Handouts for lectures on:

Introduction to Semantics

Prof. Timothy Williamson and Dr. Ofra Magidor

Trinity Term 2008

Introduction to Semantics
Week 1

I. The general framework
We will explore semantic theories which are:
- compositional
- truth-conditional
- model-theoretic
- type-theoretic (and type driven?)
- the input for semantic interpretation is provided by syntax (via binary phrase-structure

II. Some preliminaries: functions and sets
functions
- An *n*-place *relation* is a set of *n*-tuples. A ***function*** is a two-place relation R over a set D, such that for every a∈D, if <a,b> and <a,c>∈R, then b=c.
- If *f* is a function, and <a,b>∈*f* then the expression '*f*(a)' denotes b.
- For each function we may specify the ***domain*** of the function. As we defined functions above the domain is simply defined as {x: there is a y such that <x,y>∈*f*}. But sometimes we will prefer to specify a domain D independently, and define a function *f* to be *total* if and only if for every x∈D there is a y, such that <x,y>∈*f* and *partial* if this is not the case.
- We can also specify a set to be the ***range*** of a function. The only constraint on the range is that {y: there is an x such that <x,y>∈*f*} will be a subset of the range.
- We use the notation *f:* A→B to note that *f* is function where A is its domain, and B is its range.
- Given two sets A and B, the notations B^A and [A→B] are both used to denote the set of all functions whose domain is A and whose range is B.

characteristic sets and functions
- Given a function *f* with the range {T,F} (truth-values) we define its ***characteristic set*** to be {d: *f*(d)=T}.
- Given a domain D and a set S⊆D we define its ***characteristic function*** to be the function *f* with the domain D, such that for each d∈D, *f*(d)=T if d∈S, and *f*(d)=F, if d∉S.

schönfinkelization
- For any *n*-place function we can find a one-place function that is basically equivalent to it. Suppose we have a two-place function *f*: AxB→C. We can define a one-place function *f**: A→[B→C] as follows: For each a∈A, *f**(a) = g_a. So that for each b∈B, g_a(b)=*f*(a,b). This means that for every a∈A and b∈B, *f*(a,b)=(*f**(a))(b).

III. The logic
typed propositional calculus
types:
(i) t is a type.
(ii) if *a* and *b* are types then <*a*,*b*> is type.

expressions:

Name for expression	Notation	Type of expression
Propositional letters	P, Q, R	t
Negation sign	~	<t,t>
Disjunction	$\vee$	<t,<t,t>>
Conjunction	$\wedge$	<t,<t,t>>

If φ is an expression of type <*a*,*b*> and ψ is an expression of type *a* then $\varphi(\psi)$ is a well-formed expression of type *b*.

semantics:
- For each type *a* we assign a domain D_a.
- The domain of type t is $D_t = \{T,F\}$. The domain of types <*a*,*b*> is $[D_a \rightarrow D_b]$. (I.e. functions from the domain of type *a* to the domain of type *b*).
- An interpretation is a function *I* such that for each expression α of type *a*, $I(\alpha) \in D_a$.
- The interpretations of ~, $\vee$, and $\wedge$ are fixed to the usual truth-functions (after *schönfinkelization*).
- The interpretation of any formula of the form $\varphi(\psi)$ is $I(\varphi)(I(\psi))$.
- This means that once we fix the interpretations of the propositional letters the truth-value of any formula is determined.

typed predicate calculus
types:
(i) t is a type.
(ii) e is a type
(iii) if *a* and *b* are types then <*a*,*b*> is type.

Expressions:
- All the expressions of propositional calculus
- For each type *a* we have an infinite set of constants of type *a*.
- For each type *a* we have an infinite set of variables of type *a*.
- The quantifier symbols $\forall$, $\exists$ (untyped)
- The identity symbol = (untyped).
- If φ is an expression of type <*a*,*b*> and ψ is an expression of type *a* then $\varphi(\psi)$ is a well-formed expression of type *b*.
- For every expression φ of type t, and any variable (of any type) x, $\forall x\varphi$ and $\exists x\varphi$ are well-formed expression of type t.
- For every two expressions φ and ψ of type *a*, $\varphi=\psi$ is a well-formed expression of type a.

semantics:

- For each type a we assign a domain D_a.
- The domain of type t is $D_t = \{T,F\}$. The domain of type e is the set of individuals D_e. For any types a and b the domain of type $<a,b>$ is $[D_a \rightarrow D_b]$. (I.e. functions from the domain of type a to the domain of type b).
- An assignment function is a function g such that for each variable x of type a, $g(x) \in D_a$.
- The interpretation function I is a function such that for each constant c of type a $I(c) \in D_a$.
- The interpretations of $\sim$, $\vee$, and $\wedge$ are fixed to the usual truth-functions (after *schönfinkelization*).

For each expression α we define its semantic-value relative to an interpretation function I and an assignment function g (denoted by $[\alpha]^{I, g}$) as follows:

- If α is a constant then $[\alpha]^{I, g} = I(\alpha)$
- If α is a variable then $[\alpha]^{I, g} = g(\alpha)$
- If α is of the form $\varphi(\psi)$ then $[\alpha]^{I, g} = [\varphi]^{I, g}([\psi]^{I, g})$
- If α is of the form $[\varphi=\psi]$ then $[\alpha]^{I, g} = T$ if $[\varphi]^{I, g} = [\psi]^{I, g}$, and $[\alpha]^{I, g} = F$ otherwise.
- If α is of the form $\forall x\varphi$ then $[\alpha]^{I, g} = T$ if for every assignment g* differing from g at most in its assignment to x, $[\varphi]^{I, g^*} = T$, otherwise $[\alpha]^{I, g} = F$.
- If α is of the form $\exists x\varphi$ then $[\alpha]^{I, g} = T$ if for some assignment g* differing from g at most in its assignment to x, $[\varphi]^{I, g^*} = T$, otherwise $[\alpha]^{I, g} = F$.
- Note that as usual the semantic-values (i.e. truth-values) of sentences with no free variable do not depend on the assignment function.

IV. Natural language semantics

Lexical items (some tentative suggestions):

- Proper names (‘Joe’, ‘London’): type e.
- Common nouns (‘cat’); adjectives (‘red’); non-transitive verbs (‘is eating’): type <e,t>.
- Second order properties (‘is a colour’, ‘is a desirable property to have’): type <<e,t>,t>
- Transitive verbs (‘loves’): type <e,<e,t>>
- Adverbs (‘quickly’, ‘very’): type <<e,t>,<e,t>>
- Prepositions (‘in’, ‘on’, ‘with’): <e,<<e,t>,<e,t>>
- Sentential connectives (‘and’): <t,t>
- Quantifiers (‘every’): <<e,t>,<<e,t>,t>>

Stage 1: translations to typed-predicate logic

- Each lexical item is assigned an expression of the relevant type.
- Complex expressions are built out of simpler by following the phrase-structures.

Stage 2: semantic values

- The interpretation function I (which interprets the constants) is determined by the lexicon. (E.g. ‘eats’ is assigned the function f: $f(x)=T$ if x eats, and $f(x)=F$ otherwise).

- All other expressions receive their semantic-values compositionally as described above.

Q: but isn't this too extensional to count as a serious semantic theory?

