Dynamic semantics: Schlenker v. Schlenker

Barker seminar, NYU, Wed, 24 Feb, 2021

Plan:

- Schlenker *against* dynamic semantics: presupposition
- Schlenker *for* dynamic semantics: donkey anaphora
- Bonus paper: George 2014 on the middle Kleene strategy
- The dynamics of information flow in composition
  - Rory Harder: making Heim eliminative is not straightforward!

Schlenker *against* dynamic semantics: 2007

- Brillant and compelling
- Profoundly original work
- (Obviously, I'm going to come at it hard)
- Many slightly different versions scattered all over the literature
- Main point for today: generalizing Schlenker's explanatory challenge
  - PS: Heim's theory is insufficiently explanatory, why not *and**?
  - Schlenker offers an explanation for presupposition projection
  - But that explanation doesn't generalize to the linear sensitivity of donkey anaphora

Schlenker's criticisms of Stalnaker 1974

- Stalnaker 74: Context update is rational reaction to information
  - E.g., it is rational to update with the first conjunct immediately
  1. Moldavia is a monarchy and Moldavia's monarch is a king.
- Three problems (my names):
  - *When*: embedded conjunctions don't trigger global update
    2. None of my students is rich and proud of it.

    My diagnosis: this is a subproblem of the third problem
  - *Or*: when processing a disjunction, why do we update with the negation of the first disjunct?
    3. Either this building doesn't have a bathroom, or it's in a funny place.

    Schlenker does not explicitly discuss how he handles this, but we will discuss it below
  - The "make-as-if" problem: why is it rational to update on subexpressions that do not ever get asserted? In hypothetical contexts, we treat the conjunction locally as if it had been asserted in an unembedded context
- These problems motivate a grammatically-regulated solution like Heim's or Schlenker's

Schlenker's Criticism of Heim 1983: too powerful

- Sure, elegant prediction of projection facts: update with the left conjunct, then update with the right
- "it fails to explain why there couldn't be a"deviant" conjunction *and** which performed these operations in the opposite order."
    1. Moldavia's monarch is a man and* Moldavia has a king.
    2. Moldavia has a king and* Moldavia's monarch is a man.
- "her theory may be descriptively adequate, but it is not quite explanatory"

Schlenker 2008a, **the explanatory challenge**:

"Find an algorithm that predicts how any operator transmits presuppositions once its syntax and its classical semantics have been specified"

Today: **the *generalized* explanatory challenge**:

Find an algorithm that simultaneously predicts both how an operator transmits presuppositions and how antecedents bind donkey pronouns.

- For instance, a theory in which flipping a single switch simultaneously reverses the direction of the sensitivity to linear order of local presupposition satisfaction and of donkey anaphora

- What we don't want: a theory in which is is possible to describe, e.g., a language in which local presupposition satisfaction works left to right, but donkey anaphora works right to left

Schlenker's proposal: transparency theory

- "the only information that needs to be updated [in the course of a conversation] concerns *the words that the speech act participants have pronounced*" [Stalnaker's manifest update]
- "meaning is not dynamic, though sentence comprehension might well be"
- "We will also do without any recourse to trivalence"
- "a presupposition failure will simply come out as the violation of a certain pragmatic principle."

"Be articulate!": If possible, say "p and [p]q" (where [p] is a presupposition)

Principle of Transparency:

- saying "α [p]q . . . " is only ok if

$$C \Vdash \alpha(p \text{ and } \gamma)\beta \Leftrightarrow \alpha\gamma\beta$$

  for all choices of γ and β.
    - This is the second formulation ("concise"), page 338
    - C is the context set
    - γ same type as p
    - $\alpha(p$ and $\gamma)\beta$ and $\alpha\gamma\beta$ well formed

- the equivalence can depend on facts in the context

Examples:

1. a. Context: everyone is aware that P has cancer.
   b. #P is sick and he knows it.
   c. $C \Vdash$ (P is sick and γ) $\beta \Leftrightarrow \gamma\beta$ (α is null)
   d. P knows that he is sick.

