A Static Version of Heim's File Card Semantics

Daniel Rothschild and Seth Yalcin

The dynamic semantics presented by Heim (1982) is justly celebrated for its elegant treatment of donkey anaphora and presupposition projection. It is often thought that such a treatment is not easily replicable in a classical semantic framework. Treatments of donkey anaphora in a static framework tend, instead, to rely on an e-type theory of descriptions. We want to show that there is an extremely simple, static semantics for Heim's syntax which is essentially equivalent to Heim's own. This is not a mere technical exercise, but an indication that the moves Heim makes are *easily* and *naturally* replicable in a static system. In a our static system, the distinctive features of Heim's are mimicked by making the meaning of sentences relative to a domain, a set of variables, that—intuitively–are under discussion, and giving non-standard semantics for connectives and quantifiers.[1] The necessity of the latter trick falls in line with the observations by Scott Soames and Mats Rooth, that Heim's treatment of presupposition projection depends on particular stipulations about the connectives.

1 Unified Syntax

(1) Variables, $x_1, x_2 \ldots$

(2) n- place predicates, $F \ldots$

(3) Atomic formulas are of the form $F(x_1, \ldots x_n)$

(4) For any sentences α and β, we can also get $\neg\alpha$, $\alpha \wedge \beta$, and Every(α, β)

[1]The use of contexts as parameters is borrowed form Yalcin (2007), which we have show can be expanded into a system that replicates Veltman's treatment of epistemic modals.

2 Heim's Semantics

A Heimian context H is a pair consisting of a sets of world/assignment function pairs, c along with sets of individuals, d. We assume (as a constraint on legitimate contexts) that for any context $H = (H_c, H_d)$ any variable x not in H_d is free in H_c. The definition of free is as follows:

(5) A variable x is *free* in c if for any two assignment functions f and f' that differ only in their assignment to x, and any world w if $(f, w) \in H_c$ so is (f', w)

Each sentence is associated with a context change potential (CCP), a rule for changing a context. We will describe these rules in two parts: first we will describe how sentences affect the domain, then we will define how they alter the set of world-assignment function pairs. In general we write an update of a context h by a sentence α as $h[\alpha]$. We also can refer to the updates of the domain and the set of world-assignment function pairs separately, so that $H = (H_c[\alpha], H_d[\alpha])$. We first define how the the domain is updated when a sentence α is asserted.

(6) $H[F(x_1, \ldots, x_n)]_d = H_d \cup \{x_1, \ldots x_n\}$

(7) $H[\alpha \wedge \beta]_d = H[\alpha]_c \cup H[\beta]_d$

(8) $H[\neg\alpha]_d = H[\text{Every}(\alpha, \beta)]_d = H_d$

These rules make no reference to the word-assignment function pairs h_c, as we will see some of the rules for updating h_c do make reference to the domain, so cannot be defined independently.

The rule for atomic formulas is simple: it reduces the context, c, to the pairs that satisfy the atomic formula.

(9) $H[F(x_1, \ldots, x_n)]_c = \{(f, w) \in H_c : F(f(x_1) \ldots f(x_n)) = 1 \text{ in } w\}$

Conjunction (also consecutive assertion) is just consecutive action on c:

(10) $H[\alpha \wedge \beta]_c = ((H[\alpha])[\beta])_c$

The negation rule is a bit complicated as it incorporates Heim's rules for "existential closure" of chapter 2, and thus needs to make use of the domain d of H.

(11) $H[\neg\alpha]_c = \{(f,w) \in H_c :$ there is no (f',w) s.t. $f' \sim_{H_d} f$ such that $(f',w) \in H[\alpha]_c\}$
($f' \sim_{H_d} f$ is shorthand for f' agrees with f on all variables in H_d.)

What this rule does is checks that no way of assigning referents to any of the variables not in the domain of c can support α. This effectively causes any free variables in α which are not in the domain to be existentially closed before the negation..

Next is the quantification rule. Recall first that Heim's treatment of donkey anaphora under quantifiers and conditionals depends on *unselective* quantification, following Lewis (1975). So, all quantifiers, objectual or adverbial, quantify over sets of variables. We give here the rule for universal quantification, but it is easy to see how to adopt this to other quantifiers (with attendant problems with proportionality, etc). We also here need to make use of Dom(c) since variables in the nuclear that are not in Dom(c) have an effective form of existential closure.

(12) $h[\text{Every}(\alpha,\beta)]_c = \{(f,w) \in H_c :$ for every (f',w) such that $f' \sim_{H_d} f$ and $(f',w) \in h[\alpha]_c$, there is some (f'',w) such that $f'' \sim_{H[\alpha]_d} f'$ and $(f'',w) \in H[\alpha][\beta]_c\}$

Note that this rule effectively *selects* all free variables in the restrictor to be part of the quantification, and all free variables in the nuclear scope get existentially closed.