- **A1:** complicate the semantics by adding possible worlds
- **A2:** complicate the semantic in some other way (?)
- **A3:** adopt a more intentional understanding of model specifications.
- **A4:** specify meanings via trees and not only "roots".
- **A5:** This is what linguists do ☺.

IV. Lambda abstraction

Extending typed-predicate calculus

Extending the language:

- If φ is an expression of type *b* and x is a variable of type *a,* then λx.φ is an expression of type <*a*, *b*>

Extending the semantics:

- If α is of the form λx.φ, where x is a variable of type *a* and φ an expression of type *b* then $[\alpha]^{I, g}$ is the function *f*: $D_a \rightarrow D_b$ which is defined as follows. Let every $a \in D_a$, g^a is an assignment just like *g* except that $g^a(x)=a$. Then $f(a) = [\varphi]^{I, g_a}$

λ-conversion (also called 'Beta conversion'):

- An expression of the form λx.φ(ψ) can be converted or reduced to φ with every occurrence of x replaced with ψ (provided some restrictions are met.- see e.g. Gamut, p.110).

Some applications of the extension:

- Given that *loves* is an expression of type <e,<e,t>>, we can give a derived semantic value for *loves John* of type <e,t>: λx.(*loves*(x)(John)) (or in a slightly nicer notation: λx.x loves John)
- We can define *loves himself* (λx.x *loves* x).
- We can interpret sentences such as 'Drinking and driving is unwise'.

Further Readings

- Carpenter, R., *Type-Logical Semantics*, MIT Press (1997).
- Gamut, L.T.F., *Logic, Language, and Meanings* (vol.2), Chicago University Press (1991) (for current lecture: chapter 4)
- Heim, I. and Kratzer, A., *Semantics in Generative Grammar*, Oxford (1998). (for current lecture: chapters 1-3)
- Lewis, D., 'General Semantics', *Synthese 22* (1970), pp. 18-67.
- Montague, R., 'The proper treatment of quantification in ordinary English', in his *Formal Philosophy,* Yale University Press (1974). [A seminal paper, though not for the faint hearted ☺].
- Potts, C., Logic for Linguists *http://udrive.oit.umass.edu/potts/web/lsa07/lsa108P/* [probably the most accessible of all these sources, and also contains problem sets with solutions]

Introduction to Semantics

Week 5: Quantifiers

Standard (unary) quantifiers by lambda abstraction

Syntax

If $\boldsymbol{\varphi}$ is an expression of type b and $\mathbf{v}$ is a variable of type a, then $\boldsymbol{\lambda}\mathbf{v}.\boldsymbol{\varphi}$ is an expression of type $<a, b>$.

For any type a, if $\boldsymbol{\alpha}$ is an expression of type $<a, \text{t}>$, $\forall\ \boldsymbol{\alpha}$ and $\exists\ \boldsymbol{\alpha}$ are expressions of type t.

Thus instead of $(\forall\ \mathbf{x})\ \mathbf{Fx}$ and $(\exists\ \mathbf{x})\ \mathbf{Fx}$ we have $\forall\ \boldsymbol{\lambda}\mathbf{x}.\mathbf{Fx}$ and $\exists\ \boldsymbol{\lambda}\mathbf{x}.\mathbf{Fx}$.

Also, if $\mathbf{P}$ is a constant or variable of type $<a, \text{t}>$ then $\forall\ \mathbf{P}$ and $\exists\ \mathbf{P}$ are expressions of type t (NB they do not contain a variable of type a).

Semantics

Notation: If g is an assignment, $\mathbf{v}$ is a variable of type a and $\text{a} \in D_a$, then $g[\mathbf{v}/\text{a}]$ is the assignment g^* such that $g^*(\mathbf{v}) = \text{a}$ and for any variable $\mathbf{u}$ other than $\mathbf{v}$, $g^*(\mathbf{u}) = g(\mathbf{u})$.
If X is a set, $|X|$ is the number of members of X.

Let I be any interpretation and g any assignment.

If $\boldsymbol{\varphi}$ is an expression of type b and $\mathbf{v}$ is a variable of type a, then $[\boldsymbol{\lambda}\mathbf{v}.\boldsymbol{\varphi}]^{I,g}$ is the function f: $D_a \rightarrow D_b$ such that for $\text{a} \in D_a$, $\text{f(a)} = [\boldsymbol{\varphi}]^{I,g[\mathbf{v}/\text{a}]}$.

If $\boldsymbol{\alpha}$ is an expression of type $<a, \text{t}>$, $[\forall\ \boldsymbol{\alpha}]^{I,g} = \text{T}$ iff $|\{\text{a} \in D_a: [\boldsymbol{\alpha}]^{I,g}(\text{a}) \neq \text{T}\}| = 0$.

If $\boldsymbol{\alpha}$ is an expression of type $<a, \text{t}>$, $[\exists\ \boldsymbol{\alpha}]^{I,g} = \text{T}$ iff $|\{\text{a} \in D_a: [\boldsymbol{\alpha}]^{I,g}(\text{a}) = \text{T}\}| > 0$.

(We are assuming that all expressions have semantic values, so if $\boldsymbol{\varphi}$ is of type t then $[\boldsymbol{\varphi}] =$ F unless $[\boldsymbol{\varphi}] = \text{T}$.)

Extensions

Where $\exists^{\leq \mathbf{n}}$ is the quantifier "at most n" (with the same syntax as $\forall$ and $\exists$):
if $\boldsymbol{\alpha}$ is an expression of type $<a, \text{t}>$, $[\exists^{\leq \mathbf{n}}\ \boldsymbol{\alpha}]^{I,g} = \text{T}$ iff $|\{\text{a} \in D_a: [\boldsymbol{\alpha}]^{I,g}(\text{a}) = \text{T}\}| \leq \text{n}$.
Similarly for "at least n" and "exactly n" etc.

If $\boldsymbol{\beta}$ is an expression of type e, $\mathbf{Q}_{\boldsymbol{\beta}}$ is an expression of type $<<\text{e}, \text{t}>, \text{t}>$ and for any expression $\boldsymbol{\alpha}$ of type $<\text{e}, \text{t}>$, $[\mathbf{Q}_{\boldsymbol{\beta}}(\boldsymbol{\alpha})]^{I,g} = [\boldsymbol{\alpha}]^{I,g}([\boldsymbol{\beta}]^{I,g})$.

Limitations of the unary analysis for natural languages

Standard symbolizations:

All As are Bs $\forall\ \boldsymbol{\lambda x.(Ax \rightarrow Bx)}$

Some As are Bs $\exists\ \boldsymbol{\lambda x.(Ax\ \&\ Bx)}$

At most n As are B $\exists^{\leq n}\ \boldsymbol{\lambda x.(Ax\ \&\ Bx)}$ etc.

These secure the right truth-values and entailments, but look rather artifical — is there really a conditional hidden in "All As are Bs" and a conjunction in "Some As are Bs"?

For other quantifiers, the problem is worse. Consider "most". For simplicity, assume that "Most As are Bs" is true iff more As are Bs than are non-Bs.
Can we represent "Most As are Bs" as $\mathbf{M\ \lambda x.(Ax * Bx)}$, where for expressions $\boldsymbol{\varphi}$, $\boldsymbol{\psi}$ of type t, $[\boldsymbol{\varphi * \psi}]^{I,g} = [\boldsymbol{\varphi}]^{I,g} * [\boldsymbol{\psi}]^{I,g}$ and for expressions $\boldsymbol{\alpha}$ of type <e, t>, $[\mathbf{M}\ \boldsymbol{\alpha}]^{I,g} = \mathrm{M}([\boldsymbol{\alpha}]^{I,g})$ for suitable functions * and M (* will take pairs of truth-values to truth-values and M will take functions from individuals to truth-values to truth-values)?
NB For simplicity, binary connectives such as ***** have not been schönfinkelized.

