Two souls of disjunction

Towards a state-monadic update semantics

Patrick D. Elliott

July 3, 2019

Asymmetries in Language: Presuppositions and beyond – Berlin

Two broad traditions addressing semantics and pragmatics of disjunction, with little-to-no overlap:

Scalar implicature literature concerned with deriving exclusive readings and ignorance inferences while retaining inclusive disjunction as the basic meaning of natural language "or" (Sauerland 2004).

Dynamic semantics literature concerned with deriving facts concerning presupposition projection in disjunctive sentences (Heim 1983, Beaver 2001).

- Both approaches to disjunction seem necessary, but it's not obvious that the two are even *compatible* – the scalar implicature literature takes as its starting point that "or" is $\vee$, whereas dynamic semantics departs from this orthodoxy.
- Relatedly, dynamic semantics has been criticized (see, e.g., Schlenker 2009), because the dynamic entry for disjunction can't be derived from logical disjunction.

Our goal

In this talk, we'll sketch a way of systematically lifting a fragment into the dynamic world. This will be successful, with the exception of disjunction. We'll suggest that this tension can be resolved by integrating *exhaustification*.

- A brief recap of the Heim-Karttunen projection rules, update semantics, and the explanatory problem for dynamic semantics.
- Tracking information growth via the State monad.
- A type-shift from propositional to dynamic connectives (dlift), and a bad prediction for disjunction.
- Resolving the tension via exhaustification, and some possible empirical payoffs.

The dynamic approach to presupposition projection

(1) a. **Negation**

If A_π, then a sentence of the form “not A” presupposes π.

b. **Conjunction**

If A_π, and B_ρ, then a sentence of the form “A and B” presupposes π, and unless A entails ρ, also presupposes ρ

c. **Implication**

If A_π and B_ρ, then a sentence of the form “If A then B” presupposes π, and unless A entails ρ, also presupposes ρ.

d. **Disjunction**

If A_π, and B presupposes B_ρ, then a sentence of the form “A or B” presupposes π, and unless “not A” entails ρ, also presupposes ρ.

Some illustrations

(2)	Paul didn't stop vaping.	*Paul vaped*
(3)	Paul vaped and Paul didn't stop vaping.	*Presuppositionless*
(4)	If Paul and Sophie vaped, then Paul would never stop vaping.	*Presuppositionless*
(5)	Either Paul never vaped, or Paul stopped vaping.	*Presuppositionless*

- Classical dynamic semantics (Groenendijk & Stokhof 1991, Heim 1983, a.o.) builds the projection rules directly into the semantics of the connectives.
- Sentences themselves express *updates* of the common ground.
- Presuppositions place definedness conditions on updates.
- Dynamic connectives manipulate the input of subsequent juncts based on the output of previous juncts, thereby getting the projection facts.

(6) Partial assertion operator (def.)

$\mathbb{A}\,\phi \coloneqq \lambda c \,.\, c \subseteq \mathsf{dom}\,\phi \,.\, c \cap \{\, w \mid \phi\, w \,\}$

(7) $[\![\text{Paul stopped vaping}]\!] = \lambda w \,:\, \mathsf{vaped}_w\ \mathsf{p} \,.\, \neg\ \mathsf{vapes}_w\ \mathsf{p}$

(8) $\mathbb{A}\ [\![\text{Paul stopped vaping}]\!] = \lambda c \,:\, c \subseteq \{\, w \mid \mathsf{vaped}\ \mathsf{p} \,\}$

$\qquad . \, c \cap \{\, w \mid \neg \mathsf{vapes}_w\ \mathsf{p} \,\}$

(9) $\text{not } u \coloneqq \lambda c \,.\, c \setminus (u\ c)$

Take the result of updating c with u, and subtract the result from c.

(10) $u \text{ and } v \coloneqq \lambda c \,.\, (v \circ u)\ c$

First update c with u, then update the result with v.

(11) $\text{if } u \text{ then } v \coloneqq (\text{not } u\ c) \cup (u \text{ and } v)\ c$

Update c with u, and subtract the result from c – store this as c′. Next, update c with u, and then update the result with v – store this as c″. Finally, union c′ and c″.