The semantics is incomplete as we have not yet defined how domains are determined by context updates. Here are the rules.

There are two aspects to a Heimian context, the set of world assignment function pairs, and the domain. It is clear from the rules about above that the domain expands with updates. However, the set of world assignment function pairs can only contract, in fact, it always contracts by intersective update with a given set of such pairs for each sentence.

Proposition 1. *For any sentence ϕ and domain d there exists a set of world assignment function pairs S such that for any $h = (c,d)$, $H[\phi] = (H_c \cap S, H[\phi]_d)$*

Proof. This follows immediately from the results of the next section. □

3 Static Semantics

Like Heim we will use *domains*, which are sets of variables. To replicate Heim's semantics we also need a syntactic theory of domain: a mapping from sentences onto all the variables that are in their domain. We do not need to give this in terms of the context change potential of sentences but rather we can just give compositional rules for determining the domain of any complex sentence:

(13) $d(\alpha)$ is defined as follows:

a. If α is atomic then, $d(\alpha)$ = all the variables appearing in α.

b. If α is of the form $\beta \wedge \gamma$, then $d(\alpha) = d(\beta) \cup d(\gamma)$.

c. If α is of the form Every(β, γ) or $\neg\beta$ then $d(\alpha) = \emptyset$.

In addition to the domain *of* a sentence, there is a also a domain *parameter* with respect to which we evaluate a sentence. This parameter is the only departure in type from a classical semantics.

Our semantics for atomic formula is fully classical:

(14) $[\![R(x_1, \ldots x_n)]]\!]^{w,f,d}$ is true iff $R(f(x_1), \ldots, f(x_n))$ is true in w.

Like Heim, our negation is non-classical: it makes reference to the the domain parameter d.

(15) $[\![\neg\alpha]\!]^{w,f,d}$ is true iff there is no $f' \sim_d f$ such that $[\![\neg\alpha]\!]^{w,f',d}$ is true. ($f' \sim_d f$ means that f' agrees with f on all variables in d.)

This rule ensures that variables not in the domain are existentially closed under negation. The rules for conjunction and universal quantification both shift the domain:

(16) $[\![\alpha \wedge \beta]\!]^{w,f,d}$ is true iff $[\![\alpha]\!]^{w,f,d}$ is true and $[\![\beta]\!]^{w,f,d \cup d(\alpha)}$ is true

This ensures that the domain is updated before the second conjunct is evaluated.

The rule for universal conjunction is much like Heim's

(17) $[\![\text{Every}(\alpha, \beta)]\!]^{w,f,d}$, is true iff for every $f' \sim_d f$ s.t. $[\![\alpha]\!]^{w,f',d}$ is true there is a $f'' \sim_{d \cup d(\alpha)} f'$ s.t. $[\![\beta]\!]^{w,f'',d \cup d(\alpha)}$ is true

A few notes: we do not think of the f parameter as being fixed by context in the usual way. Rather like Heim we think of context, H, as an ordered pair consisting of a set of world-assignment function pairs, H_c, and a domain, H_d. In a world w and given context H a sentence α is true iff there is an (f, w) in H_c s.t. $[\![\alpha]\!]^{f,w,H_d}$ is true. In other words the constraints put by the context on assignment functions do not prevent there being an assignment function that satisfies α.

We can derivatively define an update rule for this semantics as follows, we represent the update of $H = (c, d)$ with α as follows:

(18) $H[\![\alpha]\!] = (\{(w, f) \in H_c : [\![\alpha]\!]^{w,f,H_d}$ is true $)\}, H_d \cup d(\alpha))$

The natural question is whether this update rule, derived from the static semantics, is equivalent to the update rule directly defined by Heim. We now prove that they are equivalent.

Proposition 2. *For any Heiman context $H = (H_c, H_d)$ and any sentence α, $H[\alpha] = H[\![\alpha]\!]$*

