On questions and presuppositions in typed inquisitive semantics

Lucas Champollion
champollion@nyu.edu
Joint work with Ivano Ciardelli and Floris Roelofsen∗

2nd Workshop on Inquisitiveness Below and Beyond the Sentence Boundary (InqBnB)
Amsterdam, 18-19 December 2017
University of Amsterdam

Abstract

The first part of this talk lays out a compositional account of wh-questions in typed inquisitive semantics (Ciardelli, Roelofsen & Theiler 2017). Relevant issues include multiple wh-questions, the interaction between wh-items and disjunction, and de dicto readings of which-questions.

Motivated by the asymmetry between restrictor and nuclear scope of which-questions, the second part of this talk takes first steps towards modeling presuppositions of questions in inquisitive semantics.

1 Introduction

• Groenendijk & Stokhof (1984) provided a theory of questions that improved in several respects over Karttunen (1977)

• Basic inquisitive logic (Ciardelli, Groenendijk & Roelofsen 2013) in turn improved in some ways on these theories, but did not preserve all of their achievements

∗Thanks to Theo Janssen and the audience at the 2015 Amsterdam workshop on questions in logic and semantics for helpful comments. Sections 1-6 of this talk were previously presented at that workshop. Support from the NYU URCF is gratefully acknowledged.
### Feature K77 GS84 InqB

<table>
<thead>
<tr>
<th>Feature</th>
<th>K77</th>
<th>GS84</th>
<th>InqB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compositional derivations</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Interpreting short answers</td>
<td>no</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>De dicto readings of which questions</td>
<td>no</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Ability to interpret conjoined questions</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Ability to quantify into questions</td>
<td>no</td>
<td>(yes)</td>
<td>yes</td>
</tr>
<tr>
<td>Uniform disjunction across declaratives and interrogatives</td>
<td>no</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Mention-some readings</td>
<td>no</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Conditional questions</td>
<td>no</td>
<td>no</td>
<td>yes</td>
</tr>
</tbody>
</table>

- InqB is a first-order logic, and as such does not provide a means to compositionally assign meanings to subsentential constituents. Typed Inquisitive Semantics (Ciardelli, Roelofsen & Theiler 2017) provides the bridge between InqB and compositional semantics. We will build on it here.

- As an intermediate step in the compositional derivation, Groenendijk & Stokhof (1984, 1989) compute the abstract of a question—an n-place property where n is the number of wh-words. They use it to interpret short answers:

  (1) a. Who walks? — John. *Abstract: \( \lambda x. x \text{ walks} \)
  
  b. Who loves whom? — John, Mary. *Abstract: \( \lambda y. \lambda \lambda x. x \text{ loves } y \)

- Typed Inquisitive Semantics gives us the means to compute the abstract of a question.

- Groenendijk & Stokhof (1984) point out that the following inference is invalid when murderer is taken *de dicto*:

  (2) a. Holmes knows who is tall.
  
  b. \( \Rightarrow \) Holmes knows which murderer is tall.

- Karttunen (1977) only generates the *de re* reading. The issue has not been revisited in InqB.

- Some things left for another occasion: NPI licensing in questions (Guerzoni & Sharvit, Nicolae); intervention effects (Beck, Haida, Kotek); cross-linguistic aspects (Shimoyama); superiority effects (Kotek); functional readings; sensitivity to different ways of identifying the individuals in the domain of discourse (Aloni); syntactic restrictions on mention-some readings (Fox, Xiang); pragmatics of mention-some versus mention-all (van Rooij); modals in questions (Spector).

## 2 Typed inquisitive semantics

- Typed inquisitive semantics is a combination of compositional semantics and basic inquisitive logic.
Possible worlds \((w, w' \ldots)\) are primitives (type \(s\)). Possibilities \((p, p' \ldots)\) are sets of possible worlds (type \(\langle s, t \rangle\)). Inquisitive propositions \((P, P' \ldots)\) are sets of possibilities (type \(\langle st, t \rangle\)).

