PLIN0056 Semantics Research Seminar

Lecture 1: Introduction to Question Semantics

Yasutada Sudo 6 October 2023

1 Hamblin's (1958) three postulates

1. *An answer to a question is a statement.*
2. *Knowing what counts as an answer is equivalent to knowing the question.*
3. *The possible answers to a question are an exhaustive set of mutually exclusive possibilities.*

1.1 Postulate 1: *An answer to a question is a statement.*

- An answer to a question is always a statement. When it seems it's not (a 'fragment answer'), it's an elliptical form.

  (1) Who was in the room?
  —John (was in the room).

- *Yes* and *no* are also statements:

  > "yes" and "no" customarily represent statements: we might say that they are statements in code. (Hamblin 1958: p. 162)

  For example:

  (2) Is it raining?
  a. —Yes [= it is raining]
  b. —No [= it is not raining]

  Alternatively, *yes* and *no* mean nothing themselves, but signal that some answer is following them. What follows *yes* is a positive sentence, what follows *no* is a negative sentence. When nothing follows them, it's an elliptical form.

  (3) Is it raining?
  a. —Yes, (it is raining).
  b. —No, (it is not raining).

- **Food for thought**: How would you analyse answers to negative questions?

  (4) Is it not raining?
  a. —Yes.
  b. —No.

  Some languages have a third particle (*doch* in German and Dutch, *si* in French).

1.2 Postulate 2: *Knowing what counts as an answer is equivalent to knowing the question.*

- NB: To know the meaning of a question, you don't need to know the *correct* answer. All you need to know is what is a *possible* answer.

- This postulate allows us to reduce the semantics of questions to the semantics of statements: The meaning of a question = the set of possible answers to it.
- Given Postulate 1, each possible answer is a statement, which we assume denotes a proposition.
- We assume possible world semantics where propositions are functions from possible worlds to truth-values (functions of type $\langle s,t\rangle$).

  (5) $[\![\textbf{Is it raining?}]\!] = \{\, [\lambda w'_s.\ \text{it is raining in } w'],\ [\lambda w'_s.\ \text{it is not raining in } w']\,\}$

- If we focus on total functions, propositions are isomorphic to sets they characterise.

  (6) $[\![\textbf{Is it raining?}]\!] = \{\, \{\, w'_s \mid \text{it is raining in } w'\,\},\ \{\, w'_s \mid \text{it is not raining in } w'\,\}\,\}$

1.3 Postulate 3: *The possible answers to a question are an exhaustive set of mutually exclusive possibilities.*

1.3.1 Exhaustivity and presupposition

- For Hamblin (1958), a question is *logically proper* if its denotation covers the entire set of possible worlds, $\mathcal{W}$:

  (7) $\bigcup [\![\textbf{Is it raining?}]\!] = \mathcal{W}$

- But not all yes/no questions cover the entire set of possible worlds. Such questions have *presuppositions*.

> The necessity for the set of possible answers to be exhaustive is illustrated by the classical "Have you stopped beating your wife?", which is a logically improper question just because the indicated answers "yes" and "no" do not, on the usual reckoning, cover all the logical possibilities. The question "In which continent is Luxembourg?" is like this too, because it presupposes that Luxembourg is in a continent [...] (Hamblin 1958: p. 163)

(8) $[\![\textbf{Have you stopped beating your wife?}]\!]$
$= \left\{ \begin{array}{l} [\lambda w'_s.\ \text{you have stopped beating your wife in } w'], \\ [\lambda w'_s.\ \text{you have not stopped beating your wife in } w'] \end{array} \right\}$

Worlds where you never beat your wife or worlds where you don't have a wife are not in the union of (8).

- More pragmatic version:

> When the indicated answers to a question are not exhaustive one can of course alternatively say that the question is a perfectly proper one *relative to* a certain supposition, namely the supposition expressed by the disjunction of the indicated answers.
> (Hamblin 1958: p. 164)

Stating this in Stalnakerian terms:

– We represent common ground as a proposition = set of possible worlds, representing the mutual beliefs of the conversational participants. This set of possible worlds is called the *context set*.

– A question is proper with respect to context set c if the union of its possible answers covers c.

1.3.2 Mutual exclusiveness and answerhood

- For Hamblin 1958, the possible answers of a question need to be mutually exclusive. This is to capture the notion of *complete answers*:

> Suppose on being asked "In which continent is Luxembourg?" I were to reply "Either Europe, or Asia, or Africa". It might easily be objected that I had not given a proper answer in the sense that I had not given a *complete* answer. This objection might now be put another way: The answer "Either Europe, or Asia, or Africa" cannot be a proper answer, because it does not exclude and is not excluded by other proper answers, e.g. the answer "Europe". *Complete* answers are mutually exclusive, and this is simply one of the things we mean by "completeness". (Hamblin 1958: p. 164)

- Example:

(9) A: In which continent is Luxembourg?
B: Either Europe, or Asia, or Africa.

