Bilateral Dynamic Semantics

Anaphora and presupposition projection

Patrick Elliott[†] Lisa Hofmann[‡]

August 9, 2023

[†]Heinrich-Heine University Düsseldorf and [‡]the University of Stuttgart

Introduction

Recap

- The previous unit was devoted to exploring issues with classical Dynamic Semantics (DS) in depth, paying special attention to negation and disjunction.
- Problematic datapoints we'll tackle today:
  - Discourse anaphora with double-negation.
  - Bathroom disjunctions.
  - Program disjunctions.
- **Today's main goal:** improve on classical DS, by formulating a dynamic logic in which negation and disjunction are more classical.
- Concretely, we'll develop an account of anaphoric accessibility based on a simple, independently motivated account of presupposition projection, exploiting the idea that pronouns *presuppose* an antecedent.

Roadmap

1. Introduction
2. Parallels between anaphora and presupposition
3. Bilateral dynamic semantics
4. Disjunctions in bilateral DS
5. Asserting disjunctions
6. Extensions
7. Conclusion

Parallels between anaphora and presupposition

Projection and accessibility

- Karttunen (1973) established a set of desiderata for a theory of presupposition projection.
- Karttunen's observations — which we'll survey in the following — involve the conditions under which a complex sentence inherits the presuppositions of its component parts.
- Following others, we'll see a strong parallel between presupposition projection and anaphoric accessibility (see, e.g., Karttunen 1973, Heim 1983).

(1) Sam's dog is called Louise.
Presupposes: *Sam owns a dog.*

Parallels: conjunction

Asymmetrical accessibility in conjunctions:

(2) a. A woman walked in and she$_x$ sat down. *x can be bound*
b. She$_x$ walked in and a woman sat down. *x must be free*

Parallels: conjunction

Asymmetrical accessibility in conjunctions:

(2) a. A woman walked in and she$_x$ sat down. *x can be bound*
b. She$_x$ walked in and a woman sat down. *x must be free*

Asymmetrical projection in conjunctions:

(3) a. Sam has a dog, and her dog is called Louise. *no projection*
b. Sam's dog is called Louise, and she has a cute dog. *projection*

Parallels: conditionals

Asymmetric accessibility in conditionals (cf. *donkey cataphora*; Chierchia 1995):

(4) a. If a woman walked in, then she$_x$ sat down. *x can be bound*
b. If she$_x$ walked in, then a woman sat down. *x must be free*

Parallels: conditionals

Asymmetric accessibility in conditionals (cf. *donkey cataphora*; Chierchia 1995):

(4) a. If a woman walked in, then she$_x$ sat down. *x can be bound*
b. If she$_x$ walked in, then a woman sat down. *x must be free*

Asymmetric projection in conditionals:

(5) a. If Sam has a dog, her dog is called Louise. *no projection*
b. ??If Sam's dog is called Louise, then she has a dog cute. *projection*

Parallels: disjunction

Observed independently by Evans (1977) and Barbara Partee.

(6) a. Either there's no bathroom, or it_x's upstairs. *can be bound*

b. Either it_x's upstairs, or there's no bathroom. *must be free(?)*

Parallels: disjunction

Observed independently by Evans (1977) and Barbara Partee.

(6) a. Either there's no bathroom, or it_x's upstairs. *can be bound*
b. Either it_x's upstairs, or there's no bathroom. *must be free(?)*

(7) a. Either Sam doesn't have a dog,
or her dog is called Louise. *no projection*
b. Either Sam's dog is called Louise,
or she doesn't have a dog. *projection(?)*

An aside: Note that, unlike conjunction, projection in disjunction is arguably *symmetric* (Schlenker 2008, 2010; Janek's observation yesterday). We agree that accessibility patterns with projection in this respect.

Bonus: modal subordination

(8) Pauli doesn't have a car, but it would be red.

(9) Pauli doesn't have a car, but her car would be red.

We won't be able to account for such cases today, simply because this pattern of projection is beyond the remit of a simple theory of presupposition projection (but see Roberts 1987).

Splitting the difference

- **Previous responses:** develop an intricate dynamic semantics for anaphora; model a theory of presupposition projection on this (Heim 1983, van der Sandt 1992).
- **Today:** take a simple, independently motivated theory of presupposition projection (Strong Kleene trivalent semantics; Beaver & Krahmer 2001), and layer an account of anaphoric accessibility on top of *that*.
  - Central thesis: anaphoric pronouns presuppose the existence of a discourse referent (see also Rothschild 2017, Mandelkern 2022).

Bilateral dynamic semantics

Truth and falsity in DS

Recall, sentential meanings in classical DS are *mappings from possibilities to sets of assignments*.

(10) $[\![[x]\ ;\ [\mathit{WalkedIn}(x)]]\!]^{w,g} = \{\ h \mid g[x]h \wedge h(x)$ walked in$_w\ \}$

$\widehat{/}$undefinednessextitTruth is standardly defined as follows:

(11) A sentence ϕ is **True** at world w and assignment g
iff $[\![\phi]\!]^{w,g} \neq \emptyset$

In classical DS, there is no way to distinguish between falsity and presupposition failure; both simply amount to the empty set (van den Berg 1996).

Assignments and partiality

On day 1, we set up a dynamic semantics based on total assignments. *Familiarity* (i.e., the requirement for a linguistic antecedent) was assumed to be a syntactic constraint.

One straightforward way of encoding familiarity *semantically* is to shift to a setting with *partial* assignments.

(12)
$$\begin{bmatrix} x \rightarrow \textsf{Amethyst} \\ y \rightarrow \#_e \\ z \rightarrow \textsf{Ruby} \\ \dots \end{bmatrix}$$

Assignments don't provide (determinate) values for every variable. We implement this by assuming that the domain includes a privileged value $\#_e$, which stands in for an unknown/undefined individual.

Conventions for partial assignments

Notationally, we'll often simply omit those variables that are mapped to $\#_e$.

