Hamblin Semantics & Focus

Last week

I. Generalized quantifiers

- quantificational DPs are generalized quantifiers, which denote sets of sets (of type \( \langle et, t \rangle \)); quantificational determiners denote relation between sets (of type \( \langle et, ett \rangle \)).

\[
\begin{align*}
(1) & \quad \text{a. } [\text{every cat}] = \lambda P_{\langle et, t \rangle} x \forall x [\text{cat}'(x) \rightarrow P(x)] \\
& \quad \text{b. } [\text{some cat}] = \lambda P_{\langle et, t \rangle} x \exists x [\text{cat}'(x) \wedge P(x)] \\
(2) & \quad \text{a. } [\text{every}] = \lambda Q_{\langle et, t \rangle} x \forall x [Q(x) \rightarrow P(x)] \\
& \quad \text{b. } [\text{some}] = \lambda Q_{\langle et, t \rangle} x P_{\langle et, t \rangle} \exists x [Q(x) \wedge P(x)]
\end{align*}
\]

- Some shifting operations: (i) individuals (of type \( e \)) can also be shifted into generalized quantifiers via type-lifting; (ii) the quantification domain of an \( D \)-quantifier can be extracted via the BE-shifter.

\[
\begin{align*}
(3) & \quad \text{a. } \text{LIFT} = \lambda \alpha \tau \lambda P_{\langle \tau, t \rangle} P(\alpha) \\
& \quad \text{b. } \text{LIFT}([\text{John}]) = \lambda P_{\langle et, t \rangle} P(j) \\
(4) & \quad \text{a. } \text{BE} = \lambda P \lambda z [P(\lambda y. y = z)] \\
& \quad \text{b. } \text{BE}([\text{some cat}] = \text{cat}')
\end{align*}
\]

- Quantifier raising and other phrasal LF movement:

\[
\begin{align*}
\text{A.} & \quad \text{everyone} \\
& \quad \lambda x \text{ VP} \\
& \quad \text{Mary saw } x \\
\text{B.} & \quad \text{Mary} \\
& \quad \lambda x \text{ VP} \\
& \quad \text{will } x \text{ come} \\
\text{C.} & \quad \alpha_x / \alpha_{\langle \tau, t \rangle} \\
& \quad \lambda x \text{ S} \\
& \quad \ldots x_{\tau} \ldots
\end{align*}
\]

II. Categorial approaches of question semantics

- Core assumptions: (i) short answers are bare nominal; (ii) questions denote lambda abstracts; (iii) \( wh \)-items denote lambda operators.

\[
\begin{align*}
(5) & \quad \text{a. } [\text{who came}] = \lambda x [\text{people}')(x).\text{came}'(x)] \\
& \quad \text{b. } [\text{who}] = \lambda P \lambda x [\text{people}')(x).P(x)]
\end{align*}
\]

- Advantages: (i) the relation between questions and short answers is very directly modeled; (ii) similarity of \( wh \)-questions and free relatives is captured nicely.

- Problems: (i) composing multi-\( wh \) questions suffers type mismatch; (ii) it assigns different semantic types to different questions, which makes it difficult account for question coordinations; (iii) it doesn’t account for the existential semantics of \( wh \)-words in non-interrogative sentences. [But we also saw that those problems can be overcome.]
III. Intensionality

- The extension of an expression is world-dependent. The intension of an expression $X$ is a function which applies to a possible world and returns the extension of $X$ in that world.

\[(6) \quad \text{a. } [X]^w \quad \text{('the extension of } X \text{ in } w') \]
\[(6) \quad \text{b. } \lambda w.[X]^w \quad \text{('the intension of } X') \]

- The intension of a sentence is a proposition: a function from worlds to truth values.
- The intension of a predicate of type $\langle e, t \rangle$ is a property: a function from worlds to $\langle e, t \rangle$ functions.
- The intension of a definite NP is an individual concept: a function from worlds to entities.

- Propositions can also be viewed as the set of possible worlds where this proposition is true. Hence, we can use set-theoretical operations to represent the following relations and operations:

<table>
<thead>
<tr>
<th>Relations and operations</th>
<th>Set-theoretical notations</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p$ entails $q$</td>
<td>$p \subseteq q$</td>
</tr>
<tr>
<td>$p$ contradicts $q$</td>
<td>$p \cap q = \emptyset$</td>
</tr>
<tr>
<td>$p$ and $q$</td>
<td>$p \cap q$</td>
</tr>
<tr>
<td>$p$ or $q$</td>
<td>$p \cup q$</td>
</tr>
<tr>
<td>$p$ is possible</td>
<td>$p \neq \emptyset$</td>
</tr>
<tr>
<td>$p$ is necessary</td>
<td>$p = W$</td>
</tr>
</tbody>
</table>

Plan for today

- Intensionality (cont.)
- Hamblin Semantics of questions
1 Intensionality (cont.)