Under Hamblin's analysis, (9A) denotes:

(10)
$$\left\{ \begin{array}{l} \{\, w'_s \mid \text{Lexembourg is in Europe in } w' \,\}, \\ \{\, w'_s \mid \text{Lexembourg is in Asia in } w' \,\}, \\ \{\, w'_s \mid \text{Lexembourg is in Africa in } w' \,\}, \\ \{\, w'_s \mid \text{Lexembourg is in North America in } w' \,\}, \\ \{\, w'_s \mid \text{Lexembourg is in South America in } w' \,\}, \\ \{\, w'_s \mid \text{Lexembourg is in Australia in } w' \,\}, \\ \{\, w'_s \mid \text{Lexembourg is in Antarctica in } w' \,\} \end{array} \right\}$$

(9B) denotes the union of the first three sets of possible worlds:

$$\{\, w'_s \mid \text{Lexembourg is in Europe, or Asia, or Africa in } w' \,\}$$

This is a *partial answer* to the question (9A). It only settles the question partially.

- For Hamblin, *wh*-questions also denote sets of mutually exclusive answers.

(11) $[\![\textbf{Who left?}]\!]$
$$= \left\{ \begin{array}{l} \{\, w'_s \mid \text{John and nobody else left in } w' \,\}, \\ \{\, w'_s \mid \text{Bill and nobody else left in } w' \,\}, \\ \{\, w'_s \mid \text{Mary and nobody else left in } w' \,\}, \\ \ldots \\ \{\, w'_s \mid \text{John and Bill and nobody else left in } w' \,\}, \\ \{\, w'_s \mid \text{John and Mary and nobody else left in } w' \,\}, \\ \{\, w'_s \mid \text{Bill and Mary and nobody else left in } w' \,\}, \\ \ldots \\ \{\, w'_s \mid \text{John, Bill and Mary and nobody else left in } w' \,\}, \\ \ldots \end{array} \right\}$$

So *John left* is not a complete answer to (11). But one often means 'Only John left' by *John left*, which would be a complete answer, picking out the first set in (11).

- So in Hamblin Semantics, a question induces a partition of the space of (live) possibilities. Each cell of the partition is a complete answer.

> A question is equivalent to a decomposition (or section, or division) of the possible universes. The set of possible universes is split up into a number of subsets, each subset representing an answer to the question, i.e. consisting of exactly those universes consistent with the answer.

> A yes-no question divides the possible universes in two. So, of course, does a statement. But a statement also says which subset contains the actual universe: it polarises the division. A yes-no question merely draws the dividing line, it does not polarise.
>
> (Hamblin 1958: p. 166)

2 Hamblin 1973

2.1 A quick review of intensional semantics

- We translate Hamblin's (1973) system (which builds on a pre-PTQ Montague Grammar) in a more modern framework. Let's first review intensional semantics.
- A model $\mathcal{M} = \langle \mathcal{D}, \mathcal{W}, \mathcal{V} \rangle$. The model parameter $\mathcal{M}$ is henceforth omitted.
- Types and domains

(12)
a. e and t are types.
b. If σ andτ are types, then $\langle \sigma, \tau \rangle$ is a type.
c. If τ is a type, then $\langle s, \tau \rangle$ is a type.
d. Nothing else is a type.

(13)
a. $D_e = \mathcal{D}$
b. $D_t = \{0, 1\}$
c. If $\sigma \neq s$, then $D_{\langle \sigma, \tau \rangle} = D_\tau^{D_\sigma}$ (the set of functions from D_σ to D_τ)
d. $D_{\langle s, \tau \rangle} = D_\tau^{\mathcal{W}}$ (the set of functions from $\mathcal{W}$ to D_τ)

- Example denotations

(14)
a. $[\![\textbf{John}]\!]^g = [\lambda w_s.\ j]$
b. $[\![\textbf{walks}]\!]^g = [\lambda w_s . \lambda i_{\langle s,e \rangle}.\ i(w)$ walks in $w]$
c. $[\![\textbf{saw}]\!]^g = [\lambda w_s.\ \lambda i_{\langle s,e \rangle}.\ \lambda j_{\langle s,e \rangle}.\ j(w)$ saw $i(w)$ in $w]$

- Compositional rules

(15) *Functional Application*
If $[\![\alpha]\!]^g \in D_{\langle s, \langle \langle s,\sigma \rangle, \tau \rangle \rangle}$ and $[\![\beta]\!]^g \in D_{\langle s,\sigma \rangle}$, then $[\![\alpha\ \beta]\!]^g = [\![\beta\ \alpha]\!]^g = [\lambda w_s.\ [\![\alpha]\!]^g(w)([\![\beta]\!]^g)]$.

(16) *Predicate Abstraction*
$[\![\lambda n\ \textbf{XP}]\!]^g = [\lambda w_s . \lambda i_{\langle s,e \rangle}.\ [\![\textbf{XP}]\!]^{g[n \mapsto i(w)]}(w)]$

(17) *Trace rule*
$[\![t_n]\!]^g = \lambda w_s.\ g(n)$

- Remarks: Heim & Kratzer 1998 and von Fintel & Heim 2021 postulate two Functional Application rules, Intensional and Extensional Functional Application. We dispense with the latter by achieving the same thing in the lexical denotations of extensional operators. Similarly, our formulation of Predicate Abstraction is slightly different from theirs, but does the same thing. The two systems are equivalent.