(13) $$[x \to \textsf{Amethyst}] := \begin{bmatrix} x \to \textsf{Amethyst} \\ y \to \#_e \\ z \to \#_e \\ \dots \end{bmatrix}$$

Conventions for partial assignments

Notationally, we'll often simply omit those variables that are mapped to $\#_e$.

$$(13) \qquad [x \to \textsf{Amethyst}] := \begin{bmatrix} x \to \textsf{Amethyst} \\ y \to \#_e \\ z \to \#_e \\ \dots \end{bmatrix}$$

The unique *initial assignment* maps *every variable to* $\#_e$.

$$(14) \qquad [] := \begin{bmatrix} x \to \#_e \\ y \to \#_e \\ z \to \#_e \\ \dots \end{bmatrix}$$

Partiality in the semantics

Two desiderata for the semantics of atomic sentences:

- In order to model familiarity, we'll assume that atomic sentences *presuppose* that any free variables $x_0, \ldots, x_n$ are defined at the input assignment.
- We also want to guarantee that the update induced by an atomic sentence doesn't introduce any new anaphoric information.

A bilateral semantics for atomic sentences

Our new dynamic framework is *bilateral*, which means that we'll recursively define $[\![.]\!]^w_+$ and $[\![.]\!]^w_-$ for all sentences of our fragment.

A bilateral semantics for atomic sentences

Our new dynamic framework is *bilateral*, which means that we'll recursively define $[\![.]\!]^w_+$ and $[\![.]\!]^w_-$ for all sentences of our fragment.

Like before, meanings are *updates*, i.e., functions from assignments to sets of assignments.

A bilateral semantics for atomic sentences

Our new dynamic framework is *bilateral*, which means that we'll recursively define $[\![.]\!]^w_+$ and $[\![.]\!]^w_-$ for all sentences of our fragment.

Like before, meanings are *updates*, i.e., functions from assignments to sets of assignments.

(15) a. $[\![P(x)]\!]^w_+ = \lambda g\,.\,\{\,g \mid g(x) \neq \#_e \wedge g(x) \in I(P)(w)\,\}$

b. $[\![P(x)]\!]^w_- = \lambda g\,.\,\{\,g \mid g(x) \neq \#_e \wedge g(x) \notin I(P)(w)\,\}$

A bilateral semantics for atomic sentences

Our new dynamic framework is *bilateral*, which means that we'll recursively define $[\![.]\!]^w_+$ and $[\![.]\!]^w_-$ for all sentences of our fragment.

Like before, meanings are *updates*, i.e., functions from assignments to sets of assignments.

(15) a. $[\![P(x)]\!]^w_+ = \lambda g\,.\,\{\, g \mid g(x) \neq \#_e \wedge g(x) \in I(P)(w)\,\}$
b. $[\![P(x)]\!]^w_- = \lambda g\,.\,\{\, g \mid g(x) \neq \#_e \wedge g(x) \notin I(P)(w)\,\}$

Exercise:

(16) a. $[\![P(x)]\!]^{w_a}_+([x \to a]) =$

A bilateral semantics for atomic sentences

Our new dynamic framework is *bilateral*, which means that we'll recursively define $[\![.]\!]^w_+$ and $[\![.]\!]^w_-$ for all sentences of our fragment.

Like before, meanings are *updates*, i.e., functions from assignments to sets of assignments.

(15) a. $[\![P(x)]\!]^w_+ = \lambda g\,.\,\{\, g \mid g(x) \neq \#_e \wedge g(x) \in I(P)(w)\,\}$
b. $[\![P(x)]\!]^w_- = \lambda g\,.\,\{\, g \mid g(x) \neq \#_e \wedge g(x) \notin I(P)(w)\,\}$

Exercise:

(16) a. $[\![P(x)]\!]^{w_a}_+ ([x \to a]) = \{\, [x \to a]\,\}$

A bilateral semantics for atomic sentences

Our new dynamic framework is *bilateral*, which means that we'll recursively define $[\![.]\!]^w_+$ and $[\![.]\!]^w_-$ for all sentences of our fragment.

Like before, meanings are *updates*, i.e., functions from assignments to sets of assignments.

(15) a. $[\![P(x)]\!]^w_+ = \lambda g\,.\,\{\, g \mid g(x) \neq \#_e \wedge g(x) \in I(P)(w)\,\}$
b. $[\![P(x)]\!]^w_- = \lambda g\,.\,\{\, g \mid g(x) \neq \#_e \wedge g(x) \notin I(P)(w)\,\}$

Exercise:

(16) a. $[\![P(x)]\!]^{w_a}_+([x \to a]) = \{\,[x \to a]\,\}$
b. $[\![P(x)]\!]^{w_a}_-([x \to a]) =$

A bilateral semantics for atomic sentences

Our new dynamic framework is *bilateral*, which means that we'll recursively define $[\![.]\!]^w_+$ and $[\![.]\!]^w_-$ for all sentences of our fragment.

Like before, meanings are *updates*, i.e., functions from assignments to sets of assignments.

(15) a. $[\![P(x)]\!]^w_+ = \lambda g\,.\,\{\, g \mid g(x) \neq \#_e \wedge g(x) \in I(P)(w)\,\}$
b. $[\![P(x)]\!]^w_- = \lambda g\,.\,\{\, g \mid g(x) \neq \#_e \wedge g(x) \notin I(P)(w)\,\}$

Exercise:

(16) a. $[\![P(x)]\!]^{w_a}_+([x \to a]) = \{\,[x \to a]\,\}$
b. $[\![P(x)]\!]^{w_a}_-([x \to a]) = \emptyset$

A bilateral semantics for atomic sentences

Our new dynamic framework is *bilateral*, which means that we'll recursively define $[\![.]\!]^w_+$ and $[\![.]\!]^w_-$ for all sentences of our fragment.

Like before, meanings are *updates*, i.e., functions from assignments to sets of assignments.