- The extension of a proper name or a logical expression is not world-dependent.

(7) a. For every \( w \), \([\text{John}]^w = j\) (controversial)
b. For every \( w \), \([\text{it is not the case that}]^w = \lambda p_{(s,t)} \lambda w. \neg p_w\)
c. For every \( w \), \([\text{every}]^w = \lambda Q_{(e,t)} \lambda P_{(e,t)} \forall x [Q(x) \rightarrow P(x)]\)

- Intensional-izing the theory of types

(8) Types

a. Basic types: \( e \) (individuals/entities) and \( t \) (truth values).
b. Functional types: If \( \sigma \) and \( \tau \) are types, then \( x_{\sigma,\tau} y \) is a type.
c. Intensional types: If \( \sigma \) is a type, then \( x_{s,\sigma} y \) is an intensional type.

(9) Domains

a. \( D_s = W \)
b. \( D_{\langle \sigma, \tau \rangle} = \{ f \mid f : D_{\sigma} \rightarrow D_{\tau} \} \) (functions from things of type \( \sigma \) to things of type \( \tau \))
c. \( D_{\langle s, \sigma \rangle} = \{ f \mid f : W \rightarrow D_{\sigma} \} \) (functions from possible worlds to things of type \( \sigma \))

- Intensional-izing the theory of semantic composition

(10) Intensional Functional Application

If \( \{\beta, \gamma\} \) is the set of \( \alpha \)'s daughters, \( [[\beta]] \in D_{\langle s, \sigma \rangle, \tau} \), and \( [[\gamma]] \in D_{\sigma} \), then \( [[\alpha]] = [[\beta]](\lambda w. [[\gamma]]^w) \)

Example:

(11) John believes that Mary left.

\[
\begin{array}{c}
\text{S2} \\
\text{John} \quad \text{1} \\
\text{believes}_{(st,et)} \\
\text{S1} \\
\text{that} \\
\text{Mary} \\
\text{left}
\end{array}
\]

a. \( [[\text{S1}]] = \text{left}'(m) \)
b. \( [[\text{believe}]] = \lambda p_{(s,t)} \lambda x_e. \text{believe}'(x, p) \)
c. \( [[1]] = [[\text{believe}]](\lambda w. [[\text{S1}]]^w) \) By IFA
   \( = \lambda x_e. \text{believe}'(x, \lambda w. \text{left}'_w(m)) \)
d. \( [[\text{S2}]] = [[1]]([[[\text{John}]]]) \) by FA
   \( = (\lambda x_e. \text{believe}'(x, \lambda w. \text{left}'_w(m)))(j) \)
   \( = \text{believe}'(j, \lambda w. \text{left}'_w(m)) \)

Alternatively, we can assume that the predicate \( \text{left} \) carries an world variable \( w \), which is then abstracted over by a \( \lambda \)-operator. (See Percus 2000 “Constraints on some other variables in syntax”.)

(11’)

\[
\begin{array}{c}
\text{John} \quad \text{1} \\
\text{believes}_{(st,et)} \\
\lambda w. \text{left}'_w(m) \\
\lambda w \\
\text{Mary left}_w
\end{array}
\]

\(^1\)Note that we are not actually adding \( s \) for possible worlds to our type theory. This is because (as far as we’ve seen) there are no expressions of natural language that have specific possible worlds as their values.
Discussion: (i) in (11'), what composition rule is used for node 1? (ii) Compare the following LFs. Given the lexical entry of every in (7c), consider, which LF is well-formed?

(12) Mary saw everyone.

\[
\begin{align*}
\text{S} & \quad \text{S} & \quad \text{S} \\
\lambda w \quad \lambda x & \quad \lambda w \quad \lambda x & \quad \lambda w \quad \lambda x \\
\text{everyone} & \quad \text{everyone} & \quad \text{everyone} \\
\text{S} & \quad \text{S} & \quad \text{S} \\
\text{Mary saw}_{w,x} & \quad \text{Mary saw}_{w,x} & \quad \text{Mary saw}_{w,x}
\end{align*}
\]

2 Hamblin Semantics of questions (Hamblin 1973)

2.1 Core assumptions

- A possible answer denotes a proposition. A short answer is an elliptical form of the corresponding full answer.

