R_{((et)t)(et)})(Q_1)(Q_2) = 1 \quad \text{iff} \quad \{y \in E : R(I_y) \in Q_2\} \in Q_1$$

In words: $\textsc{ows}$ is the operator of type $((((et)t)(et))(((et)t)(et))$ that sends any binary relation R between quantifiers and entities to the binary relation between quantifiers Q_1 and Q_2, s.t. the set S of y's whose individual's (I_y's) right-image under R is in Q_2 satisfies $S \in Q_1$.

Object-wide-scope TV-modifier:

(5) A duck [[ϵ_{OWS}[ϵ_{ONS} found]] every cat].

$\textsc{ows}(\textsc{ons}(\mathbf{find}'))(\mathbf{every}'(\mathbf{cat}'))(\mathbf{some}'(\mathbf{duck}'))$

$\Leftrightarrow\ \forall y[\mathbf{cat}'(y) \rightarrow \exists x[\mathbf{duck}'(x) \wedge \mathbf{find}'(y)(x)]]$

Officially, we add two phonologically empty TV modifiers to the lexicon.

Word	Type	Meaning
ϵ_{ONS}	$((e(et))(((et)t)(et))$	ONS
ϵ_{OWS}	$((((et)t)(et))(((et)t)(((et)t)t))$	OWS

Remark: Linguistically, this is only one of many proposed solutions to the problems of TV-Quantifier composition and quantifier scope ambiguity. We use it here for illustrative purposes only.

3 *De dicto/de re* ambiguity

Consider the following ambiguous sentences.

(6) Donald *believes* a cat swam.

a. Donald has a belief regarding the existence of some or other swimming cat. – *de dicto* reading

b. There is a cat x s.t. Donald has a belief regarding x's swimming abilities. – *de re* reading

(7) Donald *looked for* a cat.

a. Donald would be satisfied if he finds any cat. – *de dicto* reading

b. There is a cat x s.t. Donald would be satisfied if he find x. – *de re* reading

Under the *de dicto* reading, these sentences do not require that cats exist. In these examples we say that the italicized expression creates an *intensional context* for the underlined expression.

Questions and proposed answers:

1. How do we get *de dicto* readings?
   Possible world semantics: sentences denote *sets of possible worlds* in a model, rather than mere truth or falsity in a model.

2. How do we get *de dicto/de re* ambiguities?
   The Quine-Montague hypothesis: the same general mechanism that derives quantifier scope ambiguity also derives *de dicto/de re* ambiguities.

Basic idea: We add a domain D_s of type s for *possible worlds*. Sentences will now denote *propositions* – functions of type st, which characterize sets of of possible worlds.

The truth-conditionality criterion (intensional version): *Let S_1 and S_2 be sentences of type st. Then S_1 entails S_2 if and only if for every intended intensional model M:* $[\![S_1]\!]^M \subseteq [\![S_2]\!]^M$.

Example: The embedded clause *a cat swam* in (6) denotes a proposition. The verb *believe* thus basically denotes a binary relation between such propositions and entities (e.g. the denotation of *Donald*).

***Believe* version 1**:

$[\![\text{Donald believes a cat swam}]\!] = 1$
iff Donald stands in the *believe* relation to the set of worlds in which a cat swam
But we may need further embedding:

(8) Every duck believes Donald believes a duck swam.

***Believe* version 2**:

⟦Donald believes a cat swam⟧
= the set of worlds w s.t. <u>in w</u>, Donald stands in the *believe* relation to the set of worlds in which a cat swam

Conclusion: The type of *believe* is $(st)(e(st))$.

Abbreviation: p (propositions) – instead of (st). Hence the type of *believe* is $p(ep)$

Question: But how do we guarantee that all sentences (e.g. *a duck swam*) denote propositions?

Answer (Van Benthem 1988): A global type change – replace all t's in the lexicon by p's. Notably – one-place *predicates* (type et) will become one-place *properties* (type ep): functions from entities to propositions.

Semantics of this type change – Ben-Avi & Winter (2007), Kanazawa (2009).

The resulting intensional lexicon

Word	Type	Meaning	Definition
Donald	$(ep)p$	$I^i_{\mathbf{d}'}$	$I^i_{\mathbf{d}'}(\mathcal{B}_{ep})(w_s) = 1 \Leftrightarrow \mathbf{d}' \in \mathcal{B}^w$
every	$(ep)((ep)p)$	$\mathbf{every}^i$	$\mathbf{every}^i(\mathcal{A})(\mathcal{B})(w) = 1 \Leftrightarrow \mathcal{A}^w \subseteq \mathcal{B}^w$
a	$(ep)((ep)p)$	$\mathbf{some}^i$	$\mathbf{some}^i(\mathcal{A})(\mathcal{B})(w) = 1 \Leftrightarrow \mathcal{A}^w \cap \mathcal{B}^w \neq \emptyset$
duck	ep	$\mathbf{duck}^i$	–
cat	ep	$\mathbf{cat}^i$	–
swam	ep	$\mathbf{swim}^i$	–
flew	ep	$\mathbf{fly}^i$	–
found	$e(ep)$	$\mathbf{find}^i$	–
and	$(ep)((ep)(ep))$	$\mathbf{and}^i$	$\mathbf{and}^i(\mathcal{A})(\mathcal{B})(x)(w) = 1 \Leftrightarrow x \in \mathcal{A}^w \cap \mathcal{B}^w$
ϵ_{ONS}	$((e(ep))(((ep)p)(ep))$	ONS^i	see Ben-Avi/Winter 2007
ϵ_{OWS}	$((((ep)p)(ep))(((ep)p)(ep))$	OWS^i	see Ben-Avi/Winter 2007