(15) a. $[\![P(x)]\!]^w_+ = \lambda g\,.\,\{\, g \mid g(x) \neq \#_e \wedge g(x) \in I(P)(w)\,\}$
b. $[\![P(x)]\!]^w_- = \lambda g\,.\,\{\, g \mid g(x) \neq \#_e \wedge g(x) \notin I(P)(w)\,\}$

Exercise:

(16) a. $[\![P(x)]\!]^{w_a}_+ ([x \to a]) = \{\, [x \to a]\,\}$
b. $[\![P(x)]\!]^{w_a}_- ([x \to a]) = \emptyset$
c. $[\![P(x)]\!]^{w_a}_+ ([x \to b]) =$

A bilateral semantics for atomic sentences

Our new dynamic framework is *bilateral*, which means that we'll recursively define $[\![.]\!]^w_+$ and $[\![.]\!]^w_-$ for all sentences of our fragment.

Like before, meanings are *updates*, i.e., functions from assignments to sets of assignments.

(15) a. $[\![P(x)]\!]^w_+ = \lambda g\,.\,\{\, g \mid g(x) \neq \#_e \wedge g(x) \in I(P)(w)\,\}$
b. $[\![P(x)]\!]^w_- = \lambda g\,.\,\{\, g \mid g(x) \neq \#_e \wedge g(x) \notin I(P)(w)\,\}$

Exercise:

(16) a. $[\![P(x)]\!]^{w_a}_+([x \to a]) = \{\,[x \to a]\,\}$
b. $[\![P(x)]\!]^{w_a}_-([x \to a]) = \emptyset$
c. $[\![P(x)]\!]^{w_a}_+([x \to b]) = \emptyset$

A bilateral semantics for atomic sentences

Our new dynamic framework is *bilateral*, which means that we'll recursively define $[\![.]\!]^w_+$ and $[\![.]\!]^w_-$ for all sentences of our fragment.

Like before, meanings are *updates*, i.e., functions from assignments to sets of assignments.

(15) a. $[\![P(x)]\!]^w_+ = \lambda g\,.\,\{\,g \mid g(x) \neq \#_e \wedge g(x) \in I(P)(w)\,\}$
b. $[\![P(x)]\!]^w_- = \lambda g\,.\,\{\,g \mid g(x) \neq \#_e \wedge g(x) \notin I(P)(w)\,\}$

Exercise:

(16) a. $[\![P(x)]\!]^{w_a}_+([x \to a]) = \{\,[x \to a]\,\}$
b. $[\![P(x)]\!]^{w_a}_-([x \to a]) = \emptyset$
c. $[\![P(x)]\!]^{w_a}_+([x \to b]) = \emptyset$
d. $[\![P(x)]\!]^{w_a}_-([x \to b]) =$

A bilateral semantics for atomic sentences

Our new dynamic framework is *bilateral*, which means that we'll recursively define $[\![.]\!]^w_+$ and $[\![.]\!]^w_-$ for all sentences of our fragment.

Like before, meanings are *updates*, i.e., functions from assignments to sets of assignments.

(15) a. $[\![P(x)]\!]^w_+ = \lambda g\,.\,\{\, g \mid g(x) \neq \#_e \wedge g(x) \in I(P)(w)\,\}$
b. $[\![P(x)]\!]^w_- = \lambda g\,.\,\{\, g \mid g(x) \neq \#_e \wedge g(x) \notin I(P)(w)\,\}$

Exercise:

(16) a. $[\![P(x)]\!]^{w_a}_+([x \to a]) = \{\,[x \to a]\,\}$
b. $[\![P(x)]\!]^{w_a}_-([x \to a]) = \emptyset$
c. $[\![P(x)]\!]^{w_a}_+([x \to b]) = \emptyset$
d. $[\![P(x)]\!]^{w_a}_-([x \to b]) = \{\,[x \to b]\,\}$

A bilateral semantics for atomic sentences

Our new dynamic framework is *bilateral*, which means that we'll recursively define $[\![.]\!]^w_+$ and $[\![.]\!]^w_-$ for all sentences of our fragment.

Like before, meanings are *updates*, i.e., functions from assignments to sets of assignments.

(15) a. $[\![P(x)]\!]^w_+ = \lambda g\,.\,\{\,g \mid g(x) \neq \#_e \wedge g(x) \in I(P)(w)\,\}$
b. $[\![P(x)]\!]^w_- = \lambda g\,.\,\{\,g \mid g(x) \neq \#_e \wedge g(x) \notin I(P)(w)\,\}$

Exercise:

(16) a. $[\![P(x)]\!]^{w_a}_+([x \to a]) = \{\,[x \to a]\,\}$
b. $[\![P(x)]\!]^{w_a}_-([x \to a]) = \emptyset$
c. $[\![P(x)]\!]^{w_a}_+([x \to b]) = \emptyset$
d. $[\![P(x)]\!]^{w_a}_-([x \to b]) = \{\,[x \to b]\,\}$
e. $[\![P(x)]\!]^{w_a}_+([]) =$

A bilateral semantics for atomic sentences

Our new dynamic framework is *bilateral*, which means that we'll recursively define $[\![.]\!]^w_+$ and $[\![.]\!]^w_-$ for all sentences of our fragment.

Like before, meanings are *updates*, i.e., functions from assignments to sets of assignments.

(15) a. $[\![P(x)]\!]^w_+ = \lambda g\,.\,\{\,g \mid g(x) \neq \#_e \wedge g(x) \in I(P)(w)\,\}$
b. $[\![P(x)]\!]^w_- = \lambda g\,.\,\{\,g \mid g(x) \neq \#_e \wedge g(x) \notin I(P)(w)\,\}$

Exercise:

(16) a. $[\![P(x)]\!]^{w_a}_+([x \to a]) = \{\,[x \to a]\,\}$
b. $[\![P(x)]\!]^{w_a}_-([x \to a]) = \emptyset$
c. $[\![P(x)]\!]^{w_a}_+([x \to b]) = \emptyset$
d. $[\![P(x)]\!]^{w_a}_-([x \to b]) = \{\,[x \to b]\,\}$
e. $[\![P(x)]\!]^{w_a}_+([]) = \emptyset$

An aside: why go bilateral?

In a *trivalent* setting, the truth and falsity conditions of a sentence need to be stated separately, i.e.:

(17) *Josie stopped smoking*

a. ...is true if Josie used to smoke and doesn't now.
b. ...is false if Josie used to smoke and still does.
c. ...(unknown otherwise)

The bilateral system outlined here is one lucid way of making dynamic semantics *trivalent* (van den Berg 1996).