We can represent "Most things are A and B" and "Most things are A only if B" [material conditional] in that way, but neither is equivalent to "Most As are Bs":
If 90% of individuals are neither A nor B, 9% are A and B and 1% are A but not B, "Most As are Bs" is true while "Most things are A and B" is false.
If 90% of individuals are neither A nor B, 1% are A and B and 9% are A but not B, "Most As are Bs" is false while "Most things are A only if B" is true.

A proof that there can be no such functions as M and *:
Suppose for reductio that there are. Since T*T, T*F and F*T are all in {T, F}, at least one of the following must hold: (i) T*T = T*F; (ii) T*T = F*T; (iii) T*F = F*T. We show that each case is impossible. Consider an interpretation *I* on which all individuals are in the extension of **A**, and 90% are in the extension of **B**.

(i) For any assignment g, $[\mathbf{Ax{*}Bx}]^{I,g} = [\mathbf{Ax}]^{I,g}{*}[\mathbf{Bx}]^{I,g} = \mathrm{T}{*}[\mathbf{Bx}]^{I,g} = \mathrm{T}{*}[\neg\mathbf{Bx}]^{I,g} = [\mathbf{Ax}]^{I,g}{*}[\neg\mathbf{Bx}]^{I,g} = [\mathbf{Ax{*}\neg Bx}]^{I,g}$. Hence $[\boldsymbol{\lambda}\mathbf{x.(Ax{*}Bx)}]^{I,g} = [\boldsymbol{\lambda}\mathbf{x.(Ax{*}\neg Bx)}]^{I,g}$.
Thus $[\mathbf{M}\ \boldsymbol{\lambda}\mathbf{x.(Ax{*}Bx)}]^{I,g} = [\mathbf{M}\ \boldsymbol{\lambda}\mathbf{x.(Ax{*}\neg Bx)}]^{I,g}$. But most things that satisfy **Ax** satisfy **Bx** while it is not the case that most things that satisfy **Ax** satisfy **¬Bx**.

(ii) For any assignment g, either $[\mathbf{Bx}]^{I,g} = \mathrm{F}$, in which case $[\mathbf{Ax{*}Bx}]^{I,g} = \mathrm{T{*}F} = [\neg\mathbf{Bx{*}Bx}]^{I,g}$, or $[\mathbf{Bx}]^{I,g} = \mathrm{T}$, in which case $[\mathbf{Ax{*}Bx}]^{I,g} = \mathrm{T{*}T} = \mathrm{F{*}T} = [\neg\mathbf{Bx{*}Bx}]^{I,g}$.
Reasoning as in (i), $[\mathbf{M}\ \boldsymbol{\lambda}\mathbf{x.(Ax{*}Bx)}]^{I,g} = [\mathbf{M}\ \boldsymbol{\lambda}\mathbf{x.(\neg Bx{*}Bx)}]^{I,g}$. But most things that satisfy **Ax** satisfy **Bx** while it is not the case that most things that satisfy **¬Bx** satisfy **Bx**.

(iii) For any assignment g, either $[\mathbf{Bx}]^{I,g} = \mathrm{F}$, in which case $[\mathbf{Ax{*}\neg Bx}]^{I,g} = \mathrm{T{*}T} = [\neg\mathbf{Bx{*}Ax}]^{I,g}$, or $[\mathbf{Bx}]^{I,g} = \mathrm{T}$, in which case $[\mathbf{Ax{*}\neg Bx}]^{I,g} = \mathrm{T{*}F} = \mathrm{F{*}T} = [\neg\mathbf{Bx{*}Ax}]^{I,g}$.
Reasoning as in (i), $[\mathbf{M}\ \boldsymbol{\lambda}\mathbf{x.(Ax{*}\neg Bx)}]^{I,g} = [\mathbf{M}\ \boldsymbol{\lambda}\mathbf{x.(\neg Bx{*}Ax)}]^{I,g}$. But most things that satisfy **¬Bx** satisfy **Ax** while it is not the case that most things that satisfy **Ax** satisfy **¬Bx**.

A solution

To solve the problem, let us treat **MOST** as an untyped expression such that for any type a, if **α** is an expression of type $<a, t>$ then **(MOST α)** is an expression of type $<<a, t>, t>$, and for any expression **β** of type $<a, t>$:

$[\![\mathbf{(MOST\ \alpha)(\beta)}]\!]^{I,g}$ = T iff

$$|\{a \in D_a: [\![\boldsymbol{\alpha}]\!]^{I,g}(a) = [\![\boldsymbol{\beta}]\!]^{I,g}(a) = T\}| > |\{a \in D_a: [\![\boldsymbol{\alpha}]\!]^{I,g}(a) = T \neq [\![\boldsymbol{\beta}]\!]^{I,g}\}|.$$

We can apply a parallel treatment to **ALL**, **SOME**, **NO** etc., giving them the same syntax as **MOST** and the following semantic clauses:

$[\![\mathbf{(ALL\ \alpha)(\beta)}]\!]^{I,g} = T$ iff $|\{a \in D_a: [\![\boldsymbol{\alpha}]\!]^{I,g}(a) = T \neq [\![\boldsymbol{\beta}]\!]^{I,g}\}| = 0$.

$[\![\mathbf{(SOME\ \alpha)(\beta)}]\!]^{I,g} = T$ iff $|\{a \in D_a: [\![\boldsymbol{\alpha}]\!]^{I,g}(a) = [\![\boldsymbol{\beta}]\!]^{I,g}(a) = T\}| > 0$.

$[\![\mathbf{(NO\ \alpha)(\beta)}]\!]^{I,g} = T$ iff $|\{a \in D_a: [\![\boldsymbol{\alpha}]\!]^{I,g}(a) = [\![\boldsymbol{\beta}]\!]^{I,g}(a) = T\}| = 0$.

Expressions such as “all”, “some”, “no”, “at least seven”, “exactly seven”, “at most seven” and “most” are *determiners*. In natural languages, there is typically an asymmetry between the grammatical categories of the expressions occupying the **α** and **β** positions in **(DET α)(β)**: **α** occupies noun position while **β** occupies verb phrase position. Our analysis ignored this asymmetry, whose significance is unclear.

Note that the constituent structure of **(DET α)(β)** seems to match that of the corresponding English sentences much better than did that of the original unary analyses. For example, in “Some dogs bark”, the first two words seem to constitute a unit as in **(SOME λx.(dog(x)))(λx.(bark(x)))** but not ∃ **x (dog(x) & bark(x))**.

If we fix the type a, then we can assign types and semantic values to determiners. For example, if we put a = e, then **SOME** is of type $<<e, t>, <<e, t>, t>>$, and $[\![\mathbf{SOME}]\!]^{I,g}$ is the function F: $D_{<e, t>} \rightarrow D_{<<e, t>, t>}$ such that if g and h are both functions: $D_e \rightarrow D_t$, then $F(g)(h) = T$ iff $|\{i \in D_e: g(i) = T = h(i)\}| > 0$. This gives a ‘language independent’ characterization of the semantic value. Likewise for the other determiners. A more intuitive way of conceiving the semantic values of determiners is as relations between sets, e.g. in the case of **SOME** the relation of having a non-empty intersection.

