A Model-Theoretic Reconstruction of Type-Theoretic Semantics for Anaphora

Matthew Gotham

University of Oslo

22nd Conference on Formal Grammar, 22 July 2017

What this talk is about

- A framework for the semantics of anaphora and accessibility constraints.
- Inspired by analyses in type-theoretical approaches to semantics using dependent types, reconstructed in (more or less) simple type theory.
- We'll look at a couple of examples of cross-sentential binding and a 'donkey sentence', and see how the the system blocks inaccessible antecedents.
- There are more examples (negation, proportional quantifiers, weak and strong readings) in the paper.

‘Model-Theoretic’?

- What I mean is that meanings will be given as expressions of a logical language, which are taken to be dispensable in favour of *their* interpretations in a model (as in Montague 1973), which is where the ‘real’ semantics is.
- Expressions of the language of type theory are not understood this way in TTS—see Luo 2014 and Ranta 1994: §2.27.
- However, I don’t want to lean to heavily on this point from now on.

Pronouns bound outside of scope

(1) A donkey brays. Giles feeds it.

$\exists x(\mathbf{donkey}(x) \wedge \mathbf{bray}(x)) \wedge \mathbf{feed}(\mathbf{giles}, ?)$

(2) Every farmer who owns a donkey feeds it.

$\forall y.(\mathbf{farmer}(y) \wedge \exists x.\mathbf{donkey}(x) \wedge \mathbf{own}(y, x)) \rightarrow \mathbf{feed}(y, ?)$

Various options pursued:

- $? := x$, change the model theory to extend the scope of $\exists x$
- ? is a description, possibly indexed to situations
- ? is a constant manipulated by functions
- …etc.

In Type-Theoretic Semantics

(1) A donkey brays. Giles feeds it.

$$\big(\Sigma y : (\Sigma x : \textsc{donkey})(\textsc{bray}(x))\big)\big(\textsc{feed}(\mathbf{giles}, \pi_1 y)\big)$$

(Sundholm 1986, Ranta 1994)

$$\lambda c.\big(\Sigma w : (\Sigma u : (\Sigma x : e)(\textsc{donkey}(x)))(\textsc{bray}(\pi_1 u))\big)\big(\textsc{feed}(\mathbf{giles}, \pi_1 \pi_1 w)\big)$$

(Bekki 2014)

Witnesses

dependent pairs

(1) A donkey brays. Giles feeds it.

$$\left(\Sigma y : (\Sigma x : \textsc{donkey})(\textsc{bray}(x))\right)\left(\textsc{feed}(\mathbf{giles}, \pi_1 y)\right)$$

The type of ordered pairs $\langle\langle a, b\rangle, c\rangle$ such that:

- *a* is a donkey, and
- *b* is a proof that *a* brays, and
- *c* is a proof that Giles feeds *a*.

In Type-Theoretic Semantics

(2) Every farmer who owns a donkey feeds it.

$$\big(\Pi z : (\Sigma x : \textsc{farmer})((\Sigma y : \textsc{donkey})(\textsc{own}(x, y)))\big)\big(\textsc{feed}(\pi_1 z, \pi_1\pi_2 z)\big)$$

(Sundholm 1986, Ranta 1994)

$$\lambda c.\big(\Pi u :(\Sigma x : e)\,(\textsc{farmer}(x) \times (\Sigma v : (\Sigma y : e)(\textsc{donkey}(y)))(\textsc{own}(x, \pi_1 v)))\big)\big(\textsc{feed}(\pi_1 u, \pi_1\pi_1\pi_2\pi_2 u)\big)$$

(Bekki 2014)

Witnesses

dependent functions

(2) Every farmer who owns a donkey feeds it.

$$\big(\Pi z : (\Sigma x : \textsc{farmer})((\Sigma y : \textsc{donkey})(\textsc{own}(x, y)))\big)\big(\textsc{feed}(\pi_1 z, \pi_1 \pi_2 z)\big)$$

The type of functions f such that:

- the domain of f is the set of ordered pairs $\langle a, \langle b, c \rangle\rangle$ such that:
  - a is a farmer, and
  - b is a donkey, and
  - c is a proof that a owns b, and
- f maps every $\langle a, \langle b, c \rangle\rangle$ in its domain to a proof that a feeds b.