- We abbreviate \(\langle e, \langle et \rangle \rangle\) as \(\langle e^2, t \rangle\), and \(\langle e, \langle e, \langle et \rangle \rangle \rangle\) as \(\langle e^3, t \rangle\), etc. We also write \(p(x^n)\) for \(p(x_1)(x_2) \ldots (x_n)\). Similarly, we write \(\lambda x^n.b\) for \(\lambda x_1 \lambda x_2 \ldots x_n.b\); and similarly for quantifiers.

- We let \(\text{talks}\) denote the relation that holds between \(x\) and \(p\) iff \(p\) entails that \(x\) talks:

\[
\begin{align*}
\langle \text{talks} \rangle_g &= \lambda x\lambda p \forall w.p(w) \rightarrow \text{talk}(x)(w) \\
&= \lambda x\lambda p.p \subseteq \lambda w.\text{talk}(x)(w) & \text{type } \langle e, \langle st, t \rangle \rangle
\end{align*}
\]

- We abbreviate \(\langle st, t \rangle\) as \(T\). For \(p_0\) a possibility (type \(\langle s, t \rangle\)), we write \(\tilde{p}_0\) for the inquisitive proposition \(\lambda p.p \subseteq p_0\). Similarly, for \(p_n\) of type \(\langle e^n, \langle s, t \rangle \rangle\), we write \(\tilde{p}_n\) for \(\lambda x_1 \ldots \lambda x_n.\lambda p.p \subseteq \lambda w.p_n(x_1) \ldots (x_n)(w)\). For example:

\[
\begin{align*}
\langle \text{talks} \rangle_g &= \tilde{\text{talk}} \\
&= \lambda x\lambda p.p \subseteq \lambda w.\text{talk}(x)(w) & \text{type } \langle e, T \rangle
\end{align*}
\]

- We represent proper names as constants and use function application to combine meanings:

\[
\begin{align*}
\langle \text{John talks} \rangle_g &= \tilde{\text{talk}}(j) = \lambda p.p \in \tilde{\text{talk}}(j) = \lambda p.p \subseteq \lambda w.\text{talk}(j)(w) & \text{type } T
\end{align*}
\]

### 3 Propositional connectives

- We assume a type-polymorphic theory of coordination (e.g. Partee & Rooth 1983). Simplifying slightly, define an inquirable type as either the type \(T\) or a type \(\langle \alpha, \beta \rangle\) where \(\alpha\) is any type and \(\beta\) is an inquirable type.

- We define inquisitive negation, \(\neg\), as in basic inquisitive semantics, and generalize it to higher types:

\[
\begin{align*}
\neg_{\langle T, T \rangle} &= \lambda P \lambda p.\forall q.P(q) \rightarrow p \cap q = \emptyset & \text{type } \langle T, T \rangle \\
\neg_{\langle \alpha T, \alpha T \rangle} &= \lambda P_{\langle \alpha T \rangle} \lambda x_{\alpha}.\neg_{\langle T, T \rangle} P(x) & \text{type } \langle \alpha T, \alpha T \rangle
\end{align*}
\]

- We represent the meaning of ordinary linguistic negation via inquisitive negation:

\[
\begin{align*}
\langle \text{not} \rangle_g &= \lambda P.\neg P & \text{type } \langle \alpha T, \alpha T \rangle
\end{align*}
\]

For any inquirable type \(\tau\) we define:

\[
\begin{align*}
\langle \text{and} \rangle_g &= \lambda P_{\tau}.\lambda Q_{\tau}.P \cap Q & \text{type } \langle \tau, \tau \rangle \\
\langle \text{or} \rangle_g &= \lambda P_{\tau}.\lambda Q_{\tau}.P \cup Q & \text{type } \langle \tau, \tau \rangle
\end{align*}
\]

- As a special case, we will write \(\mathcal{A}\) (inquisitive conjunction) for the case where we conjoin two terms \(P\) and \(P'\) of type \(T\), and similarly for \(\mathcal{W}\):
· Inquisitive conjunction and disjunction share various desirable properties with ordinary conjunction and disjunction, such as idempotence and associativity.