Conservativeness

Determiners like **ALL**, **SOME, NO** and **MOST** all satisfy the following constraint, where if $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are both of category a so is $\boldsymbol{\alpha}$ **&** $\boldsymbol{\beta}$ and $[\boldsymbol{\alpha} \mathbin{\&} \boldsymbol{\beta}]^{I,g} = \mathrm{T}$ iff $[\boldsymbol{\alpha}]^{I,g} = [\boldsymbol{\beta}]^{I,g} = \mathrm{T}$:

Conservativeness $\quad [(\mathbf{DET}\ \boldsymbol{\alpha})(\boldsymbol{\beta})]^{I,g} = [(\mathbf{DET}\ \boldsymbol{\alpha})(\boldsymbol{\alpha} \mathbin{\&} \boldsymbol{\beta})]^{I,g}$

For example, "All squares are red" is equivalent to "All squares are red squares", "Some squares are red" to "Some squares are red squares", "No squares are red" to "No squares are red squares" and "Most squares are red" to "Most squares are red squares".

Any determiner **DET** will be conservative in this sense provided that $[(\mathbf{DET}\ \boldsymbol{\alpha})(\boldsymbol{\beta})]^{I,g}$ is a function of $\{a \in D_a\text{: } [\boldsymbol{\alpha}]^{I,g}(a) = [\boldsymbol{\beta}]^{I,g}(a) = \mathrm{T}\}$ and $\{a \in D_a\text{: } [\boldsymbol{\alpha}]^{I,g}(a) = \mathrm{T} \neq [\boldsymbol{\beta}]^{I,g}\}$. For we always have $\{a \in D_a\text{: } [\boldsymbol{\alpha}]^{I,g}(a) = \mathrm{T} = [\boldsymbol{\beta}]^{I,g}\} = \{a \in D_a\text{: } [\boldsymbol{\alpha}]^{I,g}(a) = \mathrm{T} = [\boldsymbol{\alpha} \mathbin{\&} \boldsymbol{\beta}]^{I,g}\}$ and $\{a \in D_a\text{: } [\boldsymbol{\alpha}]^{I,g}(a) = \mathrm{T} \neq [\boldsymbol{\beta}]^{I,g}\} = \{a \in D_a\text{: } [\boldsymbol{\alpha}]^{I,g}(a) = \mathrm{T} \neq [\boldsymbol{\alpha} \mathbin{\&} \boldsymbol{\beta}]^{I,g}\}$.

It has been conjectured as a linguistic universal that all determiners in natural language are conservative. If we treat "only" as a determiner, it will be a counterexample, since "Only squares are red" is false but "Only squares are red squares" is true. However, the variety of positions in which "only" can occur suggest that it is an adverb rather than a determiner (e.g. try substituting "all", "some" or "most" for "only" in "I only skimmed the newspaper").

Further reading

Jon Barwise and Robin Cooper, 'Generalized quantifiers and natural language', *Linguistics and Philosophy* 4 (1981): 159-219.
Irene Heim and Angelika Kratzer, *Semantics in Generative Grammar* (Oxford, Blackwell, 1998), ch. 6.
Stanley Peters and Dag Westerståhl, *Quantifiers in Language and Logic* (Oxford, Clarendon Press, 2006).

Introduction to Semantics

Week 6: Definite Descriptions

Definite descriptions are expressions such as ‘the man who broke the bank at Monte Carlo’, ‘my book’, … and their equivalents in other languages. In present terms, they result from combining an expression such as ‘the’ or ‘my’ with an expression of type <e, t> (‘man who broke the bank at Monte Carlo’, ‘book’) and in turn combine with an expression of type <e, t> (‘is lost’) to give a sentence [NB for simplicity, we continue to ignore the difference between nouns and predicates.] But this leaves it open whether they are of type <<e, t>, t> (quantifiers) or of type e (singular terms), depending on whether e.g. ‘my book’ supplies the function or its argument in ‘My book is lost’.

<u>Definite descriptions as quantifiers</u>

For simplicity, we impose typing on determiners (unlike last week).
Treat $\mathbf{DET} \in \{\mathbf{ALL, SOME, NO, MOST,} \ldots\}$ as of type <<e, t>, <<e, t>, t>>.

Similarly, we can treat **THE** as a determiner. Thus if $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are expressions of type <e, t>, **THE** $\boldsymbol{\alpha}$ is of type <<e, t>, t> (a quantifier) and $\mathbf{(THE\ \boldsymbol{\alpha})(\boldsymbol{\beta})}$ is of type t.

Semantics:

For any interpretation I and assignment g,
$[\mathbf{SOME}]^{I,g}$ is the function f: $D_{<e,t>} \rightarrow D_{<<e,t>,\, t>}$ such that for any functions g, h: $D_e \rightarrow D_t$,
$f(g)(h) = T$ iff $|\{a \in D_e\text{: } g(a) = h(a) = T\}| > 0$;
$f(g)(h) = F$ iff $|\{a \in D_e\text{: } g(a) = h(a) = T\}| = 0$.

Now since $[\mathbf{(SOME\ \boldsymbol{\alpha})(\boldsymbol{\beta})}]^{I,g} = [\mathbf{SOME}]^{I,g}([\boldsymbol{\alpha}]^{I,g})([\boldsymbol{\beta}]^{I,g}$,
$[\mathbf{(SOME\ \boldsymbol{\alpha})(\boldsymbol{\beta})}]^{I,g} = T$ iff $|\{a \in D_a\text{: } [\boldsymbol{\alpha}]^{I,g}(a) = [\boldsymbol{\beta}]^{I,g}(a) = T\}| > 0$;
$[\mathbf{(SOME\ \boldsymbol{\alpha})(\boldsymbol{\beta})}]^{I,g} = F$ iff $|\{a \in D_a\text{: } [\boldsymbol{\alpha}]^{I,g}(a) = [\boldsymbol{\beta}]^{I,g}(a) = T\}| = 0$.

Similarly:
$[\mathbf{THE}]^{I,g}$ is the function f: $D_{<e,t>} \rightarrow D_{<<e,t>t>}$ such that for any functions g, h : $D_e \rightarrow D_t$,
$f(g)(h) = T$ iff $|\{a \in D_e\text{: } g(a) = h(a) = T\}| = 1$ and $|\{a \in D_e\text{: } g(a) = T, h(a) = F\}| = 0$;
$f(g)(h) = F$ iff $|\{a \in D_e\text{: } g(a) = h(a) = T\}| \neq 1$ or $|\{a \in D_e\text{: } g(a) = T, h(a) = F\}| \neq 0$.