(12) $u \text{ or } v \coloneqq \lambda c\ .\ u\ c \cup v\ (\text{not } u\ c)$

Update c with u – store this as c′. Next, update c with u, subtract the result from c, and update this with v – store this as c″. Union c′ and c″.

(13) Paul never vaped or Paul stopped vaping. *presuppositionless*

(14) a. $\text{not}\ \mathbb{A}\ [\![\text{Paul vaped}]\!] = \lambda c\ .\ c \smallsetminus (c \cap \{\, w \mid \mathsf{vaped}_w\ \mathsf{p} \,\})$

b. $\mathbb{A}\ [\![\text{Paul stopped vaping}]\!] = \lambda c\ :\ c \subseteq \{\, w \mid \mathsf{vaped}_w\ \mathsf{p} \,\}$

$.\ c \cap \{\, w \mid \neg\ \mathsf{vapes}_w\ \mathsf{p} \,\}$

c. (14*a*) or (14*b*)

$$= \lambda c\ :\ \overbrace{(c \cap \{\, w \mid \mathsf{vaped}_w\ p \,\}) \subseteq \{\, w \mid \mathsf{vaped}_w\ \mathsf{p} \,\}}^{\top}$$

$$.\ (c \smallsetminus (c \cap \{\, w \mid \mathsf{vaped}_w\ \mathsf{p} \,\}))$$

$$\cup\ ((c \cap \{\, w \mid \mathsf{vaped}_w\ \mathsf{p} \,\}) \cap \{\, w \mid \neg\ \mathsf{vapes}_w\ \mathsf{p} \,\})$$

- Linear asymmetries built into the entry for each individual connective; concomitantly, easy to define "deviant" dynamic connectives that are truth-conditionally adequate but get the projection facts wrong.
- E.g., *reverse dynamic conjunction*.

(15) $u \ \mathsf{rand}\ v \coloneqq \lambda c\,.\,(u \circ v)\,c$

- Just by reversing the order of function composition, we predict that a subsequent conjunct could satisfy the presuppositions of previous conjuncts.

State-monadic update semantics

- Here we'll attempt to (partially) resolving the explanatory problem by stipulating the linear order of information growth *once*.
- Concretely, we follow Shan (2002), Asudeh & Giorgolo (2016), and especially Charlow (2014) in using a *monad* to extend a pure, Montagovian fragment.
- You don't have to care about what a monad is for the purposes of this talk. Here is what we're going to introduce:
  - We answer the question "what kind of semantic object is an update?" by providing a type constructor for updates.
  - An injection function, for lifting values into trivial updates.
  - A way of doing function application in the update-semantic space.

Type constructor for updates:

(16) $\mathsf{U\ a} ::= \{\,\mathsf{s}\,\} \rightarrow (\mathsf{a} * \{\,\mathsf{s}\,\})$

Injection function from an ordinary value a to an update:

(17) $a^{\rho} \coloneqq \lambda c\,.\,\langle a, c\rangle$

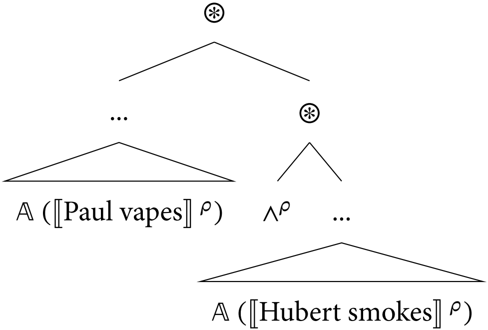
(18) $m \circledast n \coloneqq \lambda c\,.\,\langle \mathsf{A}\; x\; y, c'' \rangle$

$$\begin{aligned} \text{for} \quad \langle x, c' \rangle &\coloneqq m\; c \\ \langle y, c'' \rangle &\coloneqq n\; c' \end{aligned}$$

- Takes two *updates* m and n as inputs.
- The input context set c is first fed into m, returning a potentially updated context c'.
- c' is fed into n, returning a potentially updated output context c''.
- The ordinary values contained in the updates undergo ordinary function application.