(18) *Extensional Functional Application*
If $[\![\alpha]\!]^g \in D_{\langle s, \langle \sigma, \tau \rangle \rangle}$ and $[\![\beta]\!]^g \in D_{\langle s,\sigma \rangle}$, then $[\![\alpha\ \beta]\!]^g = [\![\beta\ \alpha]\!]^g = [\lambda w_s.\ [\![\alpha]\!]^g(w)([\![\beta]\!]^g(w))]$.

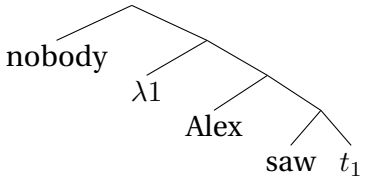
(19)
a. $[\![\textbf{walks}]\!]^g = [\lambda w_s . \lambda x_e.\ x$ walks in $w]$
b. $[\![\textbf{saw}]\!]^g = [\lambda w_s . \lambda x_e . \lambda y_e.\ y$ saw x in $w]$

In addition, they 'hide' λw by putting w as an index on $[\![\cdot]\!]$.

Exercise: Reformulate the two theories of question semantics discussed below in a system with Extensional Functional Application.

- Quantifiers covertly move (Quantifier Raising) and take scope.

(20)



(21) $[\![\textbf{nobody}]\!]^g$
$= [\lambda w_s.\ \lambda f_{\langle s,\langle\langle s,e\rangle,t\rangle\rangle}.\ \{\, x \in D_e \mid x \text{ is a person in } w \text{ and } f(w)(\lambda w'_s.\ x) = 1 \,\} = \varnothing]$

(22) $[\![\lambda 1\ \textbf{Alex saw}\ t_1]\!]^g = [\lambda w_s.\lambda i_{\langle s,e\rangle}.\ \text{Alex saw } i(w) \text{ in } w]$

2.2 Hamblin Semantics

- Denotations are sets of intensions.

(23) a. $[\![\textbf{Katie}]\!]^g_H = \{\, \lambda w_s.\ k \,\}$
b. $[\![\textbf{disappared}]\!]^g_H = \{\, \lambda w_s.\lambda i_{\langle s,e\rangle}.\ i(w) \text{ disappeared in } w \,\}$
c. $[\![\textbf{saw}]\!]^g_H = \{\, \lambda w_s.\lambda i_{\langle s,e\rangle}.\lambda j_{\langle s,e\rangle}.\ i(w) \text{ saw } j(w) \text{ in } w \,\}$

- Types and Domains

(24) a. e and t are types.
b. If σ andτ are types, then $\langle \sigma, \tau \rangle$ is a type.
c. If τ is a type, then $\langle s, \tau \rangle$ is a type.
d. If τ is a type, then $\hat{\tau}$ is a type.
e. Nothing else is a type.

(25) a. $D_e = \mathcal{D}$
b. $D_t = \{\, 0, 1 \,\}$
c. If $\sigma \neq s$, then $D_{\langle \sigma,\tau \rangle} = D_\tau^{D_\sigma}$
d. $D_{\langle s,\tau \rangle} = D_\tau^{\mathcal{W}}$
e. $D_{\hat{\tau}} = \wp(D_\tau)$

- Denotations combine via point-wise function application (which Hamblin calls "◦").

(26) *Hamblin Functional Application*
If $[\![\alpha]\!]^g_H \subseteq D_{\widehat{\langle s,\langle\langle s,\sigma\rangle,\tau\rangle\rangle}}$ and $[\![\beta]\!]^g_H \subseteq D_{\widehat{\langle s,\sigma\rangle}}$,
then $[\![\alpha\ \beta]\!]^g_H = [\![\beta\ \alpha]\!]^g_H = \{\, \lambda w_s.\ a(w)(b) \mid a \in [\![\alpha]\!]^g_H \text{ and } b \in [\![\beta]\!]^g_H \,\}$.

(27) $[\![\textbf{Katie disappeared}]\!]^g_H$
$= \left\{ \lambda w_s.\ a(w)(b) \;\middle|\; \begin{array}{l} a \in \{\, \lambda w_s.\lambda i_{\langle s,e\rangle}.\ i(w) \text{ disappeared in } w \,\} \\ \text{and } b \in \{\, \lambda w_s.\ k \,\} \end{array} \right\}$
$= \{\, \lambda w_s.\ k \text{ disappeared in } w \,\}$

- *Wh*-phrases denote a set of alternatives. Let's not worry about the sortal restriction for now.

(28) a. $[\![\textbf{what}]\!]^g_H = \{\, \lambda w_s.\ x \mid x \in D_e \,\}$
b. $[\![\textbf{What disappeared?}]\!]^g_H$
$= \left\{ \lambda w_s.\ a(w)(b) \;\middle|\; \begin{array}{l} a \in \{\, \lambda w_s.\lambda i_{\langle s,e\rangle}.\ i(w) \text{ disappeared in } w \,\} \\ \text{and } b \in \{\, \lambda w_s.\ x \mid x \in D_e \,\} \end{array} \right\}$
$= \{\, \lambda w_s.\ x \text{ disappeared in } w \mid x \in D_e \,\}$
$= \left\{ \begin{array}{l} \lambda w_s.\ e_1 \text{ disappeared in } w, \\ \lambda w_s.\ e_2 \text{ disappeared in } w, \\ \lambda w_s.\ e_3 \text{ disappeared in } w, \\ \ldots \end{array} \right\}$

- Notice that the set in (28b) is not a partition, unlike in Hamblin 1958.