Presupposition failure in bilateral DS

In bilateral DS, we'll write $[\![\phi]\!]^{w,g}_{?}$ to indicate what happens when ϕ interpreted at (w, g) is a presupposition failure.

We define this as a derivative notion, guaranteeing that presupposition failures never introduce anaphoric information. Rather, a presupposition failure is a test that succeeds if both the positive output and the negative output are empty (i.e., if the sentence can't be verified or falisified relative to the input)

(18) **Presupposition failure in bilateral DS:**
$[\![\phi]\!]^{w}_{?} = \lambda g \,.\, \{\, g \mid [\![\phi]\!]^{w}_{+}(g) = \emptyset, [\![\phi]\!]^{w}_{-}(g) = \emptyset \,\}$

Presupposition failure in bilateral DS

In bilateral DS, we'll write $[\![\phi]\!]^{w,g}_{?}$ to indicate what happens when ϕ interpreted at (w, g) is a presupposition failure.

We define this as a derivative notion, guaranteeing that presupposition failures never introduce anaphoric information. Rather, a presupposition failure is a test that succeeds if both the positive output and the negative output are empty (i.e., if the sentence can't be verified or falisified relative to the input)

(18) **Presupposition failure in bilateral DS:**
$[\![\phi]\!]^{w}_{?} = \lambda g \,.\, \{\, g \mid [\![\phi]\!]^{w}_{+}(g) = \emptyset, [\![\phi]\!]^{w}_{-}(g) = \emptyset \,\}$

Exercise:

(19) a. $[\![P(x)]\!]^{w_b}_{?}([y \to b]) =$

Presupposition failure in bilateral DS

In bilateral DS, we'll write $[\![\phi]\!]^{w,g}_{?}$ to indicate what happens when ϕ interpreted at (w, g) is a presupposition failure.

We define this as a derivative notion, guaranteeing that presupposition failures never introduce anaphoric information. Rather, a presupposition failure is a test that succeeds if both the positive output and the negative output are empty (i.e., if the sentence can't be verified or falisified relative to the input)

(18) **Presupposition failure in bilateral DS:**
$[\![\phi]\!]^{w}_{?} = \lambda g \,.\, \{\, g \mid [\![\phi]\!]^{w}_{+}(g) = \emptyset, [\![\phi]\!]^{w}_{-}(g) = \emptyset \,\}$

Exercise:

(19) a. $[\![P(x)]\!]^{w_b}_{?}([y \to b]) = \{\, [y \to b] \,\}$

Presupposition failure in bilateral DS

In bilateral DS, we'll write $[\![\phi]\!]^{w,g}_{?}$ to indicate what happens when ϕ interpreted at (w, g) is a presupposition failure.

We define this as a derivative notion, guaranteeing that presupposition failures never introduce anaphoric information. Rather, a presupposition failure is a test that succeeds if both the positive output and the negative output are empty (i.e., if the sentence can't be verified or falisified relative to the input)

(18) **Presupposition failure in bilateral DS:**
$[\![\phi]\!]^{w}_{?} = \lambda g \,.\, \{\, g \mid [\![\phi]\!]^{w}_{+}(g) = \emptyset, [\![\phi]\!]^{w}_{-}(g) = \emptyset \,\}$

Exercise:

(19) a. $[\![P(x)]\!]^{w_b}_{?}([y \to b]) = \{\, [y \to b] \,\}$
b. $[\![P(x)]\!]^{w_b}_{?}([x \to b]) =$

Presupposition failure in bilateral DS

In bilateral DS, we'll write $[\![\phi]\!]^{w,g}_{?}$ to indicate what happens when ϕ interpreted at (w, g) is a presupposition failure.

We define this as a derivative notion, guaranteeing that presupposition failures never introduce anaphoric information. Rather, a presupposition failure is a test that succeeds if both the positive output and the negative output are empty (i.e., if the sentence can't be verified or falisified relative to the input)

(18) **Presupposition failure in bilateral DS:**
$[\![\phi]\!]^{w}_{?} = \lambda g \,.\, \{\, g \mid [\![\phi]\!]^{w}_{+}(g) = \emptyset, [\![\phi]\!]^{w}_{-}(g) = \emptyset \,\}$

Exercise:

(19) a. $[\![P(x)]\!]^{w_b}_{?}([y \to b]) = \{\, [y \to b] \,\}$
b. $[\![P(x)]\!]^{w_b}_{?}([x \to b]) = \emptyset$

Truth and falsity

- A sentence ϕ is *true* at (w, g) if $[\![\phi]\!]^w_+(g) \neq \emptyset$
- A sentence ϕ is *false* at (w, g) if $[\![\phi]\!]^w_+(g) = \emptyset$ and $[\![\phi]\!]^w_-(g) \neq \emptyset$.
- Otherwise, the truth of ϕ at (w, g) is unknown (presupposition failure).

Negation and projection

As is standard in a bilateral setting, we'll define negation as a flip-flop operator:

(20) **Negation in bilateral DS**

a. $[\![\neg\phi]\!]^w_+ = \lambda g\,.\,[\![\phi]\!]^w_-(g)$

b. $[\![\neg\phi]\!]^w_- = \lambda g\,.\,[\![\phi]\!]^w_+(g)$

A couple of interesting things immediately follow from this definition:

- A doubly-negated sentence $\neg\neg\phi$ is equivalent to its positive counterpart ϕ.
  - We'll concretely show how this solves the double-negation problem in a little bit.
- The anaphoric presupposition associated with free variables projects through negation.

Random assignment

Like in classical DS, we'll define *random assignment* in bilateral DS as a privileged tautology.

(21) **Random assignment in bilateral DS**

a. $[\![[x]]\!]^w_+ = \lambda g\,.\,\{\,h \mid g[x]h\,\}$

b. $[\![[x]]\!]^w_- = \lambda g\,.\,\emptyset$

In bilateral DS, the tautological nature of random assignment means that for any possibility (w, g) only ϕ's *positive* output will be non-empty.