The idea behind this paper

(is very simple)

- Formalize those glosses in higher-order logic (Jacobs & Melham (1993) have shown how).
- Work backwards to the lexical entries we need to derive them compositionally.

N.B.:

- Limited polymorphism required.
- Event(ualitie)s play the role of proof objects.
- Discourse-level existential closure plays the role of the non-empty type condition.

Types

Base types

- 1 — unit
- e — entities
- v — eventualities
- t — booleans

Binary type constructors

- $\rightarrow$ — functional types
- $\times$ — product types

($\rightarrow$ and $\times$ associate to the right, and $\times$ binds more tightly than $\rightarrow$)

Terms

- $* : 1$ — unit
- $f^{\alpha\to\beta}(a^{\alpha}) : \beta$ — application
- $\lambda v^{\alpha}(\phi^{\beta}) : \alpha\to\beta$ — abstraction
- $(a^{\alpha}, b^{\beta}) : \alpha\times\beta$ — pairing
- $[c^{\alpha\times\beta}]_0 : \alpha$ — left projection
- $[c^{\alpha\times\beta}]_1 : \beta$ — right projection

Example lexical entries

$$
\begin{aligned}
\textit{and}\ ,\ ; &\mapsto \lambda p^{\alpha\to\beta\to t}.\lambda q^{\alpha\times\beta\to\gamma\to t}.\lambda i^{\alpha}.\lambda a^{\beta\times\gamma}.p(i)([a]_0)\wedge q(i,[a]_0)([a]_1)\\
. &\mapsto \lambda p^{1\to\alpha\to t}.\exists z^{\alpha}.p(*)(z)\\
a &\mapsto \lambda P^{e\times\alpha\to\beta\to t}.\lambda V.\lambda i^{\beta}.\lambda a^{(e\times\alpha)\times\gamma}.P([a]_0)(i)\wedge V([[a]_0]_0)(i,[a]_0)([a]_1)\\
&\quad \text{where } V : e\to\beta\times e\times\alpha\to\gamma\to t\\
\textit{donkey} &\mapsto \lambda a^{e\times 1}.\lambda i^{\alpha}.\mathbf{donkey}([a]_0)\\
\textit{brays} &\mapsto \lambda x^{e}.\lambda i^{\alpha}.\lambda e^{v}.\mathbf{bray}(x,e)\\
\textit{Giles} &\mapsto \lambda P^{e\to\alpha\times e\to\beta\to t}.\lambda i^{\alpha}.\lambda a^{e\times\beta}.P([a]_0)(i,[a_0])([a]_1)\wedge[a]_0=\mathbf{giles}\\
\textit{owns} &\mapsto \lambda D^{(e\to\alpha\to v\to t)\to\beta\to\gamma\to t}.\lambda x^{e}.D\big(\lambda y^{e}.\lambda a^{\alpha}.\lambda e^{v}.\mathbf{own}(x,y,e)\big)\\
\textit{it} &\mapsto \lambda V^{\alpha\to\beta\to\gamma\to t}.\lambda i^{\beta}.V(g^{\beta\to\alpha}(i))(i)\\
&\quad \text{where } g \text{ stands for an arbitrarily-chosen free variable}
\end{aligned}
$$