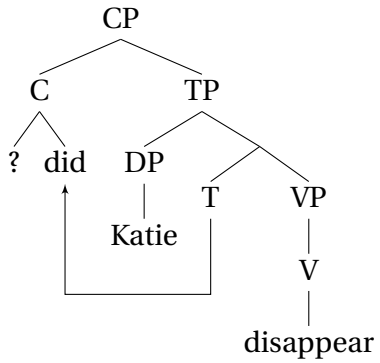
> We shall need to regard 'who walks' as itself denoting a set, namely, the set whose members are propositions denoted by 'Mary walks', 'John walks', ... and so on for all individuals. Pragmatically speaking a question sets up a choice-situation between a set of propositions, namely, those propositions that count as answers to it. (Hamblin 1973: p. 48)

- Polar questions can be accounted for by the operator in (29):

(29) $[\![\textbf{?}]\!]^g_H = \{\, \lambda w_s.\ \lambda p_{\langle s,t\rangle}.\ p(w) = 1,\ \lambda w_s.\ \lambda p_{\langle s,t\rangle}.\ p(w) = 0 \,\}$

(30) a. $[\![\textbf{Katie disappeared}]\!]^g_H = \{\, \lambda w_s.\ k \text{ disappeared in } w \,\}$

b. $[\![\textbf{? Katie disappeared}]\!]^g_H = \left\{ \begin{array}{l} \lambda w_s.\ k \text{ disappeared in } w, \\ \lambda w_s.\ k \text{ didn't disappeared in } w \end{array} \right\}$



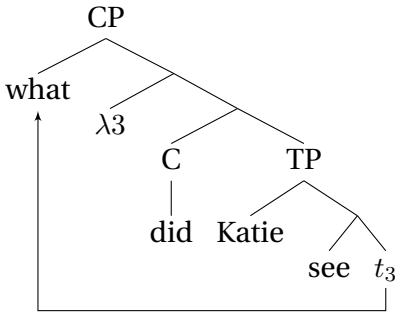
- Singleton sets of propositions are statements; Non-singleton (non-empty) sets of propositions are questions.

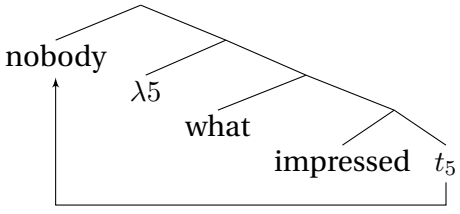
2.3 Hamblin Predicate Abstraction

- Hamblin Semantics (and Alternative Semantics for Focus) has a problem with abstraction (Shan 2004, Romero 2010, Charlow 2014).[1]
- We want to account for sentences involving movement.

(31) a. What did Katie see?

b. What impressed nobody?





Recall:

(32) $[\![\textbf{what}]\!]^g_H = \{\, \lambda w_s.\ x \mid x \in D_e \,\}$

In order to derive the correct meaning, we want the sister of *what* in (31a) to denote (33).

(33) $\{\, \lambda w_s.\lambda i_{\langle s,e\rangle}.\ k \text{ saw } i(w) \text{ in } w \,\}$

Applying Hamblin Functional Application to (32) and (33), we'll get:

[1] Hamblin 1973 original formulation also has an analogous problem with quantifying-in.