Now since $[\mathbf{(THE\ \boldsymbol{\alpha})(\boldsymbol{\beta})}]^{I,g} = [\mathbf{THE}]^{I,g}([\boldsymbol{\alpha}]^{I,g})([\boldsymbol{\beta}]^{I,g})$,
$[\mathbf{(THE\ \boldsymbol{\alpha})(\boldsymbol{\beta})}]^{I,g} = T$ iff $|\{a \in D_e\text{: } [\boldsymbol{\alpha}]^{I,g}(a) = [\boldsymbol{\beta}]^{I,g}(a) = T\}| = 1$ and
$|\{a \in D_a\text{: } [\boldsymbol{\alpha}]^{I,g}(a) = T, [\boldsymbol{\beta}]^{I,g}(a) = F\}| = 0$;
$[\mathbf{(THE\ \boldsymbol{\alpha})(\boldsymbol{\beta})}]^{I,g} = F$ iff $|\{a \in D_e\text{: } [\boldsymbol{\alpha}]^{I,g}(a) = [\boldsymbol{\beta}]^{I,g}(a) = T\}| \neq 1$ or
$|\{a \in D_a\text{: } [\boldsymbol{\alpha}]^{I,g}(a) = T, [\boldsymbol{\beta}]^{I,g}(a) = F\}| \neq 0$.

NB These are exactly the same truth and falsity conditions as Russell's theory assigns, but without the analysis of **THE** in terms of $\exists$ and $\forall$. Unlike Russell's theory, this treatment preserves **THE α** as a semantically significant constituent.

Normally, definite descriptions will be interpreted with respect to a contextually restricted domain of individuals D_e, so the uniqueness requirement is not as demanding as it sounds.

How should the quantifier account be extended to plural definite descriptions, as in

(1) The peasants are revolting.

which can be true (only?) if there is more than one peasant.
Could we get the right result by treating 'peasants' as of a different (plural?) type from 'peasant'?

Definite descriptions as singular terms

In mathematics, functional expressions such as '+' and '√' seem to form singular terms; thus '5 + 7' and '√144' seem to occupy the same grammatical position as '12'. But expressions such as '5 + 7' and '√144' correspond to definite descriptions such as 'the sum of five and seven' and 'the (positive) square root of a hundred and forty four'. This suggests treating definite descriptions as complex singular terms. Thus we treat 'the' as of type <<e, t>, e>. One might be attracted by an analogous treatment of complex demonstratives such as 'this man'.

Semantics

For any interpretation *I* and assignment *g*,
$[\mathbf{THE}]^{I,g}$ is the *partial* function f: $D_{<e,t>} \rightarrow D_e$ such that for any function g: $D_e \rightarrow D_t$,
if $|\{a \in D_e: g(a) = T\}| = 1$ then $f(g) \in D_e$ and $g(f(g)) = T$;
if $|\{a \in D_e: g(a) = T\}| \neq 1$ then f(g) is undefined.

Now since $[\boldsymbol{\beta}(\mathbf{THE}\ \boldsymbol{\alpha})]^{I,g} = [\boldsymbol{\beta}]^{I,g}([\mathbf{THE}]^{I,g}([\boldsymbol{\alpha}]^{I,g}))$,
$[\boldsymbol{\beta}(\mathbf{THE}\ \boldsymbol{\alpha})]^{I,g} = T$ iff $|\{a \in D_e: [\boldsymbol{\alpha}]^{I,g}(a) = [\boldsymbol{\beta}]^{I,g}(a) = T\}| = 1$ and
$|\{a \in D_a: [\boldsymbol{\alpha}]^{I,g}(a) = T, [\boldsymbol{\beta}]^{I,g}(a) = F\}| = 0$;
$[\boldsymbol{\beta}(\mathbf{THE}\ \boldsymbol{\alpha})]^{I,g} = F$ iff $|\{a \in D_e: [\boldsymbol{\alpha}]^{I,g}(a) = [\boldsymbol{\beta}]^{I,g}(a) = T\}| = 0$ and
$|\{a \in D_a: [\boldsymbol{\alpha}]^{I,g}(a) = T, [\boldsymbol{\beta}]^{I,g}(a) = F\}| = 1$;
$[\boldsymbol{\beta}(\mathbf{THE}\ \boldsymbol{\alpha})]^{I,g}$ is undefined iff $|\{a \in D_e: [\boldsymbol{\alpha}]^{I,g}(a) = T\}| \neq 1$.

[Similar issues to those for the previous treatment arise about plural definite descriptions, as in (1) above.]

NB This is similar to Strawson's treatment of definite descriptions. The truth value gaps are not imposed to respect native speaker intuitions that e.g. the question of whether 'The king of France is bald' is true or false 'does not arise'. Rather, it is the natural outcome of treating definite descriptions as of type e and computing semantic values by functional application: without an argument, a function cannot deliver a value. However, this treatment does not imply that sentences with definite descriptions can in effect express only singular propositions, for it allows binding into them, as in

(2) Every woman respects her mother.

(3) The capital of every country is a large city.

Contrast:

(4)* This capital of every country is a large city.

If a necessary condition for sentence $\boldsymbol{\varphi}$ to have a truth value is that sentence $\boldsymbol{\psi}$ is true, we may say that $\boldsymbol{\varphi}$ *presupposes* $\boldsymbol{\psi}$, in a Strawsonian rather than Stalnakerian sense (Stalnaker's pragmatic presuppositions concern what speakers take as common ground in a conversation and do not involve truth value gaps). On this view, 'The coin in my pocket is shiny' presupposes 'There is exactly one coin in my pocket'.

Problems:

In at least some cases where this treatment implies presupposition failure and a truth value gap, we seem instead to have straightforward falsity:

(5) The master of the universe is present in this very room.

(6) I am the master of the universe.

Heim and Kratzer (p. 77) suggest that we can handle this problem by treating the ordinary term 'false' as applicable both to cases in which the semantics assigns F to a sentence and to cases in which the semantics leaves the sentence without a semantic value. This is unsatisfying for several reasons. One is that it does not explain what 'F' means in the semantics — presumably not *falsity*, otherwise the semantics would be incorrectly rejecting the claim that the sentence is false in cases of presupposition failure. Moreover, if the semantic values of all complex expressions are determined compositionally by functional application, as the paradigm requires, then the negations of (5) and (6) will also lack semantic values:

(7) The master of the universe is not present in this very room.

(8) I am not the master of the universe.

Thus if receiving no semantic value is a way of being false, then presumably (7) and (8) should count as false. But neither seems false, and at least (8) seems straightforwardly true.

One alternative is to depart from the strict functional application paradigm. For example, one might rule that the negation of something with no truth value is to be true, so that (7) and (8) would come out true. That strategy will require all sorts of further rulings, e.g. to handle:

(9) Anyone who is the master of the universe is very powerful.

Moreover, if one departs from the strict functional application paradigm, one can simply assign F to (5) and (6) by the stipulation:

$[\boldsymbol{\beta}(\mathbf{THE}\ \boldsymbol{\alpha})]^{I,g} = \mathrm{F}$ if $|\{a \in D_e: [\boldsymbol{\alpha}]^{I,g}(a) = \mathrm{T}\}| \neq 1$.

The effect of this is to assign the same truth values to these sentences as under the quantifier treatment, and the Strawsonian presupposition failures disappear.

Another possibility is to give a different semantic treatment to occurrences of definite descriptions in predicate position, as in (6), (8) and (9), although this is unattractively disunified and does not deal with the problem that (7) is not false.

Even if definite descriptions in natural languages are quantifiers, the problem remains of functional expressions in mathematical language. More generally, *could* there be a language with complex singular terms whose semantics is Strawsonian? What sort of propositions would they express?

Further reading:

Irene Heim and Angelika Kratzer, *Semantics in Generative Grammar* (Oxford: Blackwell, 1998), pp. 73-83.
Gary Ostertag, ed., *Definite Descriptions: A Reader* (Cambridge, Mass.: MIT Press, 1998).