2. a. Context: nothing is assumed about P's health.
   b. #P has cancer and he is sick and he knows it.
   c. $C \Vdash$ P has cancer and (he is sick and γ) $\beta \Leftrightarrow$ P has cancer and $\gamma\beta$
   d. P has cancer and he knows that he is sick.

3. a. Context: nothing is assumed about P's health.
   b. #If P has cancer, he is sick and he knows it.
   c. $C \Vdash$ If P has cancer, (he is sick and γ) $\beta \Leftrightarrow$ If P has cancer, $\gamma\beta$
   d. If P has cancer, he knows that he is sick.

4. a. Context: no assumptions
   b. #Either this building doesn't have a bathroom, or this building has a bathroom and it is in a funny place.
   c. $C \Vdash$ Either this building doesn't have a bathroom, or (this building has a bathroom and γ) $\beta \Leftrightarrow$ Either this building doesn't have a bathroom, or γ β
   d. Either this building doesn't have a bathroom, or it is in a funny place.

Schlenker's Theorem 1:

- A context/sentence pair satisfies Transparency iff the sentence admits the context on Heim's theory.

- Proof depends on adding "A or B" to Heim's theory as $(c{+}A)\cup(c{+}\neg A){+}B$

How the whole thing works:

- "Be articulate!" requires that every presupposition be explicitly stated
- Explicit statement can be elided when the left context makes it redundant
- Heim's notion of local satisfaction is moved from the semantics to the realm of Gricean alternatives

Some discussion points:

1. Radical implicature: if you think this is a Manner implicature, you have to compute Gricean implicature at every point in the unfolding utterance

2. What rules out a theory Transparency*, on which $...[p]q\beta$ is ok only if for all α, γ: $C \Vdash \alpha(\gamma \text{ and } p)\beta \Leftrightarrow \alpha\gamma\beta$?

   Zhouye Zhao: "Should definability really be counted as a source of overgeneration?"

3. Is Schlenker's proposal incompatible with Heim's?

   After all, he proves that his predictions are exactly those of Heim. Here's a possibility:

   - Semantics:
     - Q. What explains presupposition projection?
     - A. Heim's CCPs.
   - Metasemantics:
     - Q. What explains the specific forms of Heim's CCPs? That is, why *and* and not *and**?
     - A. Gricean pressures as described by Schlenker.

4. What about accommodation?

   - Even Geoge Lakoff could win.
   - $C \nVdash$ (Lakoff is the most unlikely person to win and $\gamma)\beta \Leftrightarrow \gamma\beta$
   - Apparently, Transparency can routinely be violated
   - Like Heim, Schlenker assumes that we retroactively revise the context set as Lewis taught us
   - At least Schlenker doesn't have to resort to local accommodation

5. Schlenker eschews trivalent accounts.

   Sentences with presupposition violations are degraded, but grammatical

   What is the semantic value of a senence when evaluated against a context that does not satisfy transparency?

   For instance, what is the semantic value of

   7. Ann doesn't have children and her children are asleep.

6. Is local presupposition satisfaction always left to right?

   8. The king of Moldavia will be pretty powerful if Moldavia is a monarchy. [Zhao]

7. The solution is insufficiently general.

   - It does not generalize to donkey anaphora.
   - Surely the reason presupposition projection is sensitive to linear order
   - . . . and that donkey anaphora is sensitive to linear order is that
   - they both stem from the order in which expressions are evaluated
   - What kind of theory can explain both presupposition and anaphora?