(34) $\{\, \lambda w_s.\ k \text{ saw } x \text{ in } w \mid x \in D_e \,\}$

- How do we derive (33) compositionally?
  - (35) *Trace rule*
    $[\![t_n]\!]^g_H = \{\, \lambda w_s.\ g(n) \,\}$.
  - The sister constituent to the index node is assignment dependent.
    (36) a. $[\![\textbf{Katie saw } t_3]\!]^g_H = \{\, \lambda w_s.\ k \text{ saw } g(3) \text{ in } w \,\}$
    b. $[\![\textbf{what impressed } t_5]\!]^g_H = \{\, \lambda w_s.\ x \text{ impressed } g(5) \mid x \in D_e \,\}$
  - The following formulation is obviously wrong:
    (37) *Hamblin Predicate Abstraction* (ver. 1)
    $[\![\lambda n\ \textbf{XP}]\!]^g_H = \lambda w_s.\lambda i_{s,e}.\ \{\, f(w) \mid f \in [\![\textbf{XP}]\!]^{g[n\mapsto i(w)]}_H \,\}$
    (38) $[\![\lambda 3\ \textbf{Katie saw } t_3]\!]^g_H$
    $= \lambda w_s.\lambda i_{\langle s,e\rangle}.\ \{\, f(w) \mid f \in [\![\textbf{Katie saw } t_3]\!]^{g[3\mapsto i(w)]}_H \,\}$
    $= \lambda w_s.\lambda i_{\langle s,e\rangle}.\ \begin{cases} \{\,1\,\} & \text{if } k \text{ saw } i(w) \text{ in } w \\ \{\,0\,\} & \text{otherwise} \end{cases}$
  - The following formulation doesn't work either.
    (39) *Hamblin Predicate Abstraction* (ver. 2)
    $[\![\lambda n\ \textbf{XP}]\!]^g_H = \{\, \lambda w_s.\lambda i_{\langle s,e\rangle}.\ [\![\textbf{XP}]\!]^{g[n\mapsto i(w)]}_H \,\}$
    (40) $[\![\lambda 3\ \textbf{Katie saw } t_3]\!]^g_H$
    $= \{\, \lambda w_s.\lambda i_{\langle s,e\rangle}.\ [\![\textbf{Katie saw } t_3]\!]^{g[3\mapsto i(w)]}_H \,\}$
    $= \{\, \lambda w_s.\lambda i_{\langle s,e\rangle}.\ \{\, \lambda w'_s.\ k \text{ saw } g(3) \text{ in } w' \,\} \,\}$
    This is wrong in many ways.
  - Note that the following is simply ill-formed (why?):
    (41) *Hamblin Predicate Abstraction* (ver. 3)
    $[\![\lambda n\ \textbf{XP}]\!]^g_H = \{\, \lambda w_s.\lambda i_{\langle s,e\rangle}.\ f \mid f \in [\![\textbf{XP}]\!]^{g[n\mapsto i(w)]}_H \,\}$
  - Kratzer & Shimoyama (2002) propose something along the lines of (42):
    (42) *Hamblin Predicate Abstraction* (ver. 4)
    $[\![\lambda n\ \textbf{XP}]\!]^g_H = \{\, \lambda w_s.\lambda i_{\langle s,e\rangle}.f(w)(i(w)) \mid \forall x \in D_e[[\lambda w_s.\ f(w)(x)] \in [\![\textbf{XP}]\!]^{g[n\mapsto x]}_H] \,\}$
    (43) $[\![\lambda 3\ \textbf{Katie saw } t_3]\!]^g_H$
    $= \{\, \lambda w_s.\lambda i_{\langle s,e\rangle}.f(w)(i(w)) \mid \forall x \in D_e[\lambda w_s.\ f(w)(x) \in [\![\textbf{Katie saw } t_3]\!]^{g[3\mapsto x]}_H] \,\}$
    $= \{\, \lambda w_s.\lambda i_{\langle s,e\rangle}.f(w)(i(w)) \mid \forall x \in D_e[\lambda w_s.\ f(w)(x) \in \{\, \lambda w_s.\ k \text{ saw } x \text{ in } w \,\}] \,\}$
    $= \{\, \lambda w_s.\lambda i_{\langle s,e\rangle}.f(w)(i(w)) \mid \forall x \in D_e[\lambda w_s.\ f(w)(x) = \lambda w_s.\ k \text{ saw } x \text{ in } w] \,\}$
    So far so good, but as Shan (2004) points out, a problem arises in certain cases.
    (44) $[\![\lambda 5\ \textbf{what impressed } t_5]\!]^g_H$
    $= \{\, \lambda w_s.\lambda i_{\langle s,e\rangle}.\ f(w)(i(w)) \mid \forall x \in D_e[\lambda w_s.\ f(w)(x) \in [\![\textbf{what impressed } t_5]\!]^{g[5\mapsto x]}_H] \,\}$
    $= \{\, \lambda w_s.\lambda i_{\langle s,e\rangle}.\ f(w)(i(w)) \mid \forall x \in D_e[\lambda w_s.\ f(w)(x) \in \{\, \lambda w_s.\ y \text{ impressed } x \text{ in } w \mid y \in D_e \,\}] \,\}$
    This set includes the following g, among many other functions.

$$g(w)(\lambda w'_s.\ a) = 1 \Leftrightarrow d \text{ impressed } a \text{ in } w$$
$$g(w)(\lambda w'_s.\ b) = 1 \Leftrightarrow e \text{ impressed } b \text{ in } w$$
$$g(w)(\lambda w'_s.\ c) = 1 \Leftrightarrow f \text{ impressed } c \text{ in } w$$
$$\vdots$$

    Assuming the following denotation for *nobody*:
    (45) $[\![\textbf{nobody}]\!]^g_H$
    $= \{\, \lambda w_s.\ \lambda f_{\langle s,\langle\langle s,e\rangle,t\rangle\rangle}.\ \{\, x \in D_e \mid x \text{ is a person in } w \text{ and } f(w)(\lambda w'_s.\ x) = 1 \,\} = \varnothing \,\}$

the denotation of (31b) will be:

(46) $[\![\textbf{nobody}\ \lambda 5\ \textbf{what impressed}\ t_5]\!]^g_H$

$$= \left\{ \lambda w_s.\ \left\{ x \in D_e \;\middle|\; \begin{array}{l} x \text{ is a person in } w \\ \text{and } f(w)(x) = 1 \end{array} \right\} = \varnothing \;\middle|\; \forall x \in D_e \left[\begin{array}{l} \lambda w_s.\ f(w)(x) \in \\ \left\{ \begin{array}{l} \lambda w_s.\ y \text{ impressed} \\ x \text{ in } w \end{array} \;\middle|\; y \in D_e \right\} \end{array} \right] \right\}$$

Then it is predicted that the following will be a possible answer.

(47) Q: What impressed nobody?
A: #David's latest book didn't impress Alex, Emily's car didn't impress Becky, France didn't impress Chris, etc.