(13) Who came?
   a. Mary came. (full answer)
   b. Mary came. (short answer)

- A \textit{wh}-item denotes a set of individuals.

(14) a. \([\text{who}] = \{x : \text{human}(x) = 1\}\)
   b. \([\text{what}] = \{x : \text{thing}(x) = 1\}\)
   c. \([\text{which cat}] = \{x : \text{cat}(x) = 1\}\)

A question denotes the set of propositions that are possible (direct) answers of this question, called a “Hamblin (alternative) set”.

(15) a. \([\text{who came?}] = \{a \text{ came}, b \text{ came, a and b came,} \ldots\}\)
   b. \([\text{which person likes which person?}] = \{a \text{ likes b, b likes a,} \ldots\}\)
   c. \([\text{Did John come?}] = \{\text{John came, John didn’t come}\}\)
   d. \([\text{Does Mary like coffee or tea?}_{\text{ALT-Q}}]\)
      = \{\text{Mary likes coffee, Mary likes tea}\}
   e. \([\text{Does Mary like coffee or tea?}_{\text{Y/N-Q}}]\)
      = \{\text{Mary likes coffee or tea, Mary doesn’t like coffee or tea}\}
   f. \([\text{How many cats does John have?}] = \{\text{John has one cat, John has two cats,} \ldots\}\)

- Hamblin sets are composed via Point-wise Functional Application.

(16) Point-wise Functional Application

If \(\alpha\) is of type \(\langle \sigma, \tau \rangle\) and \(\beta\) is of type \(\sigma\), then

a. \([\alpha] \subseteq D_{\langle \sigma, \tau \rangle}\)

b. \([\beta] \subseteq D_{\sigma}\)

c. \(\alpha(\beta)\) is of type \(\tau\), and \([\alpha(\beta)] = \{a(b) \mid a \in [\alpha] \land b \in [\beta]\}\)
2.2 Composing questions using Hamblin Semantics

- Composing declaratives and _wh_-questions:
  - A proper name _Mary_ denotes a singleton set; thus a declarative denotes a singleton set.
  - A _wh_-item denotes a set with many individuals, thus a _wh_-question denotes a set with many propositions.

(17) a. Mary came. \[ \{ \lambda w. \text{came}'(m) \} \] 
    Mary came \[ \{ m \} \{ \lambda x. \text{came}'(x) \} \]

b. Who came? \[ \{ \lambda w. \text{came}'(x) : \text{human}'_\alpha(x) = 1 \} \] 
   who came \[ \{ x : \text{human}'_\alpha(x) = 1 \} \{ \lambda x. \text{came}'(x) \} \]

A note on _wh_-movement: In categorial approaches (and Karttunen Semantics), (non-subject) _wh_-items must undertake movement, so as to salvage type-mismatch. For _wh_-insitu languages (e.g., Chinese), categorial approaches and Karttunen semantics predicts covert movement of the _wh_-item. Hamblin Semantics has no such prediction.

- Composing polar questions:
  *Is it the case that* denotes the set with the identity function on the question nucleus and its negation.

(18) Is it the case that John left? \[ \{ \lambda w. \text{left}'(j), \lambda w. \neg \text{left}'(j) \} \]
    is it the case that \[ \{ \lambda w. \text{left}'(j) \} \]
    \begin{equation*}
    \begin{aligned}
    &\{ \lambda p.p, \lambda p\lambda w. \neg p_w \} \\
    \end{aligned}
    \end{equation*}
    John left

Exercise: Assume the type-shifted lexical entry of _or_ in (19), derive the Hamblin sets of the alternative question in (20) using PFA.

(19) \[ [\text{or}] = \lambda \alpha(\text{st},t) \lambda \beta(\text{st},t). \alpha \cup \beta \] 
    ([or] applies to two sets of propositions, and returns the union of these two sets.)

(20) Did JOHN come or MARY come?\_ALT-Q

Exercise: Compose the following polar-Q. [Consider: can we use the lexical entry of _or_ in (19)?]

(21) Is it the case that [[John came] or [Mary came]]?
2.3 Compare Hamblin Semantics and traditional categorial approaches

- **Discussion**: Are the denotations of (22a-b) equivalent under Hamblin Semantics? What about under categorial approaches? [Recall that categorial approaches assume that questions denote lambda abstracts.]