Schlenker *for* dynamic semantics: indefinites

- The presupposition discussion completely ignored variables and assignments
- The discussion of doneky anaphora will ignore presupposition projection
- Problem domain: donkey anaphora

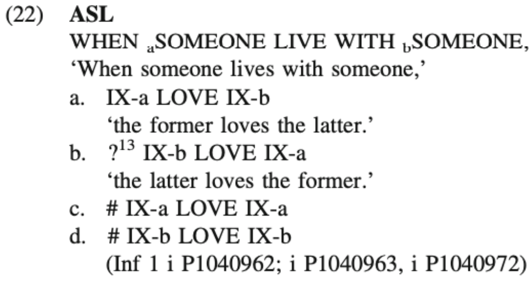
  - E-type pronouns
    1. If a farmer owns a donkey, she ~~[the farmer]~~ beats it ~~[the donkey]~~.
       - Semantics built on top of situations
       - Pronouns always have a silent nominal that makes them definite descriptions
  - Dynamic anaphora
    2. If a farmer^1 owns a donkey^2, she$_1$ beats it$_2$.
       - The local context for the consequent contains only assignments on which the value of x_1 is a donkey-owning farmer and the value of x_2 is a farmer-owned donkey
- Issue in the debate: the “formal link”
  - Code for: is this pragmatic or semantic?
    3. a. Every man who has a wife is sitting next to her.
       b. Every married man is sitting next to her.

Pronouns in American Sign Language (ASL)

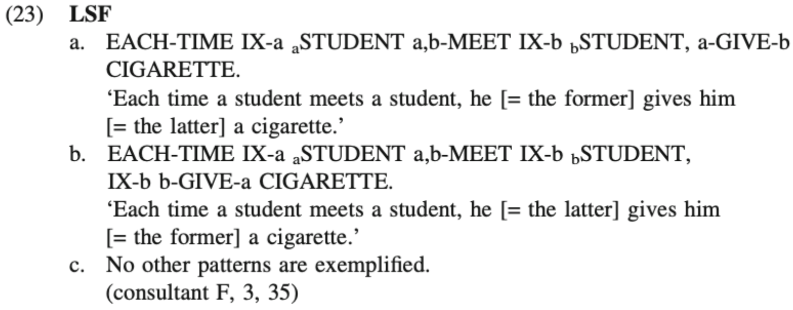
- Proper names, definite descriptions, and indefinite descriptions can be associated with a locus in signing space
- Pronouns can be accompanied by an indication of a locus
- Because there is no principled bound on the number of distinct loci, ASL can disambiguate sentences that are ambiguous in a spoken language.
- Predictions of the two main approaches to donkey anaphora (p. 354):

  a. If pointing in ASL allows a pronoun to recover/resolve the ellipsis, it should be possible to point to the same locus twice, but still have distinct individuals
  b. If pointing in ASL is a form of anaphora, pointing twice at the same locus can only result in referring to a single individual

[show video of Jon Lamberton signing examples 22a and 22b]



(22) **ASL**
WHEN ${}_{a}$SOMEONE LIVE WITH ${}_{b}$SOMEONE,
‘When someone lives with someone,’
a. IX-a LOVE IX-b
‘the former loves the latter.’
b. ?[13] IX-b LOVE IX-a
‘the latter loves the former.’
c. # IX-a LOVE IX-a
d. # IX-b LOVE IX-b
(Inf 1 i P1040962; i P1040963, i P1040972)



(23) **LSF**
a. EACH-TIME IX-a ${}_{a}$STUDENT a,b-MEET IX-b ${}_{b}$STUDENT, a-GIVE-b CIGARETTE.
‘Each time a student meets a student, he [= the former] gives him [= the latter] a cigarette.’
b. EACH-TIME IX-a ${}_{a}$STUDENT a,b-MEET IX-b ${}_{b}$STUDENT, IX-b b-GIVE-a CIGARETTE.
‘Each time a student meets a student, he [= the latter] gives him [= the former] a cigarette.’
c. No other patterns are exemplified.
(consultant F, 3, 35)

.