(19) $\mathbb{A}\ m \coloneqq \lambda c\ .\ \langle p, c' \cap p \rangle$ for $\langle p, c' \rangle \coloneqq m\ c$

$$\lambda c\ .\ \left\langle \begin{matrix} \lambda w\ .\ \mathsf{smokes}_w\ \mathsf{h} \wedge \mathsf{vapes}_w\ \mathsf{p} \\ (c \cap \{\, w \mid \mathsf{smokes}_w\ \mathsf{h} \,\} \cap \{\, w \mid \mathsf{vapes}_w\ \mathsf{p} \,\}) \end{matrix} \right\rangle$$



How do we get the dynamic connectives in this system?

To simplify the technical details, we have to assume that the lexical entry for each (classical) propositional connective comes with an additional parameter r. This is harmless.

$$\begin{aligned}
\text{not}_r\ p &:= \lambda r\,.\,r \setminus p \\
p\ \text{and}_r\ q &:= \lambda r\,.\,r \cap q \cap p \\
\text{if}\ p\ \text{then}_r\ q &:= \lambda r\,.\,(r \setminus p) \cup q \\
p\ \text{or}_r\ q &:= \lambda r\,.\,r \cap (p \cup q)
\end{aligned}$$

If the additional parameter is saturated by D_{s} we just get...the ordinary propositional connectives.

$$
\begin{aligned}
\mathsf{not}_r\ p\ D_{\mathsf{s}} &= p^- \\
(p\ \mathsf{and}_p\ q)\ D_{\mathsf{s}} &= p \cap q \\
(\mathsf{if}\ p\ \mathsf{then}_r\ q)\ D_{\mathsf{s}} &= p^- \cup q \\
(p\ \mathsf{or}_r\ q)\ D_{\mathsf{s}} &= p \cup q
\end{aligned}
$$

Now let's define our function dlift_n.

$$(\mathsf{dlift}_1\ f)\ m \coloneqq \lambda c\ .\ \left\langle \begin{matrix} f\ \ p\ D_s, \\ f c' c \end{matrix} \right\rangle \quad \text{for } \langle p, c' \rangle \coloneqq m\ c$$

$$m\ (\mathsf{dlift}_2\ f)\ n \coloneqq \lambda c\ .\ \left\langle \begin{matrix} f\ q\ p\ D_s, \\ f c'' c' c \end{matrix} \right\rangle \qquad \begin{aligned} \text{for } \langle p, c' \rangle &\coloneqq m\ c \\ \langle q, c'' \rangle &\coloneqq n\ c \end{aligned}$$

Informally – in the ordinary dimension, the connective f has its inner argument saturated by D_s, returning the ordinary propositional meaning. In the contextual dimension, the propositional connective's inner-argument is saturated by the input context, in which case we get...the Heimian connectives (with provisos).

$$\text{(20)}\quad \mathsf{dlift}_1\ \mathsf{not}_r = \lambda m\ .\ \lambda c\ .\ \left\langle \begin{matrix} p^- \\ c \setminus c' \end{matrix} \right\rangle \quad \text{for } \langle p, c' \rangle \coloneqq m\ c$$

$$\text{(21)}\quad \mathsf{dlift}_2\ \mathsf{and}_r = \lambda n\ .\ \lambda m\ .\ \lambda c\ .\ \left\langle \begin{matrix} p \cap q \\ (c \cap c') \cap c'' \end{matrix} \right\rangle \quad \begin{matrix} \langle p, c' \rangle \coloneqq m\ c \\ \langle q, c'' \rangle \coloneqq n\ c' \end{matrix}$$

$$\text{(22)}\quad \mathsf{dlift}_2\ (\mathsf{if...then}_r) = \lambda n\ .\ \lambda m\ .\ \lambda c\ .\ \left\langle \begin{matrix} p^- \cup q \\ (c \setminus c') \cup c'' \end{matrix} \right\rangle \quad \begin{matrix} \langle p, c' \rangle \coloneqq m\ c \\ \langle q, c'' \rangle \coloneqq n\ c' \end{matrix}$$

Illustration

...