(22)  
a. Did JOHN come or MARY come? \( \text{ALT-Q} \)

b. [Among John and Mary,] which person came?

**Discussion**: Can we derive a Hamblin set based on a lambda abstract? What about retrieving a lambda abstract out of the corresponding Hamblin set?

- **An inclusive comparison between categorial approaches and Hamblin Semantics**

<table>
<thead>
<tr>
<th></th>
<th>Categorial approaches</th>
<th>Hamblin Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retrieving the question nucleus</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Getting short answers</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Getting full answers</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Uniform semantic type</td>
<td>No</td>
<td>Yes: ( \langle st, t \rangle )</td>
</tr>
<tr>
<td>Question coordinations</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Type-driven ( wh )-movement</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

- **Although Hamblin Semantics treat questions uniformly as of type \( \langle st, t \rangle \), it still has imperfections in analyzing question coordinations.**

  - Conjunction is traditionally treated as set-intersection.

    (23) \([\text{John left} \land \text{Mary stayed}] = [\text{John left}] \land [\text{Mary stayed}]\)

  - But, the conjunction of two questions cannot be the intersection of the Hamblin sets of the two questions:

    (24) \([\text{who left and who stayed}] = [\text{who left}] \land [\text{who stayed}] = \emptyset \)  \( \text{NO WAY!} \)

  - Hence, Hamblin Semantics has to define conjunction as pointwise intersection. \(^2\)

    (25) \([Q_1 \land Q_2] = \{p \land q : p \in Q_1 \land q \in Q_2\}\)

\(^2\)Inquisitive Semantics maintains the basic intersection semantics of conjunction by treating questions as sets of proposition sets. (See Ciardelli et al. 2016, “Composing Alternatives”)
3 Alternatives Semantics of focus (Rooth 1992)

- Focus affects the suitability of a sentence as answer of the given question. Observe the prosodic dependence between questions and answers:

  (26) Who invited Bill?
    a. JOHN invited Bill.
    b. # John invited BILL.
    c. # JOHN invited BILL.

- Core definitions

  (27) Every expression α has an ordinary value $[\alpha]^0$ and a focus value $[\alpha]^f$.
    a. $[\alpha]^0$ is simply the truth value of α (i.e., the one that we already know).
    b. $[\alpha]^f$ is the set of all ordinary semantic values obtained by substituting alternatives for any F-marked subparts of α.

- Compute the focus value compositionally:

  (28) Terminal nodes (TN)
  $[\alpha_F] = \{[\alpha]^0\}$ if α is focused
  \[ \text{Pointwise Functional Application (PFA)} \]
  $[\alpha(\beta)]^f = \{a(b) | a \in [\alpha]^f, b \in [\beta]^f\}$

**Exercise:** Compute the focus value of the following sentences compositionally:

(29) JOHN$_F$ invited Bill.

```
     S
    / \n   JOHN$_F$ invited Bill
```

- Use the Rooth-style terms to define Hamblin sets:

  (30) a. $[\text{who}]^0$ is undefined
    b. $[\text{who}]^f = \{x : \text{human}^f_w(x) = 1\}$
    c. $[\text{TP who came}]^0$ is undefined
    d. $[\text{TP who came}]^f = \{\lambda w.\text{came}^f_w(x) : \text{human}^f_w(x) = 1\}$
    e. $[\text{C_{[+w]}} [\text{TP}]]^0 = [\text{TP}]^f$ (Beck & Kim 2006, see also Shimoyama 2001)
      (interrogative C$^0$ returns the focus-semantic value of TP as the ordinary semantic value of CP.)

---

3Principle of Interpretability: An LF must have an ordinary semantic value (Beck 2006: p. 16)
Exercise: Use the following toy LF to derive the Hamblin set for *Who does John like?*.

(31) \[
\begin{array}{c}
\text{CP} \\
\text{C} & \text{TP} \\
\text{John} & \text{invited} & \text{who}
\end{array}
\]

• Explain the prosodic dependence between questions and answers:

(32) **Question-Answer Congruence** (Rooth 1992: 86)

A sentence S is a possible answer of a question Q iff \([Q]^0 \subseteq [S]^f\)

Examples:

(33)  a. \([\text{who invited Bill?}]^0 \subseteq [\text{JOHN}_F \text{ invited Bill}]^f\)

b. \([\text{Did JOHN or MARY invited Bill?}]^0 \subseteq [\text{JOHN}_F \text{ invited Bill}]^f\)