Introduction to Semantics

Week 7: Adverbial Modifications

In what follows, grossly over-simplify by ignoring matters of tense, even though they are relevant to the problems at issue; things are difficult enough without them, and they can be factored back in later.

We are interested in the relation between (1) and (2):

(1) Mary walked.

(2) Mary walked quickly.

We don't want just to represent them as

(1a) Walked(Mary).

(2a) Walkedquickly(Mary).

where 'Walkedquickly is atomic, since this omits the overt semantic relationship between (1a) and (2a) and leads to an unacceptable proliferation of primitives when extended to other adverbial modifications, as in:

(3) Mary walked quickly home from school by the river with her books while talking to her friends.

Nor can we treat 'quickly' in (2) as predicated of Mary:

(2b) Walked(Mary) & Quick(Mary).

For, although (2b) would help to capture the entailment relations between (1) and (2), by the same procedure we would have to formalize (4) and (5) as (4b) and (5b) respectively:

(4) Mary walked quickly and talked slowly.

(5) Mary walked slowly and talked quickly.

(4b) Walked(Mary) & Quick(Mary) & Talked(Mary) & Slow(Mary)

(5b) Walked(Mary) & Slow(Mary) & Talked(Mary) & Quick(Mary)

For not only do (4b) and (5b) seem to be individually inconsistent, they are logically equivalent to each other — unlike (4) and (5).

Flat-footed application of type theory: the predicate modifier approach

Treat **walked** as of type <e, t> and **quickly** as of type <<e, t>, <e, t>>, so that for any expression **α** of type <e, t>, **α quickly** (e.g. **walked quickly**) — thought of as **quickly(α)** (e.g. **quickly(walked)**) is also of type <e, t>. Thus the difference between (4) and (5) would be represented as:

(4c) **(quickly(walked))(Mary) & (slowly(talked))(Mary)**

(5c) **(slowly(walked))(Mary) & (quickly(talked))(Mary)**

Semantics: for any interpretation *I* and assignment *g*, $[\textbf{quickly}]^{I,g}$ is a function f: $D_{<e,t>} \to D_{<e,t>}$. Roughly speaking, for any action A, if the function h: $D_e \to D_t$ maps all and only the individuals in D_e who do A to truth, then f(h): $D_e \to D_t$ maps all and only the individuals in D_e who do A quickly to truth.

This account can be generalized to more complex examples, such as (3). A less complex case is:

(6) Mary walked quickly to Oxford.

We want to treat **to Oxford** on a par with **quickly**, as of type <<e, t>, <e, t>>. Since **Oxford** is of type e, we treat **to** as of type <e, <<e, t>, <e, t>>>. Thus (6) is represented as:

(6c) **((to(Oxford))((quickly(walked))))(Mary)**

Semantics: $[\textbf{to}]^{I,g}$ is a function i: $D_e \to D_{<<e,t>, <e, t>>}$. Roughly speaking, for any individual o and action A, if the function h: $D_e \to D_t$ maps all and only the individuals in D_e who do A to truth, then (i(o))(h): $D_e \to D_t$ maps all and only the individuals in D_e who do A to o to truth.

One reason why this sort of semantics may not quite work is that there may be two actions A and A* such that exactly the same individuals in the domain do A as do A*, but some do A quickly without doing A* quickly or vice versa.

Can we solve this problem by moving from extensions to intensions? For if it is *necessary* that exactly the same individuals do A as do A*, it is less clear how anyone could do A quickly without doing A* quickly or vice versa. But even this might not be enough. For suppose that there can be a model in which the domain of individuals with respect to any given world is restricted to the things that do both A and A* in that world. Within the model, there is no intensional difference between doing A and doing A*, but there may still be an intensional difference between doing A quickly and doing A* quickly.

A different objection to the approach is that it does not help us understand the relevant logical relations, e.g. that (2) entails (1). To derive them, one must assume corresponding entailments in the metalanguage, e.g. that doing A quickly entails doing A. This is one standard motivation for switching to an alternative approach that postulates much more deeply hidden semantic structure.

Event analyses (Reichenbach, Davidson, Parsons, …)

We represent (1) and (2) as follows, where the variable '**e**' is restricted to particular events:

(1d) $\exists$ **e (Walking(e) & Agent(Mary, e))**

(2d) $\exists$ **e (Walking(e) & Agent(Mary, e) & Quick(e))**

This provides a good explanation of why (2) explains (1) but not vice versa. Similarly, (4) and (5) are represented as:

(4d) $\exists$ **e (Walking(e) & Agent(Mary, e) & Quick(e)) &** $\exists$ **e (Talking(e) & Agent(Mary, e) & Slow(e))**

(5d) $\exists$ **e (Walking(e) & Agent(Mary, e) & Slow(e)) &** $\exists$ **e (Talking(e) & Agent(Mary, e) & Quick(e))**

This pair seems to capture the appropriate logical relations. The approach can be extended to more complex cases, such as (3) and (6). For example, (6) might be represented thus:

(6d) $\exists$ **e (Walking(e) & Agent(Mary, e) & Quick(e) & To(e, Oxford))**

Once we have a systematic way of deriving such first-order representations of natural language sentences, we can apply the type-theoretic apparatus for first-order languages to obtain formal semantic treatments for the original natural language sentences. Nevertheless, some questions remain.

How far do we expect semantic analysis to explain entailments?

Consider (7)-(9):

(7) Mary notoriously tortured prawns.

(8) Mary allegedly tortured prawns.

(9) Mary tortured prawns.

(7) entails (9); (8) does not. We do not seem warranted in assuming that this difference is explained by a covert difference in semantic structure between (7) and (8); it may just be the result of a difference in lexical semantics between ‘notoriously’ and ‘allegedly’. So what is so bad about attributing the entailment of (1) by (2) to the lexical semantics of ‘quickly’?

Are events sufficiently fine-grained?

(6) might be true at the same time as (10), with reference to the same past occasion:

(10) Mary travelled slowly to Oxford.

Just as (6) is represented as (6d), (10) will be represented as (10d):

(10d) $\exists$ **e (Travelling(e) & Agent(Mary, e) & Slow(e) & To(e, Oxford))**

But on Davidson’s view Mary’s walking quickly to Oxford may be the very same event as her travelling slowly to Oxford; thus (6d) and (10d) have the same verifying instance. But how can a single event be both quick and slow? Indeed, with reference to the same occasion, (11) may be false:

(11) Mary travelled quickly to Oxford.

Now (11) will be represented as (11d):

(11d) $\exists$ **e (Travelling(e) & Agent(Mary, e) & Quick(e) & To(e, Oxford))**

But if (6d) is true and the walking is a travelling, how can (11d) be false?

How far can we read metaphysics off semantics and common sense?

(12) If ‘Mary ran’ is true, there are events.
(13) If Mary ran, ‘Mary ran’ is true.
(14) Mary ran.
----
(15) There are events.

Further reading

Donald Davidson, ‘The logical form of action sentences’, in his *Essays on Actions and Events* (Oxford: Clarendon Press, 1980).
Donald Davidson, ‘The method of truth in metaphysics’, in his *Inquiries into Truth and Interpretation* (Oxford: Clarendon Press, 1984).
Terence Parsons, *Events in the Semantics of English: A Study in Subatomic Semantics* (Cambridge, Mass.: MIT Press, 1994).
Mark Platts, *Ways of Meaning* (London: Routledge & Kegan Paul, 1979), ch. 8.
Hans Reichenbach, *Elements of Symbolic Logic* (New York: The Free Press, 1947).