· We assume that proper names can be lifted to generalized quantifiers (note the type):

\[
\text{a. } \mathcal{Lift}(\text{John}) \mathcal{Lift}(\text{Mary}) \mathcal{W} \equiv \lambda p. p \subseteq \lambda w. \text{walk}(j)(w) \lor p \subseteq \lambda w. \text{walk}(m)(w)
\]

· We can now interpret *John and Mary walk* and *John or Mary walks*.

\[
\text{a. } \mathcal{Lift}(\text{John} \land \text{Lift}(\text{Mary}) \text{ walk}) \equiv \text{walk}(j) \land \text{walk}(m) \quad \text{type } T
\]

\[
\text{b. } \mathcal{Lift}(\text{John} \lor \text{Lift}(\text{Mary}) \text{ walks}) \equiv \text{walk}(j) \lor \text{walk}(m) \quad \text{type } T
\]

· *John or Mary walks* is interpreted as an inquisitive proposition with two alternatives:

\[
\text{walk}(j) \lor \text{walk}(m) = \lambda p. p \subseteq \lambda w. \text{walk}(j)(w) \lor p \subseteq \lambda w. \text{walk}(m)(w)
\]

· We can define type-shifted versions of the inquisitive operators ! (noninquisitive closure) and ? (noninformative closure):

\[
\text{a. } ? \equiv \lambda P. P \cup \neg \neg P \quad \text{type } \langle \tau, \tau \rangle
\]

\[
\text{b. } ! \equiv \neg \circ \neg \quad \text{type } \langle \tau, \tau \rangle
\]

· We assume that any declarative sentence with falling intonation contains ! at its root.

· This has the following effect: Where \( A \lor B \) denotes the set of all possibilities that entail \( A \) or entail \( B \), \(! (A \lor B)\) denotes the set of all possibilities that entail \( A \lor B \), including those that do not entail one of the disjuncts.

\[
\mathcal{Lift}[\text{John or Mary walk}] \equiv \text{walk}(j) \lor \text{walk}(m) = \lambda p. p \subseteq \lambda w. \text{walk}(j)(w) \lor \text{walk}(m)(w)
\]

· Finally, we can define inquisitive quantifiers:

\[
\text{a. } \exists x \phi \equiv \lambda p. \exists x \phi(p)
\]

\[
\text{b. } \forall x \phi \equiv \lambda p. \forall x \phi(p)
\]
4 Wh-questions and the abstract

• We assume that questions, whether embedded or not, are headed by a silent $Q$ morpheme (Baker 1970), which projects an interrogative nucleus. The complement of $Q$ is the abstract.

\[
\begin{array}{c}
\text{interrogative} \\
\text{nucleus} \\
Q \quad \text{abstract}
\end{array}
\]

• The abstract is of type $\langle e^n, T \rangle$: e.g. $\langle e, T \rangle$ for single-$wh$ questions, $\langle e, eT \rangle$ for double-$wh$-questions.

• We could naively assume that wh-phrases like $who$ are identity functions:

\[
\begin{array}{c}
\text{e.g.} & [\text{who}]_g \text{ in subject position} = \lambda P_e \lambda x. P(x)
\end{array}
\]

• This differs from the treatment in Theiler (2014), where wh-phrases are treated as inquisitive existentials.

• But this will not work when we need to pass the abstract across sentence boundaries:

\[
\begin{array}{c}
\text{Whom do you want Mary to invite?}
\end{array}
\]

• Here, want expects a proposition, so whom must leave a trace behind or be interpreted in situ at LF.