- Assuming that Predicate Abstraction cannot be formulated in Hamblin Semantics (and Alternative Semantics for focus), some have proposed to make use of this problem to explain certain linguistic phenomena. E.g., Kotek 2018 proposes to explain so-called intervention effects in terms of illicit Predicate Abstraction.

- But this is just a technical problem and it can in fact be solved. The trick is to 'intensionalise more', including the assignment function.

(48)
a. e and t are types.
b. If σ andτ are types, then $\langle \sigma, \tau \rangle$ is a type.
c. If τ is a type, then $\langle s, \tau \rangle$ and $\langle g, \tau \rangle$ are types.
d. If τ is a type, then τ is a type.
e. Nothing else is a type.

(49)
a. $D_e = \mathcal{D}$
b. $D_t = \{0, 1\}$
c. If $\sigma \notin \{s, g\}$, then $D_{\langle \sigma,\tau \rangle} = D_\tau^{D_\sigma}$
d. $D_{\langle s,\tau \rangle} = D_\tau^{\mathcal{W}}$
e. $D_{\hat{\tau}} = \wp(D_\tau)$
f. $D_{\langle g,\tau \rangle} = D_\tau^{\mathbb{N}}$

(50)
a. $[\![\textbf{Katie}]\!]_H = \{ \lambda g_g.\lambda w_s.\ k \}$
b. $[\![\textbf{disappared}]\!]_H = \{ \lambda g_g.\lambda w_s.\lambda i_{\langle g,\langle s,e \rangle\rangle}.\ i(g)(w) \text{ disappeared in } w \}$
c. $[\![\textbf{saw}]\!]_H = \{ \lambda g_g.\lambda w_s.\lambda i_{\langle g,\langle s,e \rangle\rangle}.\lambda j_{\langle g,\langle s,e \rangle\rangle}.\ i(g)(w) \text{ saw } j(g)(w) \text{ in } w \}$
d. $[\![\textbf{nobody}]\!]_H$
$= \left\{ \lambda g_g.\lambda w_s.\ \lambda f_{\langle g,\langle s,\langle\langle g,\langle s,e \rangle\rangle,t\rangle\rangle\rangle}.\ \left\{ x \in D_e \;\middle|\; \begin{array}{l} x \text{ is a person in } w \text{ and} \\ f(g)(w)(\lambda h_g.\lambda w'_s.\ x) = 1 \end{array} \right\} = \varnothing \right\}$
e. $[\![\textbf{what}]\!]_H = \{ \lambda g_g.\lambda w_s.\ x \mid x \in D_e \}$

(51) *Hamblin Functional Application*
If $[\![\alpha]\!]_H \subseteq D_{\langle g,\widehat{\langle s,\langle\langle s,\sigma\rangle,\tau\rangle\rangle}\rangle}$ and $[\![\beta]\!]_H \subseteq D_{\langle g,\widehat{\langle s,\sigma\rangle}\rangle}$,
then $[\![\alpha\ \beta]\!]_H = [\![\beta\ \alpha]\!]_H = \{ \lambda g_g.\lambda w_s.\ a(g)(w)(b) \mid a \in [\![\alpha]\!]_H \text{ and } b \in [\![\beta]\!]_H \}$.

(52) *Hamblin Predicate Abstraction*
$[\![\lambda n\ \textbf{XP}]\!]_H = \{ \lambda g_g.\lambda w_s.\lambda i_{\langle g,\langle s,e \rangle\rangle}.f(g[n \mapsto i(g)(w)])(w) \mid f \in [\![\textbf{XP}]\!]_H \}$

(53) *Trace rule*
$[\![t_n]\!]_H = \{ \lambda g_g.\lambda w_s.\ g(n) \}$.

(54) $[\![\lambda 5\ \textbf{what impressed}\ t_5]\!]_H$
$= \{ \lambda g_g.\lambda w_s.\lambda i_{\langle g,\langle s,e \rangle\rangle}.\ f(g[5 \mapsto i(g)(w)])(w) \mid f \in [\![\textbf{what impressed}\ t_5]\!]_H \}$
$= \{ \lambda g_g.\lambda w_s.\lambda i_{\langle g,\langle s,e \rangle\rangle}.\ f(g[5 \mapsto i(g)(w)])(w) \mid f \in \{ \lambda g_g.\lambda w_s.\ x \text{ impressed } g(5) \text{ in } w \mid x \in D_e \} \}$
$= \{ \lambda g_g.\lambda w_s.\lambda i_{\langle g,\langle s,e \rangle\rangle}.\ x \text{ impressed } i(g)(w) \text{ in } w \mid D_e \}$

(55) $[\![\textbf{nobody}\ \lambda 5\ \textbf{what impressed}\ t_5]\!]_H$
$= \left\{ \lambda g_g.\lambda w_s.\ \left\{ y \in D_e \;\middle|\; \begin{array}{l} y \text{ is a person in } w \text{ and} \\ x \text{ impressed } y \text{ in } w \end{array} \right\} = \varnothing \;\middle|\; x \in D_e \right\}$

- If we were to pursue approaches like Kotek's, we would have to assume that the above rule of Predicate Abstraction is somehow unavailable.

2.4 Sortal restrictions

How do we encode sortal restrictions?