Introduction to Semantics

Week 8: Dynamic Semantics / Discourse Representation Theory

(1a) A man is walking. He is tired.

(1b) **$\exists$ x (man(x) & walking(x))   tired(x)**

(1c) **$\exists$ x (man(x) & walking(x) & tired(x))**

(2a) A farmer owns a donkey. He beats it.

(2b) **$\exists$ x $\exists$ y (farmer(x) & owns(x, y) & donkey(y))   beats(x, y)**

(2c) **$\exists$ x $\exists$ y (farmer(x) & owns(x, y) & donkey(y) & beats(x, y))**

(3a) If a farmer owns a donkey, he beats it.

(3b) **($\exists$ x $\exists$ y (farmer(x) & owns(x, y) & donkey(y))) $\rightarrow$ beats(x, y)**

(3c) **$\forall$ x $\forall$ y ((farmer(x) & owns(x, y) & donkey(y)) $\rightarrow$ beats(x, y))**

(1d) **[x: man(x), walking(x)]   [u: tired(u)]**

(1e) **[x, u: man(x), walking(x), tired(u)]**

(1f) **[x, u: u = x, man(x), walking(x), tired(u)]**

(1g) **[x: man(x), walking(x), tired(x)]**

(2d) **[x, y: farmer(x), owns(x, y), donkey(y)]   [u, v: beats(u, v)]**

(2e) **[x, y, u, v: farmer(x), owns(x, y), donkey(y), beats(u, v)]**

(2f) **[x, y, u, v: u = x, v = y, farmer(x), owns(x, y), donkey(y), beats(u, v)]**

(2f) **[x, y: farmer(x), owns(x, y), donkey(y), beats(x, y)]**

DRSs and DRS-conditions

DRS = discourse representation structure

A DRS is a pair <U, C> where U is a set of *discourse referents*, and C is a set of *DRS-conditions*.
NB Discourse referents are abstract representational devices, not the referents in the usual sense; for example, in (2e) the discourse referent '**x**' is not a farmer and the discourse referent '**y**' is not a donkey. The DRS exists whether or not there really is a farmer and a donkey.

If **P** is an n-place predicate in the language and $\mathbf{x_1}, \ldots, \mathbf{x_n}$ are discourse referents, then $\mathbf{P(x_1, \ldots, x_n)}$ is a DRS-condition.
If **x** and **y** are discourse referents, then $\mathbf{x = y}$ is a DRS-condition.

Semantics for DRSs

A *model* is a pair <D, I>, where D is a domain of individuals and I is an interpretation function mapping each atomic n-place predicate in the language to a set of n-tuples of members of D (and each name to a member of D).

An *embedding* of a DRS <U, C> in a model <D, I> is a function f such that domain(f) = U and range(f) $\subseteq$ D.

A DRS K is *true* in a model M iff some embedding of K in M *verifies* K relative to M.

Henceforth we leave the relativization to M tacit.

We must now explain what it is for an embedding f to verify a DRS-condition.

f verifies $\mathbf{P(x_1, \ldots, x_n)}$ iff $\langle f(\mathbf{x_1}), \ldots, f(\mathbf{x_n}) \rangle \in I(\mathbf{P})$.

f verifies $\mathbf{x = y}$ iff $f(\mathbf{x}) = f(\mathbf{y})$.

f verifies a DRS <U, C> iff f verifies every condition in C.

NB Verification here is a purely non-epistemic notion.

Examples

If f is an embedding of (1g), domain(f) = {**x**};
f verifies (1g) iff $f(\mathbf{x}) \in I(\textbf{man})$, $f(\mathbf{x}) \in I(\textbf{walking})$ and $f(\mathbf{x}) \in I(\textbf{tired})$.
There is such an embedding iff there is a member d of D such that $d \in I(\textbf{man})$, $d \in I(\textbf{walking})$ and $d \in I(\textbf{tired})$, i.e. iff (1c) is true (some man is walking and tired), i.e. we recover the desired truth-conditions.

If f is an embedding of (1f), domain(f) = {**x**, **u**};
f verifies (1f) iff $f(\mathbf{u}) = f(\mathbf{x})$, $f(\mathbf{x}) \in I(\textbf{man})$, $f(\mathbf{x}) \in I(\textbf{walking})$ and $f(\mathbf{u}) \in I(\textbf{tired})$.
There is such an embedding iff (1c) is true, i.e. we recover the desired truth-conditions.

If f is an embedding of (1e), domain(f) = {**x**, **u**};
f verifies (1g) iff $f(\mathbf{x}) \in I(\textbf{man})$, $f(\mathbf{x}) \in I(\textbf{walking})$ and $f(\mathbf{u}) \in I(\textbf{tired})$.
There is such an embedding iff some man is walking and something is tired (different truth-conditions).

If f is an embedding of (2g), domain(f) = {**x**, **y**};
f verifies (2g) iff $f(\mathbf{x}) \in I(\textbf{farmer})$, $\langle f(\mathbf{x}), f(\mathbf{y})\rangle \in I(\textbf{owns})$, $f(\mathbf{y}) \in I(\textbf{donkey})$, $\langle f(\mathbf{x}), f(\mathbf{y})\rangle \in I(\textbf{beats})$.
There is such an embedding iff (2c) is true (some man owns and beats some donkey), i.e. we recover the desired truth-conditions.
(2f) and (2e) also follow the pattern of (1f) and (1e).

<u>Conditionals</u>

We return to the analysis of (1a), and stipulate:
K and K* are DRSs, then $K \Rightarrow K^*$ is a DRS-condition.

If K is a DRS $\langle U, C\rangle$ and f and g are embeddings, then f[K]g (g extends f to K) iff $f \subseteq g$ and domain(g) = domain(f) $\cup$ U.

An embedding f verifies $K \Rightarrow K^*$ iff for every embedding g such that f[K]g and g verifies K there is an embedding h such that g[K*]h and h verifies K*.

(3d) **[: [x, y: farmer(x), owns(x, y), donkey(y)] ⇒ [u, v: beats(u, v)]]**

(3e) **[: [x, y: farmer(x), owns(x, y), donkey(y)] ⇒ [u, v: u = x, v = y, beats(u, v)]]**

(3e) is true iff some embedding of (3e) verifies (3e).
Since (3e) is of the form$\langle \{\}, C\rangle$, its only embedding is {}.
Thus (3e) is true iff {} verifies (3e).
Let K = **[x, y: farmer(x), owns(x, y), donkey(y)]**
K* = **[u, v: u = x, v = y, beats(u, v)]]**
{} verifies (3e) iff for every embedding g such that {}[K]g and g verifies K there is an embedding h such that g[K*]h and h verifies K*.
For any embedding g, {}[K]g iff domain(g) = {**x**, **y**}.
For any such g and embedding h, g[K]h iff

$$h(\mathbf{x}) = g(\mathbf{x}),\ h(\mathbf{y}) = g(\mathbf{y}) \text{ and domain}(h) = \{\mathbf{x}, \mathbf{y}, \mathbf{u}, \mathbf{v}\}.$$