• This process can violate islands, so an in-situ based account is preferable (cf. Reinhart 1997):

\[
\begin{array}{c}
\text{a. Who thinks that who walks?} \\
\text{b. Who will be offended if we invite whom?}
\end{array}
\]

• So we assume instead, following Baker (1970), that who carries an index, and that the $Q$ morpheme binds such indices or triggers lambda abstraction below it:

\[
\begin{array}{c}
\text{a. Who walks? } \leadsto Q \begin{bmatrix} 1 [\text{who}, \text{walks}] \end{bmatrix} \\
\text{b. Who loves whom? } \leadsto Q \begin{bmatrix} 1 [2 [\text{who}, \text{loves whom}] \end{bmatrix} \\
\text{c. Who thinks that who walks? } \leadsto Q \begin{bmatrix} 1 [2 [\text{who}, \text{thinks} [\text{that who}, \text{walks}]]] \end{bmatrix}
\end{array}
\]

• Sometimes the abstract will be noninquisitive:

\[
\begin{array}{c}
\text{a. } [ [1 [\text{who}, \text{walks}] ] ] = \lambda x_1. \text{walk}(x_1) \\
\text{b. } [ [1 [2 [\text{who}, \text{loves whom}] ] ] ] = \lambda x_1, x_2. \text{love}(x_2)(x_1)
\end{array}
\]

• Sometimes it will be inquisitive:
Who walks or talks?

a. \([Q \left[ 1 \left[ \text{who}_1 \left[ \text{walks or talks} \right] \right] \right]]\)

b. \(\left[[1 \left[ \text{who}_1 \left[ \text{walks or talks} \right] \right] \right]] = \lambda x_1. \text{walk}(x_1) \lor \text{talk}(x_1)\)

- The basic meaning InqB assigns to (21a) and (21b) captures their mention-some reading:

\begin{align}
(24) \quad & \exists x. \text{walk}(x) \\
& \exists x \exists y. \text{love}(y)(x)
\end{align}

- For example, \((24a)\) has the following alternatives:

*that John walks, that Mary walks, …, that nobody walks*

- It would be a mistake to treat inquisitive abstracts in the same way, however:

\begin{align}
(25) \quad & \exists x. \text{walk}(x) \lor \text{talk}(x)
\end{align}

- This has the following alternatives: *that John walks, that John talks, that Mary walks, that Mary talks, …, and that nobody walks or talks.*

- A better translation uses noninquisitive closure:

\begin{align}
(26) \quad & \exists x. !\left( \text{walk}(x) \lor \text{talk}(x) \right)
\end{align}

- This has the alternatives *that John walks or talks, that Mary walks or talks, …, that nobody walks or talks.*

- What is responsible for the introduction of noninquisitive closure?

- Compositionally, we seem to have two options: \(!\) is introduced by \(Q\), or by the \(wh\)-phrases.

- In non-\(wh\) questions, \(Q\) often does not seem to introduce \(!\).

\begin{align}
(27) \quad & \text{Would you like coffee}↑, \text{or tea}↓? \\
& \neq \text{Is it the case that you would like either coffee or tea?}
\end{align}

- So we assume that it is the \(wh\)-phrases that are responsible for the introduction of \(!\).

\begin{align}
(28) \quad & \left[ \text{who}_1 \right]_g = \lambda P(e,T) P(g(i)) \quad \text{type} \langle eT, T \rangle
\end{align}
• In nonsubject position, we resolve type mismatches by type-shifting (Hendriks 1993).

5 The Q operator

• The Q operator maps abstracts to inquisitive propositions.

• As is well known, there are several relevant candidate propositions:

(31) John knows who is tall.
    a. John knows of some x that x is tall.  

mention-some
Some special cases:

- Groenendijk & Stokhof (1984) take (31c) as basic, which makes it hard to model (31a) and (31b) (Heim 1994, Beck & Rullmann 1999).
- Theiler (2014) takes (31a) as basic, and models (31c) through an additional exhaustification operation.
- We assume that exhaustification optionally takes place within the interrogative nucleus; the precise “flavor” of exhaustivity in embedded questions is determined by an embedding operation that does not play a role in root questions (Theiler, Roelofsen & Aloni 2016).
- We base the meaning of $Q$ on the inquisitive existential $∃$ and on the operator $?. (This is a simplification. For certain purposes involving non-$wh$ questions, it would be more accurate to use $⟨?⟩$, which leaves inquisitive meanings alone, and applies $? only to noninquisitive meanings.)