(56) $[\![\textbf{what}]\!]^g_H = \{\, \lambda w_s.\, x \mid x \in D_e \,\}$

The following don't work (why?):

(57) a. $[\![\textbf{who}]\!]^g_H = \{\, \lambda w_s.\, x$ is a person in w and $x \mid x \in D_e \,\}$
b. $[\![\textbf{who}]\!]^g_H = \{\, \lambda w_s.\, x \mid x \in D_e$ and x is a person in $w \,\}$

One analytical possibility is to encode it as a presupposition:

(58) $[\![\textbf{who}]\!]^g_H = \{\, \lambda w_s\colon\ x$ is a person in $w.\ x \mid x \in D_e \,\}$

We'll come back to presuppositions in questions later.

3 Karttunen 1977

Karttunen 1977 proposes a more conservative extension of Montague Grammar (he uses PTQ). We'll translate it in a more modern system.

3.1 Declarative sentences

- We don't need to abstract over g anymore, so we'll treat it as an index. But remember that formally it's the same thing.
- Non-interrogative part of the semantics is as usual.

(59) a. $[\![\textbf{Katie}]\!]^g_K = \lambda w_s.\, k$
b. $[\![\textbf{swims}]\!]^g_K = \lambda w_s.\lambda i_{\langle s,e\rangle}.\, i(w)$ swims in w

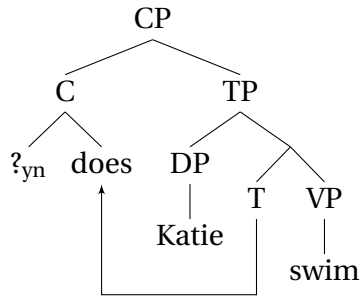
- We can use the usual Functional Application rule.

(60) *Functional Application*
If $[\![\alpha]\!]^g_K \in D_{\langle s,\langle\langle s,\sigma\rangle,\tau\rangle\rangle}$ and $[\![\beta]\!]^g_K \in D_{\langle s,\sigma\rangle}$, then $[\![\alpha\ \beta]\!]^g_K = [\![\beta\ \alpha]\!]^g_K = [\lambda w_s.\, [\![\alpha]\!]^g_K(w)([\![\beta]\!]^g_K)]$.

3.2 Polar Questions

A yes/no question is formed out of a declarative sentence by the polar question operator.

(61) $[\![\textbf{?}_{\textbf{yn}}]\!]^g_K = \lambda w_s.\lambda p_{\langle s,t\rangle}.\lambda q_{\langle s,t\rangle}.\, p = q$ or $q = \lambda w'_s.\, p(w') = 0$



(62) $[\![\textbf{Katie swims}]\!]^g_K = \lambda w_s.\ k$ swims in w

(63) $[\![\textbf{?}_{\textbf{yn}}\ \textbf{Katie swims}]\!]^g_K = \lambda w_s.\lambda q_{\langle s,t\rangle}.\ [q = \lambda w_s.\ k$ swims in $w]$ or $[q = \lambda w_s.\ k$ doesn't swim in $w]$

For any w, $[\![\textbf{?}_{\textbf{yn}}\ \textbf{Katie swims}]\!]^g_K(w)$ characterises:

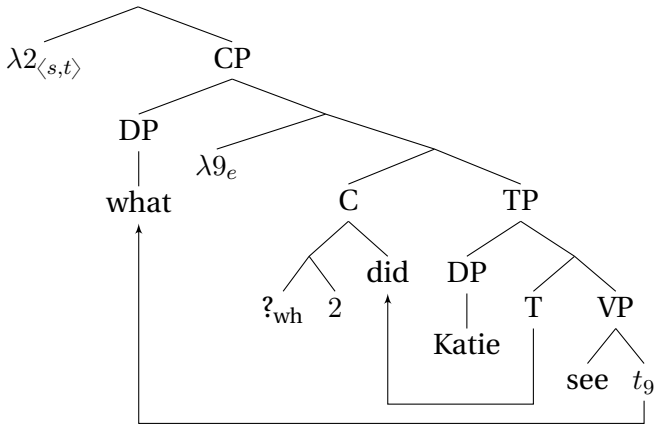
$$\left\{ \begin{array}{l} \lambda w'_s.\ k \text{ swims in } w', \\ \lambda w'_s.\ k \text{ doesn't swim in } w' \end{array} \right\}$$

3.3 *Wh*-Questions

A *wh*-question is formed by another question operator:

(64) $[\![\textbf{?}_{\textbf{wh}}]\!]^g_K = \lambda w_s.\lambda p_{\langle s,t\rangle}.\lambda q_{\langle s,t\rangle}.\ p = q$

(65) What did Katie see?



It's crucial that there's abstraction over propositions taking scope over *what*. To enable this, we will postulate another Predicate Abstraction rule.