George 2014: Trivalent connectives allow pointwise Heim

George, B. R. 2014. Some remarks on certain trivalent accounts of presupposition projection Journal of Applied Non-Classical Logics 24.1–2, 86–117. http://dx.doi.org/10.1080/11663081.2014.911521

- Super clear and orderly thinking
- Seeks to generate Heimian update functions from a restrictive template
- “How constraining an account is critically depends not just on the connectives it defines, but on the space of possible connectives from which those are understood as being drawn.”
- Provides a framework in which updates are always equivalent to trivalent truth conditions
- The logical connectives are asymmetric (linear order sensitive).
  - Dynamic composition of content
  - Without context change
- Observes that “One interesting constraint imposed by the use of trivalent truth functions is that they make presupposition projection *extensional*, in the sense that the presuppositions of compound sentences will be determined on a world-by-world basis.” [what we’ve been calling “pointwise” or “distributive” –CB]

- “The rule that I will employ, which I call Peters-Kleene function deployment, exploits **Krahmer’s ‘middle Kleene’ intuition**, combining the intuition behind the strong Kleene logic (cf. Kleene, 1952; Beaver and Krahmer, 2001) with a left-to-right incremental evaluation strategy.”
- Middle Kleene: weak on the left, strong on the right, because…
- “information about earlier arguments of a function can inform the evaluation of later arguments, but not vice versa”

Two comments:

- There is no need for the logical connectives to be aware of presupposition failure–conjunction has nothing to do with presupposition! (Solution: use a Maybe monad, keep logical operators pure)
- Like Schlenker’s solution, no obvious path to generalize to anaphora—yet…

General discussion

- The moral of the bathroom example
    1. Either this building doesn’t have a bathroom, or it’s in a funny place.
- If GSV can get bathroom examples, doesn’t that undercut the central motivation for dynamic anaphora?
- Well, GSV can only “get” the bathroom exx. if we assume that the local context after update with “Either this building doesn’t have a bathroom, or”—that is, the local context for “it’s in a funny place”—is a context in which the assignment functions have an index that points to bathrooms. But this is exactly what we already need in order to handle a goat walking in (“It stinks!”).
- Having grammatically-managed discourse referents (indefinites, negation) alongside inferential discourse referents (goats, bathrooms) are in tension. If you have to have the inferential kind (goats and bathrooms), how can you justify also having the grammatically-managed kind? If you have both, what exactly is the mechanism by which they get combined into a comprehensive half-grammar half-inference discourse state?
- Hope to address this on Lewis day, 10 Mar

Explanatory problem, or engineering problem?

- Dynamics without context change
  - Refactoring Heim 1983
    - Making presupposition pointwise is easy (George, and below)
      - trade off: connectives must be lazy (= mixed Kleene)
    - Indefinites are harder
      - global variable
      - threaded eval (Heim)
      - State monad
- Information flow
  - Pure strict compositionality: the meaning of a complex expression depends only on the meanings of its immediate subparts, and on nothing else:
    $$[\![\beta \cdot \gamma]\!] = f([\![\beta]\!], [\![\gamma]\!])$$
  - But: the content of expressions depends on context, so we need…

Downward information flow

- Contextual information must at least flow downward from the top:

$$[\![\beta \cdot \gamma]\!]^c = f([\![\beta]\!]^c, [\![\gamma]\!]^c)$$

  - Information flows downwards, but not upwards or sideways
  - Familiar from the standard treatment of intensionality
  - In functional programming talk, a Reader monad

`* H&K Function Application (FA):`

- In H&K, also used for assignment functions, that won't work in a donkey-anaphora setting
- As defined, only copies assignment coordinate from mother to daughters, but H&K clearly intend to generalized to any and all evaluation coordinates (see, e.g., p. 307).
- Bracket: H&K's Intionsional Function Application