\[
[Q^0]^g = \lambda P_{(e,T)}.?∃x^n.P(x^n)
\]

Some special cases:

\[
[Q^1]_{g} = \lambda P_{(e,T)}.?∃x.P(x)
\]

Some special cases:

\[
[Q^2]_{g} = \lambda R_{(e,T)}.?∃x∀y.R(x)(y)
\]

A second version of the operator has exhaustivity optionally built in:

\[
[Q^{+\text{exh}}^n]_{g} = \lambda R_{(e,T)}.\lambda p.∀q \subseteq p.(\lambda x^n. R(x^n)(q) = \lambda x^n. R(x^n)(p))
\]

Some special cases:

\[
[Q^{+\text{exh}}^0]_{g} = \lambda P_{(e,T)}.?P = \lambda P_{(e,T)} \lor \neg P
\]

\[
[Q^{+\text{exh}}^1]_{g} = \lambda P_{(e,T)}.\lambda p.∀q \subseteq p.(\lambda x^n. R(x^n)(q) = \lambda x^n. R(x^n)(p))
\]

\[
[Q^{+\text{exh}}^2]_{g} = \lambda R_{(e,T)}.\lambda p.∀q \subseteq p.(\lambda yλx.R(y)(x)(q) = λyλx.R(y)(x)(p))
\]

6 Which-questions

- Following Groenendijk & Stokhof (1984), we assume that which-questions interpret their restrictor noun in situ.

\[
[\text{which}.]^g = \lambda N_{(e,T)}\lambda P_{(e,T)}.![N(g(i)) \land P(g(i))]
\]
• As for who, in nonsubject position we resolve type mismatches by type-shifters on the verb.

\[
\lambda p. \forall q \subseteq p. ((\lambda x. q \in ![\text{murderer}(x) \land \text{tall}(x)]) = (\lambda x. p \in ![\text{murderer}(x) \land \text{tall}(x)]))
\]

\[
\begin{array}{c}
\lambda p. \forall q \subseteq p.((\lambda x. q \in ![\text{murderer}(x) \land \text{tall}(x)]) \\
= (\lambda x. p \in ![\text{murderer}(x) \land \text{tall}(x)])
\end{array}
\]

\[
\begin{array}{c}
\langle e, T \rangle \\
\lambda x. ![\text{murderer}(x) \land \text{tall}(x)]
\end{array}
\]

\[
\begin{array}{c}
\lambda P_{(e, T)}. \lambda p. \forall q \subseteq p. \\
(\lambda x. P(x)(q) = \lambda x. P(x)(p))
\end{array}
\]

\[
\begin{array}{c}
\langle eT, T \rangle \\
\lambda P_{(e, T)}. ![\text{murderer}(g(1)) \land \text{tall}(g(1))]
\end{array}
\]

\[
\begin{array}{c}
\langle eT, T \rangle \\
\lambda P_{(e, T)}. ![\text{murderer}(g(1)) \land P(g(1))]
\end{array}
\]

\[
\begin{array}{c}
\langle eT, T \rangle \\
\lambda P_{(e, T)}. ![\text{murderer}(g(1)) \land P(g(1))]
\end{array}
\]

\[
\begin{array}{c}
\lambda x. \text{tall}(x)
\end{array}
\]

\[
\begin{array}{c}
\lambda N_{(e, T)} \lambda P_{(e, T)}. ![N(g(1)) \land P(g(1))]
\end{array}
\]

\[
\begin{array}{c}
\lambda x. \text{murderer}(x)
\end{array}
\]

• This gives us access to the kind of object we need in order to compute a de dicto reading.

7 The presuppositions of wh-questions

• Following Groenendijk & Stokhof (1984) we have given what amounts to a symmetric account. This, however, can’t be the whole story:

\[
\text{(38) Adapted from Higginbotham (1996):}
\]

\[
\text{a. Which man is a bachelor?}
\]

\[
\text{b. #Which bachelor is a man?}
\]

• Rullmann & Beck (1998) capture the contrast between these questions in terms of the presuppositions of the which-phrase. We will implement a version of their account in a presuppositional inquisitive semantics.