(66) *Extensional Predicate Abstraction (type e)*
$[\![\lambda n_e\ \textbf{XP}]\!]^g_K = \lambda w_s.\lambda i_{\langle s,e\rangle}.\ [\![\textbf{XP}]\!]^{g[n\mapsto i(w)]}_K(w)$

(67) *Trace rule*
$[\![t_n]\!]^g_K = \lambda w_s.\ g(n)$

(68) *Intensional Predicate Abstraction (type $\langle s,t\rangle$)*
$[\![\lambda n_{\langle s,t\rangle}\ \textbf{XP}]\!]^g_K = \lambda w_s.\lambda p_{\langle s,t\rangle}.\ [\![\textbf{XP}]\!]^{g[n\mapsto p]}(w)$

(69) *Variable rule*
$[\![n]\!]^g_K = g(n)$

(70) a. $[\![\textbf{?}_{\textbf{wh}}\ 2]\!]^g_K = \lambda w_s.\ \lambda q_{\langle s,t\rangle}.\ g(2) = q$
b. $[\![\textbf{Katie saw}\ t_9]\!]^g_K = \lambda w_s.\ k \text{ saw } g(9) \text{ in } w$
c. $[\![\textbf{[?}_{\textbf{wh}}\ 2\textbf{]\ Katie saw}\ t_9]\!]^g_K = \lambda w_s.\ g(2) = [\lambda w'_s.\ k \text{ saw } g(9) \text{ in } w']$

Wh-phrases are existential quantifiers.

(71) $[\![\textbf{what}]\!]^g_K = [\lambda w_s.\lambda P_{\langle s,\langle\langle s,e\rangle,t\rangle\rangle}.\ \exists x_e[P(w)(\lambda w'_s.\ x) = 1]]$

(72) a. $[\![\lambda 9_e\ \textbf{[?}_{\textbf{wh}}\ 2\textbf{]\ Katie saw}\ t_9]\!]^g_K = \lambda w_s.\lambda i_{\langle s,e\rangle}.\ g(2) = [\lambda w'_s.\ k \text{ saw } g(i(w)) \text{ in } w']$
b. $[\![\textbf{what}\ \lambda 9_e\ \textbf{[?}_{\textbf{wh}}\ 2\textbf{]\ Katie saw}\ t_9]\!]^g_K = \lambda w_s.\ \exists x \in D_e[g(2) = [\lambda w'_s.\ k \text{ saw } x \text{ in } w']]$
c. $[\![\lambda 2_{\langle s,t\rangle}\ \textbf{what}\ \lambda 9_e\ \textbf{[?}_{\textbf{wh}}\ 2\textbf{]\ Katie saw}\ t_9]\!]^g_K = \lambda w_s.\lambda p_{\langle s,t\rangle}.\ \exists x \in D_e[p = [\lambda w'_s.\ k \text{ saw } x \text{ in } w']]$

For any w, $[\![\lambda 2_{\langle s,t\rangle}\ \textbf{what}\ \lambda 9_e\ \textbf{[?}_{\textbf{wh}}\ 2\textbf{]\ Katie saw}\ t_9]\!]^g_K(w)$ characterises

$$\{\ \lambda w'_s.\ k \text{ saw } x \text{ in } w' \mid x \in D_e\ \}$$

- We can encode sortal restrictions straightforwardly.

  (73) $[\![\textbf{who}]\!]^g_K = [\lambda w_s.\lambda P_{\langle s,\langle\langle s,e\rangle,t\rangle\rangle}.\ \exists x_e[x \text{ is a person in } w \text{ and } P(w)(\lambda w'_s.\ x) = 1]]$

  For any w, $[\![\lambda 2_{\langle s,t\rangle}\ \textbf{who}\ \lambda 9_e\ \textbf{[?}_{\textbf{wh}}\ 2\textbf{]\ Katie saw}\ t_9]\!]^g_K(w)$ characterises

  $$\{\ \lambda w'_s.\ k \text{ saw } x \text{ in } w \mid x \in D_e \text{ and } x \text{ is a person in } w\ \}$$

- In order to prevent indefinites like *somebody* to function like a *wh*-phrase, we need to assume that indefinites cannot take scope between $?_{\text{wh}}$ and the propositional abstractor in the CP area for syntactic reasons.

References

Charlow, Simon. 2014. *On the semantics of exceptional scope*. New York University dissertation.

von Fintel, Kai & Irene Heim. 2021. Intensional semantics. Lecture notes, Massachusetts Institute of Technology.

Hamblin, Charles Leonard. 1958. Questions. *Australasian Journal of Philosophy* 36. 159–168.

Hamblin, Charles Leonard. 1973. Questions in Montague Grammar. *Foundations of Language* 10. 41–53.

Heim, Irene & Angelika Kratzer. 1998. *Semantics in Generative Grammar*. Oxford: Blackwell.

Karttunen, Lauri. 1977. Syntax and semantics of questions. *Linguistics and Philosophy* 1. 3–44.

Kotek, Hadas. 2018. *Composing questions*. Cambridge, MA: MIT Press.

Kratzer, Angelika & Junko Shimoyama. 2002. Indeterminate pronouns: The view from Japanese. In Yukio Otsu (ed.), *Proceedings of the 3rd Tokyo Conference on Psycholinguistics*, 1–25. Tokyo: Hituzi Syobo.

Romero, Maribel. 2010. Alternative-based semantics combined with movement: the role of presupposition. Handout of the talk given at the Workshop on Alternative-Based Semantics.

Shan, Chung-chieh. 2004. Binding alongside Hamblin alternatives calls for variable-free semantics. In *Proceedings of SALT 14*, 289–304.