The connectives: projection and strong Kleene

- Unlike in classical DS, there is nothing privileged about *conjunction* in Bilateral Update Semantics (BUS).
- To define the connectives, we make use of the *Strong Kleene truth tables*, which represent an algorithm for reasoning about truth/falsity in the presence of uncertainty/a simple theory of presupposition projection.

Presupposition projection and Strong Kleene

$\phi \wedge \psi$	1	0	?
1	1	0	?
0	0	0	0
?	?	0	?

Figure 1: Strong Kleene conjunction

- We'll use the blue region as a recipe for dynamically computing the **positive** update of a conjunctive sentence.
- We'll use the red region as a recipe for dynamically computing the **negative** update of a conjunctive sentence.

Translating the Strong Kleene schema

We translate the Strong Kleene schema into our bilateral schema by taking each cell to be a relational composition of the corresponding updates; presupposition failure stands in for the third truth-value ?.

The only way to verify conjunction is for both conjuncts to be true. That translates straightforwardly into the following positive update:

(22) **Conjunction in bilateral DS (pos.)**
$[\![\phi \wedge \psi]\!]^w_+ = [\![\phi]\!]^w_+ \,;\, [\![\psi]\!]^w_+$

The positive update of conjunction is therefore just like $\phi \,;\, \psi$ in classical DS.

Conjunction: the negative case

The negative output of conjunction is computed by computing *the ways in which a conjunctive sentence is dynamically falsified*.

(23) **Conjunction in bilateral DS (neg.)**

$$\begin{aligned}
[\![\phi \wedge \psi]\!]^w_- = \lambda g\,.\ & [\![\phi]\!]^w_- \,;\, [\![\psi]\!]^w_+ (g) && \text{1st false; 2nd true}\\
\cup\ & [\![\phi]\!]^w_- \,;\, [\![\psi]\!]^w_- (g) && \text{both false}\\
\cup\ & [\![\phi]\!]^w_- \,;\, [\![\psi]\!]^w_? (g) && \text{1st false; 2nd undefined}\\
\cup\ & [\![\phi]\!]^w_+ \,;\, [\![\psi]\!]^w_- (g) && \text{1st true; 2nd false}\\
\cup\ & [\![\phi]\!]^w_? \,;\, [\![\psi]\!]^w_- (g) && \text{1st undefined; 2nd false}
\end{aligned}$$

By the logic of the strong Kleene truth-table, a conjunction can be falsified even if one conjunct is undefined.

We'll go through some more detailed examples a little later on.

Towards existential quantification

We might try to model existential quantification like in classical DS.

(24) $\exists x\phi := [x] \wedge \phi$

This works nicely in the positive case (in fact, it's completely parallel to classical DS).

(25) $$\begin{aligned}[\![[x] \wedge P(x)]\!]^w_+ &= [\![[x]]\!]^w_+ ; [\![P(x)]\!]^w_+ \\ &= \lambda g \,.\, \{\, h \mid g[x]h, h(x) \in I(P)(w) \,\}\end{aligned}$$

A weak form of Egli's theorem falls out in the positive dimension of meaning, thanks to the associativity of (positive) conjunction.

$$[\![[x] \wedge (\phi \wedge \psi)]\!]^w_+ (g) = [\![([x] \wedge \phi) \wedge \psi]\!]^w_+ (g), \forall g$$

A problem

It turns out that looking at the negative case, this does not provide a reasonable semantics for existential quantification.

In fact, we predict that *existential quantification commutes with negation.* This is definitely not desirable.

(26) a. Somebody didn't stay. She was unhappy.
b. Nobody stayed. #She was unhappy.

Let's see why in more detail.

The negative case

Since random assignment is a tautology — only its positive output is non-empty — the only case we need to compute for the negative output of conjunction is the case where conjunction is falsified by the second conjunct.

(27) $$\begin{aligned}[\![[x] \wedge P(x)]\!]^w_- &= [\![[x]]\!]^w_+ ; [\![P(x)]\!]^w_- \\ &= \lambda g\,.\,\{\, h \mid g[x]h, h(x) \notin I(P)(w) \,\}\end{aligned}$$

The negative update of $[x] \wedge P(x)$ introduces a discourse referent x which isn't a P.

Oops! This is exactly the same as the positive output of $[x] \wedge \neg P(x)$.

In order to remedy this, we need to introduce a new operator which prevents existential quantification from automatically taking pseudoscope.

Closure

In order to define existential quantification, we'll introduce a special operator † which leaves the positive output unchanged, but ensures that the negative output is a test.

(28) **Closure**

a. $[\![\dagger\phi]\!]^w_+ = \lambda g \;\; [\![\phi]\!]^w_+(g)$

b. $[\![\dagger\phi]\!]^w_- = \lambda g \,.\, \{\, g \mid [\![\phi]\!]^w_+(g) = \emptyset, [\![\phi]\!]^w_-(g) \neq \emptyset \,\}$

Specifically, it tests whether the prejacent is classically false (note the similarity with ordinary DS negation).

Existential quantification in bilateral DS can be translated as follows:

(29) **Existential quantification in bilateral DS**

$\exists x\phi := \dagger([x] \wedge \phi)$

Existentials under negation

Let's see how closure resolves the interaction between negation and existential quantification:

$$(30) \qquad [\![\dagger([x] \wedge P(x))]\!]^w_- = \lambda g \,.\, \left\{ g \;\middle|\; \begin{array}{l} [\![[x] \wedge P(x)]\!]^w_+ (g) = \emptyset, \\ [\![[x] \wedge P(x)]\!]^w_- (g) \neq \emptyset \end{array} \right\}$$
$$= \lambda g \,.\, \{\, g \mid I(P)(w) = \emptyset \,\}$$

Negated existential statements *test* whether nobody satisfies the exisential statement.

The requirement that the negative output of the prejacent be empty is important for the projection of anaphoric presuppositions...

Projection from existentials

The requirement imposed by closure that the negative output be non-empty ensures that variables that are free within the scope of $\exists$ project anaphoric presuppositions out of $\exists$.