• In InqB, the semantic content of a sentence is a set of possibilities. Each possibility, in turn, is a set of possible worlds.

• We can think of each possibility as representing a potential update of the common ground.
We now propose to model possibilities as having two components, one presuppositional and one at-issue.

The presuppositional component represents the precondition of the potential update, the at-issue component determines the effect of the update in case the precondition is met (for reasons of space and time a detailed discussion of the discourse pragmatics that we assume is suppressed here).

For a possibility $s$ to support *Which bachelor is a man?* there should be some individual $d$ such that:

- the presuppositional component of $s$ entails that $d$ is a bachelor, and
- the at-issue component of $s$ entails that $d$ is a man.

Given the background assumption that all bachelors are men (which can be encoded using a meaning postulate), the presuppositional support requirement guarantees that the at-issue support requirement will be satisfied. In other words, the at-issue support requirement is vacuous, in light of the presuppositional requirement. This explains, in short, why the question is odd.

Making this more precise calls for a presuppositional extension of inquisitive semantics.

## 8 Presuppositional typed inquisitive semantics

### 8.1 Basic semantic notions

#### Possibilities

- A possibility $s$ is pair $\langle \pi_s, \alpha_s \rangle$, where $\pi_s$ and $\alpha_s$ are both sets of worlds, and $\pi_s \supseteq \alpha_s$.

- We call $\pi_s$ the presupposition of $s$, and $\alpha_s$ the at-issue information of $s$.
  - If $\pi_s = W$, then the presupposition of $s$ is trivial.
  - If $\alpha_s = \pi_s$, then the at-issue information of $s$ is trivial.

- Given that $\alpha_s \subseteq \pi_s$, $s$ can always be represented as a three-valued total function from worlds into $\langle 1, 0, \# \rangle$, but we focus on the pair-representation.

#### Ordering possibilities

- $s \sqsubseteq t$ iff $\pi_s \subseteq \pi_t$ and $\alpha_s \subseteq \alpha_t$

- The inconsistent possibility is $\langle \emptyset, \emptyset \rangle$. It is the strongest of all possibilities.

- A possibility $s$ is at-issue inconsistent iff $\alpha_s = \emptyset$.

#### Abbreviations
\[ s^\top = \langle \pi_s, \pi_s \rangle \] trivializes the at-issue information of \( s \)

\[ s^\perp = \langle \pi_s, \emptyset \rangle \] makes the at-issue information of \( s \) inconsistent

### Propositions

- An inquisitive proposition is a set of possibilities \( P \in D_{\langle st, \langle st, t \rangle \rangle} \) such that:
  - \( P \) is non-empty;
  - If \( s \in P \) and \( t \subseteq s \) then \( t \in P \) as well.

#### 8.2 Type theory

- We use relational type theory, TT2, which has the same basic types as TY2, but in addition to functional types it also has relational (cartesian product) types.
  - Basic types: \( e, s, t \)
  - Functional types: if \( \alpha \) and \( \beta \) are types, then \( \langle \alpha, \beta \rangle \) is as well
  - Relational types: if \( \alpha \) and \( \beta \) are types, then \( \alpha \times \beta \) is as well

- The type of inquisitive propositions: \( T := \langle st \times st, t \rangle \)

- Abbreviations in the object language:
  - \( \pi_s := \pi_1(s) \)
  - \( \alpha_s := \pi_2(s) \)
  - \( s^\top = \langle \pi_s, \pi_s \rangle \)
  - \( s^\perp = \langle \pi_s, \emptyset \rangle \)
  - \( s^\alpha = \langle \alpha_s, \alpha_s \rangle \)
  - \( s \subseteq t := \alpha_s \subseteq \alpha_t \wedge \pi_s \subseteq \pi_t \)
  - true(\( P, w \)) := \( P(\{w\}, \{w\}) \)
  - presup(\( P \)) := \( \lambda s. P(s^\perp) \)
  - \( A(P) := \lambda s. P(s^\alpha) \)
  - \( |P| := \lambda w. true(P, w) \)
  - \( s[P] := \langle \pi_s \cap |P|, \alpha_s \cap |P| \rangle \)
  - \( Inq(P) := \forall s[Ps \rightarrow \alpha_s \subseteq \pi_s] \wedge \forall s \forall t[t \subseteq s \wedge Ps \rightarrow Pt] \)