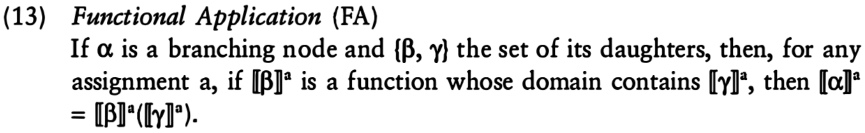
`Only relevant for predicates like attitude verbs that take intensional arguments`

Sideways information flow: indefinite novelty

$${}^i[\![\beta \cdot \gamma]\!]^k = f({}^i[\![\beta]\!]^j, {}^j[\![\gamma]\!]^k)$$

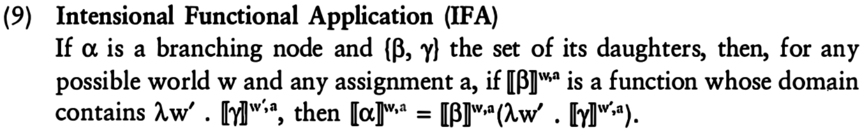
- The least unused index depends on expressions to the left
- should remind you of DPL's relational semantics
- "global variable" = threaded computation = State monad

Figure 1: Information flow in a reader monad



(13) *Functional Application* (FA)
If α is a branching node and $\{\beta, \gamma\}$ the set of its daughters, then, for any assignment a, if $[\![\beta]\!]^{a}$ is a function whose domain contains $[\![\gamma]\!]^{a}$, then $[\![\alpha]\!]^{a} = [\![\beta]\!]^{a}([\![\gamma]\!]^{a})$.