(31) Nobody^x t_x rescued him$_y$.

(32) $[\![\dagger([x] \wedge R(x,y))]\!]^w_-$
$= \lambda g\,.\,\{\, g \mid [\![[x] \wedge R(x,y)]\!]^w_+(g) = \emptyset, [\![[x] \wedge R(x,y)]\!]^w_-(g) \neq \emptyset \,\}$
$= \lambda g\ \ \{\, g \mid g(y) \neq \#_e, \neg\exists x[(x, g(y)) \in I(R)(w)] \,\}$

The resulting update is a *test* on g, which succeeds just in case y is defined, and nobody rescued y.

Double negation illustration

Anaphora from double-negation is still straightforwardly solved, due to the definition of negation as a flip-flop operator:

(33) It's not the case that nobody left

$$
\begin{aligned}
&[\![\neg\neg\dagger([x] \wedge \mathit{Left}(x))]\!]^{w}_{+} \\
&= [\![\neg\dagger([x] \wedge \mathit{Left}(x))]\!]^{w}_{-} \\
&= [\![\dagger([x] \wedge \mathit{Left}(x))]\!]^{w}_{+} \\
&= [\![[x] \wedge \mathit{Left}(x)]\!]^{w}_{+} = \lambda g \,.\, \{\, h \mid g[x]h, h(x) \in I(\mathit{Left})(w) \,\}
\end{aligned}
$$

Disjunctions in bilateral DS

Strong Kleene disjunction

$\phi \vee \psi$	1	0	?
1	1	1	1
0	1	0	?
?	1	?	?

Figure 2: Strong Kleene disjunction

- **True** if either disjunct is true.
- **False** if both disjuncts are false.
- **Undefined** otherwise.

Probably the simplest explanatory theory of presupposition projection in disjunctions (Beaver & Krahmer 2001).

Negated disjunctions

With conjunction, it was simpler to start with the positive case; for disjunction, the negative case is simpler.

(34) **Disjunction in bilateral DS (pos.)**
$[\![\phi \vee \psi]\!]^w_- = [\![\phi]\!]^w_- \,;\, [\![\psi]\!]^w_-$

This is a by-product of de Morgan's equivalences, which are valid in bilateral DS — a negated disjunction is the same as a conjunction of negations.

Bathroom disjunctions: the negative case

Consider the following:

(35) Neither is there no^{x} bathroom, nor is it$_{x}$ upstairs.
$\neg(\neg\exists x[B(x)] \vee U(x))$

Bathroom disjunctions: the negative case

Consider the following:

(35) Neither is there no^x bathroom, nor is it$_x$ upstairs.
$\neg(\neg\exists x[B(x)] \vee U(x))$

Since we just sequence the negative update of each disjunct, we straightforwardly predict this to be equivalent to:

(36) It's not the case that there is no^x bathroom,
and it$_x$'s not upstairs.
$\neg\neg\exists x[B(x)] \wedge \neg U(x)$

Bathroom disjunctions: the negative case

Consider the following:

(35) Neither is there no^x bathroom, nor is it$_x$ upstairs.
$\neg(\neg\exists x[B(x)] \vee U(x))$

Since we just sequence the negative update of each disjunct, we straightforwardly predict this to be equivalent to:

(36) It's not the case that there is no^x bathroom,
and it$_x$'s not upstairs.
$\neg\neg\exists x[B(x)] \wedge \neg U(x)$

Since DNE is valid, this is in turn equivalent to:

(37) There's a bathroom, and it's not upstairs.
$\exists x[B(x)] \wedge \neg U(x)$

Disjunction in bilateral DS

The positive update of disjunction is more complicated, and mirrors the negative update of conjunction (again, due to de Morgan's).

(38) **Disjunction in bilateral DS (pos.)**

$$\begin{aligned}[\![\phi \wedge \psi]\!]^w_+ = \lambda g\,.\; & [\![\phi]\!]^w_+ ; [\![\psi]\!]^w_+ (g) \\ & \cup [\![\phi]\!]^w_+ ; [\![\psi]\!]^w_- (g) \\ & \cup [\![\phi]\!]^w_+ ; [\![\psi]\!]^w_? (g) \\ & \cup [\![\phi]\!]^w_- ; [\![\psi]\!]^w_+ (g) \\ & \cup [\![\phi]\!]^w_? ; [\![\psi]\!]^w_+ (g)\end{aligned}$$

The central intuition: we're gathering up all the different ways in which the disjunction can be dynamically verified.

Bathroom disjunctions: the positive case

It will be useful to break down the positive update associated with a bathroom disjunction into two parts:

- **Verification via the first disjunct:** there's no bathroom.
- **Verification via the second disjunct:** there is a bathroom, and it's upstairs.

(39) Either there's no^{x} bathroom, or it$_{x}$'s upstairs.

There's no bathroom

If there's no bathroom, then the test imposed by the first disjunct succeeds for any g, returning g (thanks to †).

(40) $[\![\neg\exists_x \mathit{Bathroom}(x)]\!]^w_+ = \lambda g\,.\,\{\, g \mid \text{there's no bathroom in } w \,\}$

The second conjunct is atomic, so also a test. If x is defined at g then either the positive or negative test will succeed (depending on whether $g(x)$ is a P). If x is undefined at g, the test for presupposition failure will succeed.

(41) $\lambda g\,.\,[\![\exists_x B(x)]\!]^w_- \,;\, [\![U(x)]\!]^w_+(g) \quad = [\![\exists_x B(x)]\!]^w_-$
$\cup\, [\![\exists_x B(x)]\!]^w_- \,;\, [\![U(x)]\!]^w_-(g)$
$\cup\, [\![\exists_x B(x)]\!]^w_- \,;\, [\![U(x)]\!]^w_?(g)$

There’s no bathroom cont.

That takes care of the first half of the positive update — verification via the first disjunct essentially renders the contribution of the second disjunct inert. We now move onto the case where there is a bathroom.