$$\lambda c \,.\, \left\langle \begin{array}{l} c \setminus (c \cap \{\, w \mid \mathsf{vaped}_w\ p + s \,\}) \\ \cup\, (c \cap \{\, w \mid \mathsf{vaped}_w\ p + s \,\}) \cap \{\, w \mid \neg\ \mathsf{vapes}_w\ p \,\} \end{array} \right\rangle$$

$$\lambda c \,.\, \left\langle \begin{array}{l} \{\, w \mid \mathsf{vaped}_w\ p + s \,\}, \\ c \cap \{\, w \mid \mathsf{vaped}_w\ p + s \,\} \end{array} \right\rangle$$

$\mathbb{A}$ Paul and Sophie vaped

$\mathsf{dlift}_2\ (\mathsf{if...then}_r)$

$$\lambda c : c \subseteq \{\, w \mid \mathsf{vaped}_w\ p \,\} \\ \left\langle \begin{array}{l} \{\, w \mid \neg\ \mathsf{vapes}_w\ p \,\} \\ c \cap \{\, w \mid \neg\ \mathsf{vapes}_w\ p \,\} \end{array} \right\rangle$$

$\mathbb{A}$ Paul stopped vaping

Post dlift, if...then feeds c updated with *Paul and Sophie vape* as the input context to the conditional consequent, locally satisfying the presupposition.

The Heim rule for disjunction, recast in a state-monadic setting:

$$\mathsf{or}_{Heim} \coloneqq \lambda n \,.\, \lambda m \,.\, \lambda c \,.\, \left\langle \begin{matrix} p \cup q \\ c \cap (c' \cup c'') \end{matrix} \right\rangle \quad \begin{matrix} \langle p, c' \rangle \coloneqq m\ c \\ \langle q, c'' \rangle \coloneqq n\ (c \setminus c') \end{matrix}$$

Application of dlift to classical disjunction:

$$\mathsf{dlift\ or} \coloneqq \lambda n \,.\, \lambda m \,.\, \lambda c \,.\, \left\langle \begin{matrix} p \cup q \\ c \cap (c' \cup c'') \end{matrix} \right\rangle \quad \begin{matrix} \langle p, c' \rangle \coloneqq m\ c \\ \langle q, c'' \rangle \coloneqq n\ c' \end{matrix}$$

Not correct! Predicts that the local context for the second disjunct is just c updated with the first disjunct.

(23) #Either Paul vaped or Paul stopped vaping.

predicted to be presuppositionless

Enter exh

- Maybe this prediction isn't as bad as it seems. There is independent need for an exhaustification operator exh.
- How should we define exh in a system with updates? We'll suggest the following entry:

(24) $\textsf{exh}\ m \coloneqq \lambda c\, .\langle p, c' \rangle$

$$\langle p, c' \rangle \coloneqq m\ (c \cap \bigwedge_{q \in \textsf{excl}\ p} \{\, \textsf{not}\ q\, \})$$

- Quasi formally:
  - exh takes an update m as its prejacent, and updates the input context c with the implicatures of its prejacent, resulting in an updated context c'.
  - exh updates the context c' with its prejacent.
  - In the ordinary dimension, exh just returns its prejacent.

Prediction

Implicature computation can feed presupposition satisfaction.

Assumptions concerning alternatives

In a sentence of the form “P or Q”, P is an alternative to Q.

Perhaps counter-intuitively, this just follows from, e.g., Fox & Katzir’s (2011) algorithm for computing alternatives, since, at the point we reach the second disjunct, the first has been mentioned, and therefore should be in the substitution source.

We now have everything we need to rescue our deviant prediction for conjunction. We simply assume that the second disjunct may be exhaustified.

Illustration

(25) Paul never vaped or Paul stopped vaping.

...

... ...