Thus (3e) is true iff for every embedding g such that domain(g) = {**x**, **y**} and g verifies K there is an embedding h such that domain(h) = {**x**, **y**, **u**, **v**}, h(**x**) = g(**x**), h(**y**) = g(**y**) and h verifies K*. But h verifies K* iff h(**u**) = h(**x**), h(**v**) = h(**y**), $\langle h(\mathbf{u}), h(\mathbf{v})\rangle \in I(\mathbf{beats})$.
For given g, there is an embedding h such that domain(h) = {**x**, **y**, **u**, **v**}, h(**x**) = g(**x**), h(**y**) = g(**y**), h(**u**) = h(**x**), h(**v**) = h(**y**) and $\langle h(\mathbf{u}), h(\mathbf{v})\rangle \in I(\mathbf{beats})$ only if $\langle g(\mathbf{x}), g(\mathbf{y})\rangle \in I(\mathbf{beats})$.
Conversely, if $\langle g(\mathbf{x}), g(\mathbf{y})\rangle \in I(\mathbf{beats})$ then there is an embedding h such that domain(h) = {**x**, **y**, **u**, **v**}, h(**x**) = g(**x**), h(**y**) = g(**y**) and h(**u**) = h(**x**), h(**v**) = h(**y**), $\langle h(\mathbf{u}), h(\mathbf{v})\rangle \in I(\mathbf{beats})$, since we can stipulate that h(**u**) = h(**x**) = g(**x**) and h(**v**) = h(**y**) = g(**y**).
Thus (3e) is true iff for every embedding g such that domain(g) = {**x**, **y**} and g verifies K, $\langle g(\mathbf{x}), g(\mathbf{y})\rangle \in I(\mathbf{beats})$.
But g verifies K iff $g(\mathbf{x}) \in I(\mathbf{farmer})$, $\langle g(\mathbf{x}), g(\mathbf{y})\rangle \in I(\mathbf{owns})$, $g(\mathbf{y}) \in I(\mathbf{donkey})$.
Hence (3e) is true iff for every embedding g, if domain(g) = {**x**, **y**}, $g(\mathbf{x}) \in I(\mathbf{farmer})$, $\langle g(\mathbf{x}), g(\mathbf{y})\rangle \in I(\mathbf{owns})$ and $g(\mathbf{y}) \in I(\mathbf{donkey})$, then $\langle g(\mathbf{x}), g(\mathbf{y})\rangle \in I(\mathbf{beats})$.
Hence (3e) is true iff for all members c and d of D,
if $c \in I(\mathbf{farmer})$, $\langle c, d\rangle \in I(\mathbf{owns})$, $d \in I(\mathbf{donkey})$, then $\langle c, d\rangle \in I(\mathbf{beats})$,
i.e. iff (3c) is true (for every farmer and every donkey, if the former owns the latter then the former beats the latter), i.e. the desired truth-conditions.

This also shows that (3e) is equivalent to the simpler:

(3f) **[: [x, y: farmer(x), owns(x, y), donkey(y)] ⇒ [: beats(x, y)]]**

The semantics of conditional DRS-conditions is *dynamic* in the sense that the verification of the consequent (when it occurs) ‘builds on’ the verification of the antecedent rather than starting from the same point as the verification of the antecedent started from.

<u>Negation</u>

In what follows, names will be treated as discourse referents that every embedding f is required to map to their assigned values in the model, so that f(**Pedro**) = I(**Pedro**). Thus (4a) is represented as (4b):

(4a) Pedro owns a donkey.

(4b) **[Pedro, y: owns(Pedro, y), donkey(y)]**

We cannot represent (5a) as (5b):

(5a) Pedro does not own a donkey.

(5b) **[Pedro, y: not-owns(Pedro, y), donkey(y)]**

For (5b) is true provided that there is some donkey d that Pedro does not own, since it is verified by an embedding f such that f(**y**) = d. By contrast, (5a) is true only if no donkey is owned by Pedro. Rather, we apply negation to DRSs themselves:

If K is a DRS, ¬K is a DRS-condition.

We can now represent (5a) as:

(5c) **[Pedro: ¬[y: owns(Pedro, y), donkey(y)]]**

f verifies ¬K iff no embedding g such that f[K]g verifies K.

f is an embedding of (5c) that verifies (5c) iff domain(f) = {**Pedro**}, f(**Pedro**) = I(**Pedro**) and f verifies **¬[y: owns(Pedro, y), donkey(y)]**, i.e. no embedding g such that domain(g) = {**Pedro**, **y**} and g(**Pedro**) = f(**Pedro**) verifies **[y: owns(Pedro, y), donkey(y)]**. But g verifies **[y: owns(Pedro, y), donkey(y)]** iff <g(**Pedro**), g(**y**)> $\in$ I(**owns**), g(**y**) $\in$ I(**donkey**). Since there is such an embedding g iff for some d $\in$ D, <I(**Pedro**), d> $\in$ I(**owns**) and d $\in$ I(**donkey**), (5c) has the desired truth-conditions.

NB With both negative and conditional DRS conditions, we must ensure that discourse referents cannot be introduced more than once in the same DRS.

Accessibility

Why is (7a) bad in a way in which (6a) is not, when ‘it’ is anaphoric on ‘a donkey’?

(6a) Pedro owns a donkey. It is sad.

(7a)* Pedro doesn’t own a donkey. It is sad.

After initial merging, they are represented as follows:

(6b) **[Pedro, y, u: owns(Pedro, y), donkey(y), sad(u)]**

(7b) **[Pedro, u: ¬[y: owns(Pedro, y), donkey(y)], sad(u)]**

When we (attempt to) add the anaphoric links, the results are:

(6c) **[Pedro, y, u: u = y, owns(Pedro, y), donkey(y), sad(u)]**

(7c)* **[Pedro, u: u = y, ¬[y: owns(Pedro, y), donkey(y)], sad(u)]**

The difference must be that **y** in (6c) is accessible to **u** in some sense in which it is not in (7c). This idea can be made more rigorous as follows.

We define accessibility as:
the smallest reflexive transitive relation between DRSs such that
a. if K contains K* $\Rightarrow$ K** then K is accessible to K* and K* is accessible to K**
b. if K contains ¬K* then K is accessible to K*
(more clauses will need to be added for a language with more operations for making DRS conditions from DRSs).
If <U, C> and <U*, C*> are DRSs, and **x** and **x*** are discourse referents such that **x** $\in$ U and **x*** $\in$ U*, then **x** is accessible to **x*** iff <U, C> is accessible to <U*, C*>.

Now **y** is accessible to **u** in (6c) but not in (7c) (and the semantics does not make the value of ‘**y**’ outside the negative condition in (7c) covary with its value inside).
Thus (7c) but not (6c) violates *the Accessibility Constraint*:

A discourse referent that represents an anaphoric expression must be equated with a discourse referent accessible to it.

Further reading

L.T.F Gamut, *Logic, Language, and Meaning, Volume 2: Intensional Logic and Logical Grammar* (Chicago: University of Chicago Press, 1991), pp. 264-97 (with exercises).

Bart Geurts, ‘Piggyback anaphora: accessibility, binding, and bridging’, http://www.ru.nl/ncs/bart/papers/piggyback.pdf

Hans Kamp and Uwe Reyle, *From Discourse to Logic* (Dordrecht: Kluwer, 1993).