Figure 2: Heim and Kratzer 1998:95 Function Application

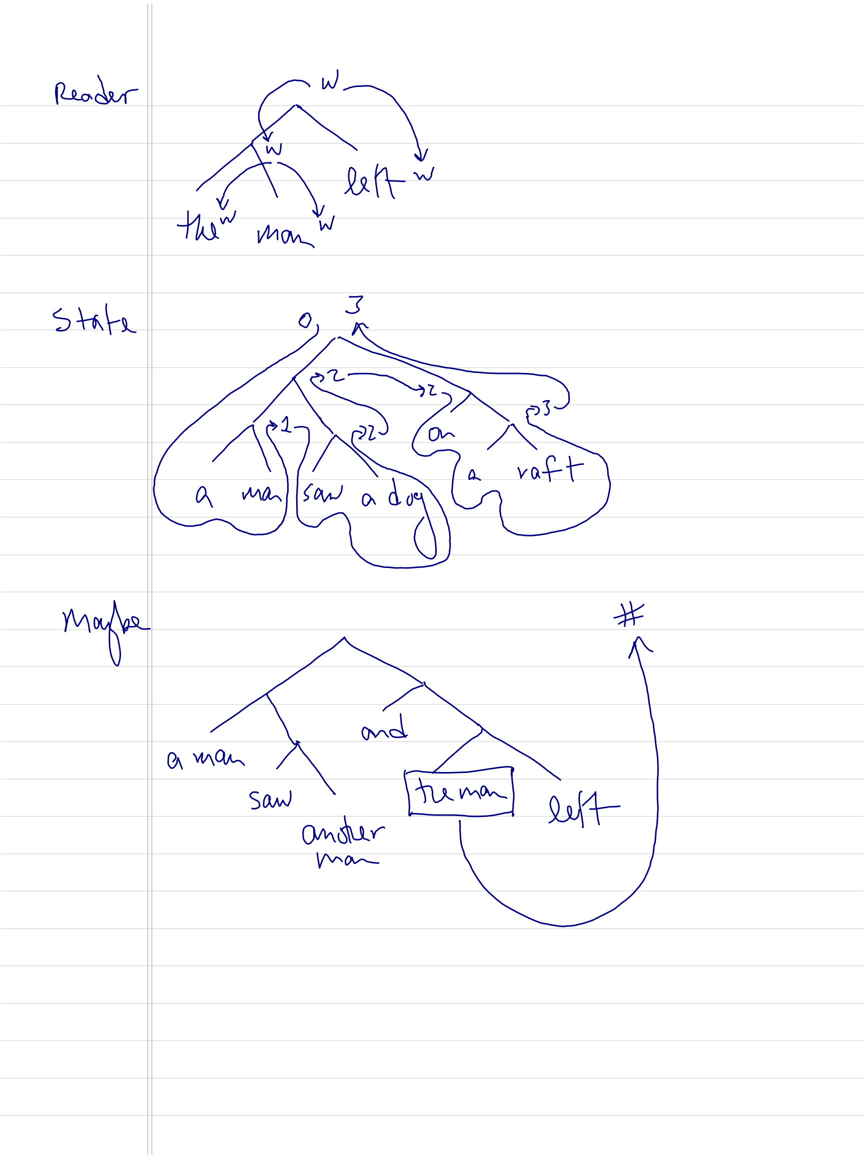


(9) Intensional Functional Application (IFA)
If α is a branching node and $\{\beta, \gamma\}$ the set of its daughters, then, for any possible world w and any assignment a, if $[\![\beta]\!]^{w,a}$ is a function whose domain contains $\lambda w' . [\![\gamma]\!]^{w',a}$, then $[\![\alpha]\!]^{w,a} = [\![\beta]\!]^{w,a}(\lambda w' . [\![\gamma]\!]^{w',a})$.

Figure 3: Heim and Kratzer 1998:308 Intensional Function Application

Exception handling

- Presupposition failure, character death, division by zero
- Independent of Reader and State information flow
- # = `Maybe` monad = throw/catch



.

Summary of information flow:

- intensionality <=> reader
- presupposition projection <=> exceptions
- donkey anaphora <=> state

Today's fragment:

```
-- Heim 1983, presup only, pointwise
--   presup projection implemented as lazy logical operators which are pure wrt presup failu

import Control.Monad.Reader
import Control.Monad.Trans.Maybe

type M a = MaybeT (Reader Int) a

mand, mor, mif :: M Bool -> M Bool -> M Bool
mand ml mr = ml >>= (\l -> if l then mr else return False)
mor ml mr = ml >>= (\l -> if l then return True else mr)
mif ml mr = ml >>= (\l -> if l then mr else return True)

mnot :: M Bool -> M Bool
mnot ma = ma >>= (\l -> return (not l))

fa a f = f a            -- backwards function application
pm l r x = l x && (r x) -- predicate modification

mallowable :: M (Int -> Bool)
mallowable = ask >>= (\w -> return (\i -> i < w))

the :: M (Int -> Bool) -> M Int
the mr = do
  r <- mr
  let ext = filter r [1..5] in
    if length ext == 1          -- presupposes singleton extension
      then return (head ext)
      else mzero

run :: M Bool -> [Maybe Bool]
run s = map (\w -> runReader (runMaybeT s) w) [1..5]

s1 = liftM fa (return 1) <*> mallowable
-- 1 is allowable
-- run s1 ~~> [Just False,Just True,Just True,Just True,Just True]
```

```
s2 = mnot (liftM fa (return 2) <*> mallowable)
-- 2 is not allowable
-- run s2 ~~> [Just True,Just True,Just False,Just False,Just False]

s3 = liftM fa (the mallowable) <*> (return odd)
-- the allowable number is odd
-- run s3 ~~> [Nothing,Just True,Nothing,Nothing,Nothing]

s4 = mand s1 s2
-- 1 is allowable and 2 is not allowable
-- run s4 ~~> [Just False,Just True,Just False,Just False,Just False]

s5 = mand s4 s3
-- (1 is allowable and 2 is not allowable) and the allowable number is odd
-- run s5 ~~> [Just False,Just True,Just False,Just False,Just False]

s6 = mand s3 s4
-- the allowable number is odd and (1 is allowable and 2 is not allowable)
-- run s6 ~~> [Nothing,Just True,Nothing,Nothing,Nothing]
```