There is a bathroom

If the first disjunct is *false* (i.e., there is a bathroom), then since DNE is valid, it introduces a bathroom Discourse Referent (DR):

(42) $[\![\neg\exists x[B(x)]]\!]^w_- = [\![\exists x[B(x)]]\!]^w_+ = \lambda g\,.\,\{\,h \mid g[x]h, h(x) \text{ is a bathroom in } w\,\}$

The second disjunct is can now not be a presupposition failure, since x is defined. For the disjunction to be verified, x should be upstairs. This amounts to a simple case of discourse anaphora.

(43) $[\![\exists_x B(x)]\!]^w_+ \,;\, [\![U(x)]\!]^w_+$

Putting the pieces together

Now, the positive update of the entire bathroom disjunction is just the union of (i) the case where there's no bathroom, and (ii) the case where there's a bathroom upstairs:

(44) $[\![\neg\exists x[B(x)] \vee U(x)]\!]^w_+ = \lambda g\,.\,\{\,g \mid \text{there's no bathroom in } w\,\}$
$\cup\,\{\,h \mid g[x]h, h(x) \text{ is an upstairs bathroom in } w\,\}$

We thereby account for anaphora in bathroom disjunctions; the definedness of the sentence does not depend on x being defined at the input.

Weak truth conditions

What truth-conditions do we predict for bathroom disjunctions? If we adopt the simplest conceivable notion of truth in bilateral DS, they should receive *existential truth-conditions*.

(45) **Truth in bilateral DS** (recap)
ϕ is *true* at (w, g) if $[\![\phi]\!]^w_+(g) \neq \emptyset$,
false at (w, g) if $[\![\phi]\!]^w_+(g) = \emptyset$, and $[\![\phi]\!]^w_-(g) \neq \emptyset$.

Predicted truth-conditions:

(46) Either there's no bathroom or it's upstairs.
If there's a bathroom, then there's an upstairs bathroom

This means that the sentence can be true if there's a bathroom upstairs, and a bathroom downstairs.

Asserting disjunctions

External staticity and partial familiarity

You may have noticed that, by dint of how we've defined disjunction, it's capable of introducing referential information; nothing in the semantics of disjunction guarantees that it's a test.

In fact, evaluated relative to a world in which there's an upstairs bathroom *b*, our bathroom disjunction introduces a DR.

(47) $[\![\neg\exists_x B(x) \vee U(x)]\!]^{w_u}_{+}([]) = \{\,[x \to b]\,\}$

Doesn't this erroneously predict that anaphora should be possible in the following?

(48) Either there's no bathroom or it's upstairs. #It's dirty!

No! To appreciate why, we have to step back and consider the effects of asserting a disjunction on the context set.

Asserting disjunctions

Let's construct the kind of file context against which a bathroom disjunction might be felicitously and informatively asserted.

- $w_{\emptyset}$: there's no bathroom.
- $w_{\underline{a}}$: a is a bathroom upstairs, there is no bathroom downstairs.
- $w_{\underline{\underline{a}}}$: a is a bathroom downstairs, there is no bathroom upstairs.

We'll assume that each world is paired with the empty assignment (i.e., no anaphoric information has been introduced).

(49) **Assertion in bilateral DS**

a. In order for assertion of ϕ at c to be defined, it's necessary that $\bigcup_{(w,g)\in c} [\![\phi]\!]^w_? (g) = \emptyset$.

b. If defined, the result of asserting ϕ at c is $\bigcup_{(w,g)\in c} [\![\phi]\!]^w_+ (g)$

Assertion requires that ϕ not be a presupposition failure at any possibility in the context set; this immediately captures Heimian familiarity.

Asserting disjunctions cont.

(50) $[\![\neg\exists_x B(x) \vee U(x)]\!]_+^{w_\emptyset}([]) = \{\,[]\,\}$ *true, no DR introduced*

(51) $[\![\neg\exists_x B(x) \vee U(x)]\!]_+^{w_{\underline{\underline{a}}}}([]) = \{\,[x \to a]\,\}$ *true, DR introduced*

(52) $[\![\neg\exists_x B(x) \vee U(x)]\!]_+^{w_{\overline{\overline{a}}}}([]) = \emptyset$ *not true*

The world in which there is a bathroom but no bathroom is upstairs gets eliminated, since the disjunction is false. A DR is conditionally introduced if there is a bathroom upstairs.

(53) $\left\{ \begin{array}{l} (w_\emptyset, []), \\ (w_{\underline{\underline{a}}}, []), \\ (w_{\overline{\overline{a}}}, []) \end{array} \right\} [\neg\exists_x B(x) \vee U(x)] = \left\{ \begin{array}{l} (w_\emptyset, []) \\ (w_{\underline{\underline{a}}}, [x \to a]) \end{array} \right\}$

Note that in the resuling file context, *x isn't familiar!*

Asserting disjunctions cont.

We may ask, under what conditions would asserting our bathroom disjunction make *x* familiar (relative to an initial context)?

Necessarily, the file context should entail the existence of a bathroom — relative to non-bathroom worlds, the disjunction is true but doesn't introduce a DR.

But in such a context the first disjunct would be contextually trivial (a contextual contradiction).

(54) Context: *it's been established that there's definitely a bathroom in the house, but we don't know where it is.*
#Either there's no bathroom, or its upstairs.

Program disjunctions

Since disjunctions are externally dynamic by default, program disjunctions follow straightforwardly.

Consider the following context against which a program disjunction may be asserted:

- w_{pr}: a, a professor, is in the meeting.
- w_{po}: a, a postdoc, is in the meeting.
- $w_{\emptyset}$: nobody is in the meeting.

Program disjunctions cont.

Now consider the following program disjunction (which licenses subsequent discourse anaphora):

(55) Either a professor is in the meeting, or a postdoc is.
$\exists_x Pr(x) \vee \exists_x Po(x)$

(56) a. $[\![\exists_x Pr(x) \vee \exists_x Po(x)]\!]_+^{w_{pr}}([]) = \{ [x \to a] \}$
b. $[\![\exists_x Pr(x) \vee \exists_x Po(x)]\!]_+^{w_{po}}([]) = \{ [x \to a] \}$
c. $[\![\exists_x Pr(x) \vee \exists_x Po(x)]\!]_+^{w_{\emptyset}}([]) = \emptyset$

All the remaining worlds are onces in which a DR x is introduced, so program disjunctions may make discourse referents fully familiar.