- Inquisitive connectives and transplication in the object language:
  - \( \perp := \lambda s. (\alpha_s = \emptyset) \)
  - \( P \wedge Q := \lambda s. P(s) \wedge Q(s[P]) \)
  - \( P \rightarrow Q := \lambda s. (P(s^\perp) \wedge \forall t \subseteq s. (P(t) \rightarrow Q(t[P]))) \)
\[ \neg P := P \rightarrow \bot \]
\[ P \lor Q := \lambda s. P(s) \lor Q(s) \quad \text{[existential projection]} \]
\[ P \lor_{lr} Q := \lambda s. P(s) \lor Q(s[\neg P]) \quad \text{[left to right filtering]} \]
\[ P \lor_{sym} Q := \lambda s. P(s[\neg Q]) \lor Q(s[\neg P]) \quad \text{[symmetric filtering]} \]
\[ P Q := \lambda s. P(s) \land Q(s^T) \quad \text{[transplication]} \]

- Projection operators:
  - \(? : P \lor \neg P\)
  - \! : \lambda s. \exists S. (\forall t \in S : t \models \phi \land \bigcup S = s) \quad \text{where } \bigcup S = \langle \bigcup_{s \in S} s_{\pi} , \bigcup_{s \in S} s_{\alpha} \rangle

- Conditional inquisitive projection:
  - \langle \? \rangle := \lambda P. \lambda s. P(s) \lor (P = \! P \land \neg \neg P(s))

This gives:
- If \( P \) is non-inquisitive then \( \langle \? \rangle P = ?P \)
- If \( P \) is inquisitive then \( \langle \? \rangle P = P \)

- Inquisitive universal and existential quantifiers:
  - \( \forall x.P := \lambda s. \forall x. P(s) \)
  - \( \exists x.P := \lambda s. \exists x. P(s) \)

- Abbreviation for entailment in the object language:
  - \( P \models Q := \forall s. P(s) \rightarrow Q(s) \)

- Exhaustivity operator, needed for which-questions:
  - \( \text{exh}(x, R_{(e,T)}) := R(x) \land \lambda s. \forall y. (Rx \not\models Ry \rightarrow \neg(Ry)(s)) \)

9  

9.1  Predicates

- Non-presuppositional predicate:
  - \( \text{walk}' = \lambda x \lambda s. \alpha_s \subseteq \pi_s \land \alpha_s \subseteq W(x) \)

- For \( p_o \) a state (type \( \langle s, t \rangle \)), we write \( \widehat{p_o} \) for the inquisitive proposition:
  - \( \lambda s. \alpha_s \subseteq \pi_s \land \alpha_s \subseteq p_o \)

Similarly, for \( p_n \) of type \( \langle e^n, st \rangle \), we write \( \widehat{p_n} \) for:
- \( \lambda x \lambda s. \alpha_s \subseteq \pi_s \land \alpha_s \subseteq p_n(x) \)
• So now we can write: \( \text{walk'} = \hat{W} \)

• Presuppositional predicate:

\[
\begin{align*}
\text{bachelor'} &= \lambda x. \text{unmarried}(x)_{\text{male}(x)} \\
&= \lambda x. \lambda s. \text{unmarried}(x)(s) \land \text{male}(x)(s^\top) \\
&= \lambda x. \lambda s. \alpha_s \subseteq \pi_s \land \alpha_s \subseteq \text{unmarried}(x) \land \pi_s \subseteq \text{male}(x)
\end{align*}
\]

9.2 First steps towards presuppositional questions

• Interrogative operator:

\[
Q'_n = \lambda P.(e^\cdot T).((\langle \text{?} \rangle \exists \bar{x}.P(\bar{x}))(\text{?}) \exists \bar{x}.P(\bar{x}))
\]

• This presupposes that at least one of the alternatives holds.