Instantiated lexical entries

$$a \mapsto \lambda P^{e\times 1\to 1\to t}.\lambda V^{e\to 1\times e\times 1\to v\to t}.\lambda i^{1}.\lambda a^{(e\times 1)\times v}.P([a]_0)(i) \wedge V([[a]_0]_0)(i, [a]_0)([a]_1)$$

$$donkey \mapsto \lambda a^{e\times 1}.\lambda i^{1}.\mathbf{donkey}([a]_0)$$

$$brays \mapsto \lambda x^{e}.\lambda i^{1\times e\times 1}.\lambda e^{v}.\mathbf{bray}(x, e)$$

left context for the whole sentence
NP witness, part of the left context for the VP
VP witness

$$a\ donkey\ brays \mapsto \lambda i^{1}.\lambda a^{(e\times 1)\times v}.\mathbf{donkey}([[a]_0]_0) \wedge \mathbf{bray}([[a]_0]_0, [a]_1)$$

Instantiated lexical entries

Giles $\mapsto$

$\lambda P^{e\to(1\times(e\times1)\times v)\times e\to v\to t}.\lambda i^{1\times(e\times1)\times v}.\lambda a^{e\times v}.P([a]_0)(i,[a]_0)([a]_1)\wedge[a]_0=\mathbf{giles}$

owns $\mapsto\lambda D.\lambda x^{e}.D\big(\lambda y^{e}.\lambda a^{(1\times(e\times1)\times v)\times e}.\lambda e^{v}.\mathbf{own}(x,y,e)\big)$

where $D:(e\to(1\times(e\times1)\times v)\times e\to v\to t)\to(1\times(e\times1)\times v)\times e\to v\to t$

it $\mapsto\lambda V^{e\to(1\times(e\times1)\times v)\times e\to v\to t}.\lambda i^{(1\times(e\times1)\times v)\times e}.V(g^{(1\times(e\times1)\times v)\times e\to e}(i))(i)$

left context
NP witness
VP witness

Giles owns it $\mapsto$

$\lambda i^{1\times(e\times1)\times v}.\lambda a^{e\times v}.\mathbf{own}([a]_0,g^{(1\times(e\times1)\times v)\times e\to e}(i,[a]_0),[a]_1)\wedge[a]_0=\mathbf{giles}$

Instantiated lexical entries

$$; \mapsto \lambda p^{1\to(e\times 1)\times v\to t}.\lambda q^{1\times(e\times 1)\times v\to e\times v\to t}.\lambda i^{1}.\lambda a^{((e\times 1)\times v)\times e\times v}.p(i)([a]_0) \wedge q(i,[a]_0)([a]_1)$$

$$. \mapsto \lambda p^{1\to((e\times 1)\times v)\times e\times v\to t}.\exists z^{((e\times 1)\times v)\times e\times v}.p(*)(z)$$

left context
first sentence witness, part of the left context for the second sentence
second sentence witness

A donkey brays; Giles owns it. $\mapsto$

$$\exists z^{((e\times 1)\times v)\times e\times v}.\big(\mathbf{donkey}([[[z]_0]_0]_0) \wedge \mathbf{bray}([[[z]_0]_0]_0,[[z]_0]_1)\big) \wedge \big(\mathbf{own}\big([[z]_1]_0, g^{(1\times(e\times 1)\times v)\times e\to e}((*,[z]_0),[[z]_1]_0),[[z]_1]_1\big) \wedge [[z]_1]_0 = \mathbf{giles}\big)$$

Resolution of the free variable

$$g^{(1\times(e\times 1)\times v)\times e\rightarrow e}$$

Natural resolution: a function that selects an element of (an element of…) a tuple (of tuples…)

For any types α, β and γ:

- $\lambda b^{\alpha}.b$ is a natural resolution function (NRF).
- $\lambda b^{\alpha\times\beta}.[b]_0$ is an NRF.
- $\lambda b^{\alpha\times\beta}.[b]_1$ is an NRF.
- For any terms $F : \beta\rightarrow\gamma$ and $G : \alpha\rightarrow\beta$, $\lambda b^{\alpha}.F(G(b))$ is an NRF if F and G are NRFs.