Rothschild's observation

Note that this makes a clear prediction:

- Disjunctions may make DRS *partially familiar*.
- Over the course of a conversation, the context set may evolve such that a previously partially familiar DR becomes fully familiar.

(57) Context: *All renovated houses have bathrooms.*

a. A: Either there's no bathroom or it's upstairs.

b. B: This house has recently been renovated.
It's upstairs on the left.

The witness condition

More generally:

A pronoun indexed *x* is licensed if an indefinite indexed *x* has previously been used, and a witness to the indefinite is contextually entailed.

See Mandelkern 2022, discussion from the previous unit.

Extensions

Incorporating presupposition

Non-anaphoric presuppositions can simply be encoded in bilateral DS in a completely standed way; like anaphoric presuppositions, they are cases where both the positive and negative update can be empty.

(58) a. $[\![\text{Garnet stopped smiling}]\!]^w_+ =$
$\lambda g \,.\, \{\, g \mid \text{Garnet smiled in the past and isn't now in } w \,\}$

b. $[\![\text{Garnet stopped smiling}]\!]^w_- =$
$\lambda g \,.\, \{\, g \mid \text{Garnet smiled in the past and still is in } w \,\}$

This is because trivalent logics can equivalently be stated as bilateral logics.

Recall that asserting a disjunction can make a DR *partially familiar*; this isn't sufficient to license an anaphoric pronoun, which require definedness in *every possibility*.

This raises the question — are they cases where a pronoun can be used when the DR is licensed in only a subset of the possibilities in the context set?

(59) There might be someone^{x} outside. They$_{x}$ might be smoking.

Idea: The modalized existential statement introduces a DR *just in the worlds where the prejacent is (classically) true*, similar to "either someone is outside, or nobody is.". The subsequent statement is licensed just in case *x* is defined in *some* possibilities of the context set (presupposition filtration).

Conclusion

Conceptual issues resolved?

Strong Kleene trivalent semantics is the simplest, empirically viable account of presupposition projection on the market (Beaver & Krahmer 2001).

In BUS, the anaphoric potential associated with complex operators is tied directly to their truth-conditional import.

The result is a theory in which pronouns presuppose the existence of a discourse referent; DR introduction is more classical, and tracks truth-conditions.

This theory however has nothing to say about the interaction between anaphora and modality — we'll turn to this topic in the next unit.

Questions?

- Beaver, David & Emiel Krahmer. 2001. A Partial Account of Presupposition Projection. *Journal of Logic, Language and Information* 10(2). 147–182. https://doi.org/10.1023/A:1008371413822. http://link.springer.com/10.1023/A:1008371413822 (21 July, 2023).
- van den Berg, M. H. 1996. Some Aspects of the Internal Structure of Discourse. The Dynamics of Nominal Anaphora. Publisher: AmsterdamILLC. https://dare.uva.nl/search?arno.record.id=7073 (31 August, 2020).
- Chierchia, Gennaro. 1995. *Dynamics of Meaning - Anaphora, Presupposition, and the Theory of Grammar.* Chicago: University of Chicago Press.
- Evans, Gareth. 1977. Pronouns, Quantifiers, and Relative Clauses (I). *Canadian Journal of Philosophy* 7(3). Publisher: [Taylor & Francis, Ltd., Canadian Journal of Philosophy] JSTOR: 40230703, 467–536.

References ii

- Heim, Irene. 1983. **File Change Semantics and the Familiarity Theory of Definiteness.** In *Meaning, Use, and Interpretation of Language*, 164–189. De Gruyter. https://doi.org/10.1515/9783110852820.164. https://www.degruyter.com/document/doi/10.1515/9783110852820.164/html (27 January, 2022).
- Karttunen, Lauri. 1973. **Presuppositions of Compound Sentences.** *Linguistic Inquiry* 4(2). JSTOR: 4177763, 169–193.
- Mandelkern, Matthew. 2022. **Witnesses.** *Linguistics and Philosophy*. Publisher: Springer, 1–27.
- Roberts, Craige. 1987. ***Modal Subordination, Anaphora and Distributivity.*** University of Massachusetts, Amherst Doctoral dissertation.
- Rothschild, Daniel. 2017. **A Trivalent Account of Anaphora and Presupposition.** In *Proceedings of the 21st amsterdam colloquium*, vol. 21, 1–13. 21st Amsterdam Colloquium.

References iii

- Schlenker, Philippe. 2008. **Be Articulate: A Pragmatic Theory of Presupposition Projection.** *Theoretical Linguistics* 34(3). https://doi.org/10.1515/THLI.2008.013. https://www.degruyter.com/view/j/thli.2008.34.issue-3/thli.2008.013/thli.2008.013.xml (13 September, 2020).
- Schlenker, Philippe. 2010. **Local contexts and local meanings.** *Philosophical Studies: An International Journal for Philosophy in the Analytic Tradition* 151(1). 115–142. https://doi.org/10.2307/40856594.
- van der Sandt, Rob A. 1992. **Presupposition projection as anaphora resolution.** *Journal of Semantics* 9(4). 333–377. https://doi.org/10.1093/jos/9.4.333. https://doi.org/10.1093/jos/9.4.333.

Distinguishing presupposition failure from undefinedness

Presentation as four-valued logic

δ maps false to ?

Δ maps false to #.

Connectives project ? as Strong Kleene, and # as Weak Kleene.

(60) $\varepsilon_x := \lambda gh\,.\,\Delta(g[x]h) \wedge g[x]h$

(61) $P(x) := \lambda gh\,.\,\Delta(g = h) \wedge \delta(h(x) \neq \#_e) \wedge P(h(x))$

(62) $\phi\,;\psi := \lambda gh\,.\,\exists(\lambda i\,.\,\phi(g)(i) \wedge \psi(i)(h))$

(63) $\neg\phi := \lambda gh\,.\,\neg(\phi)(g)(h)$