• \( \text{Who} \):

\[
\text{who}' = \lambda P. !P(x_i)
\]

• Basic entry for \( \text{which} \):

\[
\text{which}' = \lambda N. \lambda P. !P(x_i)_N(x_i)
\]

Refinement to capture that the extension of \( N \) is presupposed to be fixed:

\[
\text{which}' = \lambda N. \lambda P. !P(x_i)_N(x_i); \forall x?N(x)
\]

Refinement to capture the fact that \( \text{which} \)-questions only generate strongly exhaustive readings:

\[
\text{which}' = \lambda N. \lambda P. [\text{exh}(x_i, \lambda x. !(N(x) \land P(x)))]_N(x_i); \forall x?N(x)
\]

With a singular noun as a complement, this amounts to:

\[
\text{which}' = \lambda N. \lambda P. [!(N(x_i) \land P(x_i)) \land \\
\lambda s. \forall y \neq x_i. (\neg !(N(y) \land P(y))(s))]_N(x_i); \forall x?N(x)
\]

This captures:

- Together with the \( Q \) operator, which triggers the presupposition that at least one of the alternatives holds, we predict a uniqueness presupposition for singular \( \text{which} \)-questions.
The resolution conditions require a possibility $s$ such that for some set of individuals $x_i$, $s$ presupposes that $x_i$ are $N$, and it asserts that $x_i$ are the only $N$ who are $P$. This improves on Rullmann & Beck (1998). There it is predicted that it is sufficient to establish of some $x_i$ that they are $N$ who are $P$ (not that they are the only ones).

- Symmetry example, singular predicates:

  (46) Which man is a bachelor?

  (47) $[Q_1 \text{[which_1 man is a bachelor]}]$

  (48) $\exists x [\text{exh}(x, \lambda y (\text{male}(y) \land \text{unmarried}(y)))]|_{\text{male}(x)}! \exists x \ldots$

  For a possibility (potential update) $s$ to support the sentence:

  - the presuppositional component of $s$ entails that there is exactly one (man who is a) bachelor
  - Of some specific individual $d$, the presuppositional component of $s$ entails that $d$ is a man, and
  - the at-issue component of $s$ entails that $d$ is the unique (man who is a) bachelor.

- The other symmetry example, still singular predicates:

  (49) Which bachelor is a man?

  (50) $[Q_1 \text{[which_1 bachelor is a man]}]$

  (51) $\exists x [\text{exh}(x, \lambda y.\text{unmarried}(y))]|_{\text{male}(y); \text{unmarried}(x); \text{male}(x)}! \exists x \ldots$

  For a possibility (potential update) $s$ to support the sentence:

  - the presuppositional component of $s$ entails that there is exactly one man who is a bachelor
  - Of some specific individual $d$, the presuppositional component of $s$ entails that $d$ is a bachelor, and
  - the at-issue component of $s$ entails that $d$ is the unique (man who is a) bachelor.

The presuppositional support requirements guarantee that the at-issue support requirement will be satisfied. In other words, the at-issue support requirement is vacuous, in light of the presuppositional requirements. Call this a Strawson tautology. This explains why the question is odd.

- A sentence $\varphi$ is a Strawson tautology iff for any state $s$: $s \models \varphi \iff s^T \models \varphi$
10 Conclusion

- Typed inquisitive semantics provides compositional derivations
- By reusing the abstract as in Groenendijk & Stokhof (1984), we can interpret short answers
- By assuming that which-questions interpret their noun in situ, we account for de dicto readings
- Presuppositional inquisitive semantics allows us to capture presuppositions of which-questions and restrictor-nucleus asymmetries
- To do: multiple which-questions and much more!

References