In this case, the resolution that we want gives us
$g := \lambda b^{(1\times(e\times 1)\times v)\times e}.[[[[b]_0]_1]_0]_0$

With the pronoun resolution

$g := \lambda b^{(1\times(e\times 1)\times v)\times e}.[[[[b]_0]_1]_0]_0$

$$\begin{aligned}\Rightarrow_\beta\ & \exists z^{((e\times 1)\times v)\times e\times v}.\big(\mathbf{donkey}([[[z]_0]_0]_0)\wedge\mathbf{bray}([[[z]_0]_0]_0,[[z]_0]_1)\big)\\ &\qquad \wedge\big(\mathbf{own}\big([[z]_1]_0,([[[[((*,[z]_0),[[z]_1]_0)]_0]_1]_0]_0),[[z]_1]_1\big)\\ &\qquad\quad \wedge [[z]_1]_0=\mathbf{giles}\big)\end{aligned}$$

$$\begin{aligned}\Rightarrow_\beta\ & \exists z^{((e\times 1)\times v)\times e\times v}.\big(\mathbf{donkey}([[[z]_0]_0]_0)\wedge\mathbf{bray}([[[z]_0]_0]_0,[[z]_0]_1)\big)\\ &\qquad \wedge\big(\mathbf{own}\big([[z]_1]_0,[[[z]_0]_0]_0,[[z]_1]_1\big)\wedge [[z]_1]_0=\mathbf{giles}\big)\end{aligned}$$

$$\equiv\quad \exists x^e.\exists e^v.\exists y^e.\exists d^v.\big(\mathbf{donkey}(x)\wedge\mathbf{bray}(x,e)\big)\wedge\big(\mathbf{own}(y,x,d)\wedge y=\mathbf{giles}\big)$$

More lexical entries

every $\mapsto$

$$\lambda P^{e\times\alpha\to\beta\to t}.\lambda V^{e\to\beta\times e\times\alpha\to\gamma\to t}.\lambda i^{\beta}.\lambda f^{e\times\alpha\to\gamma}.\forall a^{e\times\alpha}.P(a)(i)\to V([a]_0)(i,a)(f(a))$$

who $\mapsto$

$$\lambda V^{e\to\beta\times e\times\alpha\to\gamma\to t}.\lambda P^{e\times\alpha\to\beta\to t}.\lambda a^{e\times\alpha\times\gamma}.\lambda i^{\beta}.P([a]_0,[[a_1]_0])(i) \\ \wedge V([a]_0)(i,[a]_0,[[a]_1]_0)([[a]_1]_1)$$

A donkey sentence

Every farmer who owns a donkey feeds it. $\mapsto$

$$\exists f^{e\times 1\times(e\times 1)\times v\rightarrow v}.\forall a^{e\times 1\times(e\times 1)\times v}.\big(\textbf{farmer}([a]_0)\wedge\textbf{donkey}([[[[a]_1]_1]_0]_0)\\ \wedge\,\textbf{own}([a]_0,[[[[a]_1]_1]_0]_0,[[[a]_1]_1]_1)\big)\\ \rightarrow\textbf{feed}([a]_0,g^{1\times e\times 1\times(e\times 1)\times v\rightarrow e}(*,a),f(a))$$

(empty) left context

NP witness

The resolution we want: $g := \lambda b^{1\times e\times 1\times(e\times 1)\times v}.[[[[[b]_1]_1]_1]_0]_0$

Resolved

$g := \lambda b^{1\times e\times 1\times(e\times 1)\times v}.[[[[[b]_1]_1]_1]_0]_0$

$$\begin{aligned}\Rightarrow_\beta\ \ \exists f^{e\times 1\times(e\times 1)\times v\to v}.\forall a^{e\times 1\times(e\times 1)\times v}.\big(&\mathbf{farmer}([a]_0)\wedge\mathbf{donkey}([[[[a]_1]_1]_0]_0)\\ &\wedge\mathbf{own}([a]_0,[[[[a]_1]_1]_0]_0,[[[a]_1]_1]_1)\big)\\ &\to\mathbf{feed}([a]_0,[[[[a]_1]_1]_0]_0,f(a))\end{aligned}$$

$$\equiv\ \ \forall x^e.\forall y^e.\forall e^v.\big(\mathbf{farmer}(x)\wedge\mathbf{donkey}(y)\wedge\mathbf{own}(x,y,e)\big)\to\exists d^v.\mathbf{feed}(x,y,d)$$