Proof. We prove by induction on complexity. The base case is trivial.
Our inductive hypothesis, then, is that for any H and some ϕ, ψ, $H[\phi] = H[\![\phi]\!]$. That the domain d of H is always shifted in the right way follows immediately by definition, so we will not write out domains when describing Heimian contexts.
Negation:
$H[\![\neg\phi]\!]_c = (\{(w, f) \in H_c : [\![\neg\phi]\!]^{w,f,H_d}$ is true$\}$
$= \{(w, f) \in H_c :$ there is no $f' \sim_d f$ s.t. $[\![\phi]\!]^{w,f',H_d}$ is true$\}$
$= \{(w, f) \in H_c :$ there is no $f' \sim_d f$ s.t. $(w, f) \in H[\![\alpha]\!]_c$
$= \{(w, f) \in H_c :$ there is no $f' \sim_d f$ s.t. $(w, f) \in H[\alpha]_c)$ (By induction hypothesis)
$= H[\neg\alpha]$
Conjunction:
$H[\![\phi \wedge \psi]\!]_c = \{(w, f) \in H_c : [\![\phi]\!]^{w,f,H_d}$ is true and $[\![\phi]\!]^{w,f,H_d \cup d(\alpha))}$ is true$\}$
$= \{(w, f) \in H_c : (w, f) \in H[\![\phi]\!]_c$ and $(w, f) \in H^{d(\alpha)}[\![\psi]\!]$ where $H^{d(\alpha)}$ is identical to H except that its domain includes $d(\alpha)\}$
$= \{(w, f) \in H_c : (w, f) \in H[\![\phi]\!][\![\psi]\!]_c\}$
$= \{(w, f) \in H_c : (w, f) \in H[\phi][\psi]_c\} = H[\phi \wedge \psi]_c$
Universal quantification:
$H[\![\text{Every}(\phi, \psi)]\!]$
$= \{(w, f) \in H_c :$ for every $f' \sim_{H_d} f$ s.t. $[\![\alpha]\!]^{w,f',d}$ is true there is a $f'' \sim_{d \cup d(\alpha)}$ s.t. $[\![\beta]\!]^{w,f',d \cup d(\alpha)}$ is true $\}$
$= \{(w, f) \in H_c :$ for every $f' \sim_{H_d} f \in H[\![\alpha]\!]$ there is a $f'' \sim_{H_d \cup d(\alpha)} f'$ s.t. $(f'', w) \in H^{d(\alpha)}[\![\beta]\!]_c\}$
$= \{(w, f) \in H_c :$ for every $f' \sim_{H_d} f \in H[\![\alpha]\!]$ there is a $f'' \sim_{H_d \cup d(\alpha)} f'$ s.t. $(f'', w) \in H[\![\alpha]\!][\![\beta]\!]_c\}$
$= \{(w, f) \in H_c :$ for every $f' \sim_{H_d} f \in H[\alpha]$ there is a $f'' \sim_{H_d \cup d(\alpha)} f'$ s.t. $(f'', w) \in H[\alpha][\beta]_c\}$
$= H[\text{Every}(\phi, \psi)]$ □

4 Notes

We cannot capture certain aspects of presupposition projection, since we only have domains available at local context and no further information. So, we cannot capture 1) gender/number presuppositions of pronouns, 2) descriptive presuppositions in definite descriptions, 3) presuppositions where there is no explicit antecedent (e.g. propositional presuppositions such as that triggered by "know", "stop" etc). (PS projection thus requires more complex semantic types than handling donkey anaphora allows, a surprising result in some ways.)

We capture, however, definite indefinite distinction in "local" context. A variable is if it's in domain (of its local context), indefinite otherwise.

5 Purely Intersective Updates

We can also give a semantics that allows us to have purely intersective updates (seemingly inequivalent as has symmetric conjunction—weird). Sentences are again true relative to worlds, assignment functions and domain. Domains of sentences are defined syntactically as above. We now require that an atomic formula is only true if all its variables are in its domain.

(19) $[\![R(x_1, \ldots x_n)]\!]^{w,f,d}$ is true iff $R(f(x_1), \ldots, f(x_n))$ is true in w and $x_1 \ldots x_n$ are in d.

Negation is again non-classical:

(20) $[\![\neg\alpha]\!]^{w,f,d}$ is true iff there is no $f' \sim_d f$ such that $[\![\neg\alpha]\!]^{w,f',d\cup d(\alpha)}$ is true.

Conjunction however is classical, unlike in standard dynamic systems.

(21) $[\![\alpha \wedge \beta]\!]^{w,f,d}$ is true iff $[\![\alpha]\!]^{w,f,d}$ is true and $[\![\beta]\!]^{w,f,d}$ is true.

Quantification is similar to that in the previous system.

(22) $[\![\text{Every}(\alpha, \beta)]\!]^{w,f,d}$, is true iff for every $f' \sim_d f$ s.t. $[\![\alpha]\!]^{w,f',d\cup d(\alpha)}$ is true if there is a $f'' \sim_{d\cup d(\alpha)} f'$ s.t. $[\![\beta]\!]^{w,f'',d}$ is true.

How do we understand this semantics? Think of contexts as sets of triplets of worlds, assignment functions, and domains, (w, f, d).

We'll call such a context an S. Updates are now purely intersective: when we update with a sentence α we eliminate all triplets that α is not true on.

How do we understand such contexts and what counts "truth" in a context in this system?

A sentence α with no variables in the context is true at w iff there exists an f s.t. $[\![\alpha]\!]^{w,f,d}$ is true.

References

Heim, Irene (1982). *The Semantics of Definite and Indefinite Noun Phrases*. Ph.D. thesis, University of Massachusetts, Amherst.

Lewis, David (1975). Adverbs of quantification. In Edward L. Keenan (ed.), *Formal Semantics of Natural Language*. Cambridge University Press.

Yalcin, Seth (2007). Epistemic modals. *Mind*, 116:983–1026.