$\mathcal{A}^w$ is the left-image of w_s under $\mathcal{A}_{e(st)}$ – the et predicate that is the *extension* of $\mathcal{A}$ in w.

Omission: For the definition of ONS^i and OWS^i see Ben-Avi & Winter/Kanazawa's general *intensionalization* procedure.

Claim (Ben-Avi & Winter, and more elegantly and generally – Kanazawa): In a grammar generated by this general intensionalization procedure, the intentional truth-conditionality criterion is equivalent to the extensional truth-conditionality criterion.

But now we can also add items like the following:

believes	$p(ep)$	$\mathbf{believe}'$	–
looked for	$((ep)p)(ep)$	$\mathbf{look\_for}'$	–

(9) Donald [[looked for] a cat].

$I^i_{\mathbf{d}'}((\mathbf{look\_for}')(\mathbf{some}^i(\mathbf{cat}^i)))$

$= \{w \in D_s : \mathbf{look\_for}'(\mathbf{some}^i(\mathbf{cat}^i))(\mathbf{d}')(w)\}$ – *de dicto* reading
(note mistaken omission of '(w)' in handout previous version)

(10) Donald [ϵ_{OWS}[looked for] a cat].

$\text{OWS}^i(\mathbf{look\_for}')(\mathbf{some}^i(\mathbf{cat}^i))(I^i_{\mathbf{d}'})$

$= \{w \in D_s : \exists x[\mathbf{cat}^i(x)(w) \wedge \mathbf{look\_for}'(I^i_x)(\mathbf{d}')(w)]\}$ – *de re* reading
(note mistaken omission of second '(w)' in handout previous version)

This gives an easy analysis of cases of coordination like:

(11) Donald looked for and found a cat.

a. Donald [[looked for and [ϵ_{ONS} found]] a cat]. – *de dicto*
b. Donald [[[ϵ_{OWS}[looked for]] and [ϵ_{OWS}[ϵ_{ONS} found]]] a cat]. – *de re*

References

Ben-Avi, G. & Winter, Y. (2007), The semantics of intensionalization, *in* 'Proceedings of the Workshop on New Directions in Type-theoretic Grammars'.

Carpenter, B. (1997), *Type-Logical Semantics*, MIT Press, Cambridge, Massachusetts.

Cooper, R. (1975), Montague's Semantic Theory and Transformational Syntax, PhD thesis, University of Massachusetts at Amherst.

de Groote, P. (2001), Towards abstract categorial grammars, *in* 'Proceedings of the 39th annual meeting of the Association for Computational Linguistics (ACL)'.

Heim, I. & Kratzer, A. (1997), *Semantics in Generative Grammar*, Blackwell.

Hendriks, H. (1993), Studied Flexibility: categories and types in syntax and semantics, PhD thesis, University of Amsterdam.

Kanazawa, M. (2009), A note on intensionalization. Unpublished ms. NII, Tokyo. December 27, 2009. `http://research.nii.ac.jp/~kanazawa/publications/intensionalization.pdf`.

May, R. (1977), The Grammar of Quantification, PhD thesis, Massachusetts Institute of Technology.

Montague, R. (1973), The proper treatment of quantification in ordinary English, *in* J. Hintikka, J. Moravcsik & P. Suppes, eds, 'Approaches to Natural Languages: proceedings of the 1970 Stanford workshop on grammar and semantics', D. Reidel, Dordrecht. Reprinted in R. Thomason, editor (1974), *Formal Philosophy: selected papers of Richard Montague*, Yale, New Haven.

Muskens, R. (2003), Language, Lambdas, and Logic, *in* G.-J. Kruijff & R. Oehrle, eds, 'Resource Sensitivity in Binding and Anaphora', Studies in Linguistics and Philosophy, Kluwer, pp. 23–54.

Partee, B. & Rooth, M. (1983), Generalized conjunction and type ambiguity, *in* R. Bauerle, C. Schwarze & A. von Stechow, eds, 'Meaning, Use and Interpretation of Language', De Gruyter, Berlin.

van Benthem, J. (1988), Strategies of intensionalization, *in* I. Bodnár, A. Máté & L. Pólos, eds, 'Intensional Logic, History of Philosophy, and Methodology: To Imre Ruzsa on the occasion of his 65th birthday', Department of Symbolic Logic, Eötvös University, Budapest, pp. 41–59.

van Benthem, J. (1991), *Language in Action: categories, lambdas and dynamic logic*, North-Holland, Amsterdam.