Accessibility

Every donkey brays; Giles owns it. $\mapsto$

$$\exists a^{(e\times 1\to v)\times e\times v}.\forall x^{e\times 1}(\mathbf{donkey}([x]_0) \to \mathbf{bray}([x]_0, [a]_0(x)))$$
$$\wedge\ (\mathbf{own}([[a]_1]_0, g^{1\times(e\times 1\to v)\times e\to e}(*, [a]_0), [[a]_1]_1)$$
$$\wedge\ [[a]_1]_0 = \mathbf{giles})$$

first sentence witness, part of the left context for the second sentence
second sentence witness

Given the type of g, there is no way for the pronoun to be bound to donkeys.

Plurals

$$\textit{two} \mapsto \lambda P.\lambda V.\lambda i^{\beta}.\lambda X.\mathbf{two}\big(\lambda x^{e}.\exists a^{\alpha}.\exists d^{\gamma}.X((x,a),d)\big)$$

$$\wedge\, \forall b^{e\times\alpha}.\forall c^{\gamma}.X(b,c) \rightarrow \big(P(b)(i) \wedge V([b]_0)(i,b)(c)\big)$$

where $P : e{\times}\alpha{\rightarrow}\beta{\rightarrow}t$, $V : e{\rightarrow}\beta{\times}e{\times}\alpha{\rightarrow}\gamma{\rightarrow}t$ and $X, Y : (e{\times}\alpha){\times}\gamma{\rightarrow}t$

$$\textit{them} \mapsto \lambda V^{\alpha\rightarrow\beta\rightarrow\gamma\rightarrow t}.\lambda i^{\beta}.\lambda c^{\gamma}.\forall a^{\alpha}.G^{\beta\rightarrow\alpha\rightarrow t}(i) \rightarrow V(a)(i)(c)$$

where G stands for an arbitrarily-chosen free variable

Instantiated

Two donkeys bray; Giles owns them. $\mapsto$

$$\exists a^{((e\times 1)\times v\to t)\times e\times v}.\mathbf{two}\big(\lambda x^e.\exists y^1.\exists d^v.[a]_0((x,y),d)\big)$$
$$\wedge\ \forall b^{e\times 1}\big(\forall c^v.[a]_0(b,c)\to\big(\mathbf{donkey}([b]_0)\wedge\mathbf{bray}([b]_0,c)\big)\big)$$
$$\wedge\ \forall y^e.G^{(1\times((e\times 1)\times v\to t))\times e\to e\to t}((*,[a]_0),[[a]_1]_0)(y)$$
$$\to\big(\mathbf{own}([[a]_1]_0,y,[[a]_1]_1)\wedge[[a]_1]_0=\mathbf{giles}\big)$$

first sentence witness, part of the left context for the second sentence
second sentence witness

Additional resolution conventions for plural (set) entities

For any types α, β and γ:

- $\lambda b^{\alpha \times \beta \to t}.\lambda Y^{\alpha}.\exists Z^{\beta}.b(Y, Z)$ is an NRF.
- $\lambda b^{\alpha \times \beta \to t}.\lambda Y^{\alpha}.\exists Z^{\beta}.b(Z, Y)$ is an NRF.