$\mathbb{A}\,(\llbracket \text{Paul never vaped} \rrbracket)^{\rho}$

dlift or $\quad \lambda c\,.\begin{cases} \left\langle \begin{matrix} \lambda w\,.\,\neg\,\text{vapes}_w\ \text{p} \\ (c \cap \{\, w \mid \neg\ \text{vaped}_w\ \text{p}\,\}) \end{matrix} \right\rangle & \top \\ \cap\,\{\, w \mid \neg\text{vapes}_w\ \text{p}\,\} & \\ \# & \text{otherwise} \end{cases}$

exh
(Paul never vaped
$\in$ alt (Paul stopped vaping))

$\lambda c\,.\begin{cases} \left\langle \begin{matrix} \lambda w\,.\,\neg\,\text{vapes}_w\ \text{p} \\ c \cap \{\, w \mid \neg\text{vapes}_w\ \text{p}\,\} \end{matrix} \right\rangle & c \subseteq \{\, w \mid \text{vaped}_w\ \text{p}\,\} \\ \# & \text{otherwise} \end{cases}$

$\mathbb{A}\,(\llbracket \text{Paul stopped vaping} \rrbracket)^{\rho}$

- Based on Fox & Katzir (2011), if in a sentence "A or B", A is an alternative to B, and A is *complex*, then every subconstituent of A should be an alternative to B.

Prediction

If, in a sentence of the form "A or B_π", the negation of A doesn't entail π, *but* the negation of some subconstituent of A entails π, the sentence as a whole should be presuppositionless.

The relevant cases involve a conjunctive first disjunct; the negation of the conjunctive disjunct is weak, but the negation of each conjunct is strong enough to satisfy the presupposition of the second disjunct.

(26) Either [Paul never vaped and he jogged every day],
or Paul stopped vaping. *Doesn't presuppose that Paul vaped*

(27) Either [there is no monarch and the country is in chaos],
or the monarch is in exile. *Doesn't presuppose a monarch*

(28) [Either nobody left early]
or only Josie left early. *Doesn't presuppose that J left early*

Conclusion

- Contrary to popular belief, we've shown that it's possible to have a dynamic semantics where the order of information growth *isn't* baked into the meaning of each individual connective. See Rothschild (2017) for a related proposal.
- Rather, in our state-monadic fragment, we baked the order of information growth into the state-sensitive application rule, and nowhere else.
- This still might be subject to an explanatory problem, but it certainly seems like an improvement.

- We showed that systematically lifting a static fragment into a dynamic one makes good predictions, with the notable exception of disjunction.
- We suggested that this tension can be resolved by incorporating *exhaustification* into a dynamic setting, which in any case we need.
- A detailed comparison to explanatory theories of presupposition projection, such as Schlenkerian local contexts, is next on the agenda.

Thanks especially to Paul Marty, Matt Mandelkern, and Daniel Rothschild for helpful discussion, as well as to audiences at ZAS and the Frankfurt semantics colloquium.

Implementation

A haskell implementation of the state monadic fragment outlined here can be found at: `https://github.com/patrl/monadicHeim`

- Asudeh, Ash & Gianluca Giorgolo. 2016. Perspectives. *Semantics and Pragmatics* 9.
- Beaver, David I. 2001. *Presupposition and assertion in dynamic semantics*. (Studies in logic, language, and information). Stanford, California: CSLI. 314 pp.
- Charlow, Simon. 2014. *On the semantics of exceptional scope*.
- Fox, Danny & Roni Katzir. 2011. On the characterization of alternatives. *Natural Language Semantics* 19(1). 87–107.
- Groenendijk, Jeroen & Martin Stokhof. 1991. Dynamic predicate logic. *Linguistics and Philosophy* 14(1). 39–100.
- Heim, Irene. 1983. On the projection problem for presuppositions. In *Proceedings of WCCFL 2*, 114–125. Stanford University.

- Rothschild, Daniel. 2017. A trivalent approach to anaphora and presupposition. unpublished manuscript.
- Sauerland, Uli. 2004. Scalar implicatures in complex sentences. *Linguistics and Philosophy* 27(3). 367–391.
- Schlenker, Philippe. 2009. Local contexts. *Semantics and Pragmatics* 2.
- Shan, Chung-chieh. 2002. A continuation semantics for interrogatives that accounts for Baker's ambiguity. In Brendan Jackson (ed.), *Salt xii*, 246–265. Massachussetts Institute of Technology: Linguistic Society of America.