So $G^{(1 \times ((e \times 1) \times v \to t)) \times e \to e \to t}$ can be resolved to
$\lambda b^{(1 \times ((e \times 1) \times v \to t)) \times e}.\lambda x^{e}.\exists n^{1}.\exists e^{v}.[[b]_0]_1((x, n), e)$

Resolved

$G := \lambda b^{(1\times((e\times 1)\times v\to t))\times e}.\lambda x^e.\exists n^1.\exists e^v.[[b]_0]_1((x,n),e)$

$$
\begin{aligned}
&\exists a^{((e\times 1)\times v\to t)\times e\times v}.\mathbf{two}\big(\lambda x^e.\exists y^1.\exists d^v.[a]_0((x,y),d)\big)\\
&\qquad\wedge \forall b^{e\times 1}\big(\forall c^v.[a]_0(b,c)\to\big(\mathbf{donkey}([b]_0)\wedge\mathbf{bray}([b]_0,c)\big)\big)\\
&\qquad\wedge \forall y^e.\exists n^1\big(\exists e^v.[a]_0((y,n),e)\big)\\
&\qquad\qquad\to\big(\mathbf{own}([[a]_1]_0,y,[[a]_1]_1)\wedge[[a]_1]_0=\mathbf{giles}\big)
\end{aligned}
$$

$$
\begin{aligned}
\equiv\quad &\exists R^{e\times v\to t}.\exists z^e.\exists e^v.\mathbf{two}\big(\lambda x^e.\exists d^v.R(x,d)\big)\\
&\qquad\wedge \forall v^e\big(\forall c^v.R(v,c)\to(\mathbf{donkey}(v)\wedge\mathbf{bray}(v,c))\big)\\
&\qquad\wedge \forall y^e.\exists b^v(R(y,b))\to(\mathbf{own}(z,y,e)\wedge z=\mathbf{giles})
\end{aligned}
$$

Conclusion

- Reconstruction of type-theoretic treatment of anaphora in (more or less) simple type theory.
- Similarity to list-, or stack-based approaches to dynamic semantics (Dekker 1994, van Eijck 2001, de Groote 2006, Nouwen 2007).
- A semantic account of pronoun accessibility.
- Not much yet to say about:
  - Anti-locality effects (‘Principle B’).
  - Crossover.
  - Quantificational/modal subordination.
  - Many other issues.

References I

Bekki, Daisuke. 2014. Representing anaphora with dependent types. In Nicholas Asher & Sergei Soloviev (eds.), *Logical aspects of computational linguistics* (Lecture Notes in Computer Science 8535), 14–29. Berlin, Heidelberg: Springer.

de Groote, Philippe. 2006. Towards a Montagovian account of dynamics. In Masayuki Gibson & Jonathan Howell (eds.), *Proceedings of Semantics and Linguistic Theory*, vol. 16, 1–16.

Dekker, Paul. 1994. Predicate logic with anaphora. In Mandy Harvey & Lynn Samelmann (eds.), *Proceedings of Semantics and Linguistic Theory*, vol. 4, 79–95.

van Eijck, Jan. 2001. Incremental dynamics. *Journal of Logic, Language and Information* 10. 319–351.

References II

Jacobs, Bart & Tom Melham. 1993. Translating dependent type theory into higher order logic. In *TLCA 1993: International conference on typed lambda calculi and applications* (Lecture Notes in Computer Science 664). Berlin, Heidelberg: Springer.

Luo, Zhaohui. 2014. Formal semantics in modern type theories: is it model-theoretic, proof-theoretic, or both?. In Nicholas Asher & Sergei Soloviev (eds.), *Logical aspects of computational linguistics: 8th international conference* (Lecture Notes in Computer Science 8535), 177–188. Berlin, Heidelberg: Springer.

Montague, Richard. 1973. The proper treatment of quantification in ordinary English. In Patrick Suppes, Julius Moravcsik & Jaakko Hintikka (eds.), *Approaches to natural language*, 221–242. Dordrecht: D. Reidel.

Nouwen, Rick. 2007. On dependent pronouns and dynamic semantics. *Journal of Philosophical Logic* 36(2). 123–154.

References III

Ranta, Aarne. 1994. *Type-theoretical grammar*. (Indices 1). Oxford: Oxford University Press.

Sundholm, Göran. 1986. Proof theory and meaning. In Dov Gabbay & Franz Guenther (eds.), *Handbook of philosophical logic*, vol. 3, 471–506. Dordrecht: D. Reidel.