Some questions in typed inquisitive semantics

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

Workshop on questions in logic and semantics
University of Amsterdam
December 15, 2015

Abstract

This talk lays out a compositional account of wh-questions in typed inquisitive semantics (Theiler, 2014; Ciardelli and Roelofsen, 2015). Relevant issues include multiple wh-questions, the interaction between wh-items and disjunction, and de dicto readings of which-questions.

1 Introduction

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

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

<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 logic, and as such does not provide a means to compositionally assign meanings to subsentential constituents. Typed Inquisitive Semantics (Theiler, 2014; Ciardelli and Roelofsen, 2015) provides the bridge between InqB and compositional semantics. We will build on it here.

∗Thanks to Theo Janssen for helpful comments. Support from the NYU URCF is gratefully acknowledged.
As an intermediate step in the compositional derivation, Groenendijk and Stokhof (1984, 1989) compute the abstract of a question—an n-place property where n is the number of wh-words—and use it to interpret short answers:

(1)  
\[\begin{align*}
\text{a.} & \quad \text{Who walks? — John. Abstract: } \lambda x.x \text{ walks} \\
\text{b.} & \quad \text{Who loves whom? — John, Mary. Abstract: } \lambda y\lambda x.x \text{ loves } y
\end{align*}\]

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

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

(2)  
\[\begin{align*}
\text{a.} & \quad \text{Holmes knows who is tall.} \\
\text{b.} & \quad \Rightarrow \text{Holmes knows which murderer is tall.}
\end{align*}\]

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

## 2 Typed inquisitive semantics

Typed inquisitive semantics is a combination of compositional semantics and basic inquisitive logic (Theiler, 2014; Ciardelli and Roelofsen, 2015).

Possible worlds \((w, w' \ldots)\) are primitives (type \(s\)). States \((p, p' \ldots)\) are sets of possible worlds (type \(\langle s, t \rangle\)). Inquisitive propositions \((P, P' \ldots)\) are sets of states (type \(\langle s, t \rangle\)).

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 \(\lambda x_1 \ldots x_n.b\); and similarly for quantifiers.

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

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

We abbreviate \(\langle st, t \rangle\) as \(T\). For \(p_o\) a state (type \(\langle s, t \rangle\)), we write \(\widehat{p_o}\) for the inquisitive proposition \(\lambda p. p \subseteq p_o\). Similarly, for \(p_n\) of type \(\langle e^n, \langle s, t \rangle \rangle\), we write \(\widehat{p_n}\) for \(\lambda x_1 \cdots x_n\lambda p. p \subseteq \lambda w.p_n(x_1) \cdots (x_n)(w)\). For example:

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

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

\[\begin{align*}
\text{[[John talks]]}_g = \widehat{\text{talk}}(j) = \lambda p.p \in \widehat{\text{talk}}(j) = \lambda p.p \subseteq \lambda w.\text{talk}(j)(w) & \text{ type } T
\end{align*}\]
3 Propositional connectives

- We assume a type-polymorphic theory of coordination (e.g. Partee and 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:

\[ \neg_{\langle T, T \rangle} = \lambda P \lambda q. P(q) \rightarrow p \land q = \emptyset \quad \text{type } \langle T, T \rangle \]
\[ \neg_{\langle \alpha T, \alpha T \rangle} = \lambda P \lambda x. \neg_{\langle (T, T) \rangle} P(x) \quad \text{type } \langle \alpha T, \alpha T \rangle \]

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

\[ \lambda P. \neg_{\langle T, T \rangle} P \]

For any inquirable type $\tau$ we define:

\[ \begin{align*}
\langle \text{and} \rangle_{\tau} & = \lambda P \lambda Q. P \land Q & \text{type } \langle \tau, \tau \tau \rangle \\
\langle \text{or} \rangle_{\tau} & = \lambda P \lambda Q. P \lor Q & \text{type } \langle \tau, \tau \tau \rangle
\end{align*} \]

- As a special case, we will write $\land$ (inquisitive conjunction) for the case where we conjoin two terms $P$ and $P'$ of type $T$, and similarly for $\lor$:

\[ \begin{align*}
\langle \text{Li} \rangle_{\text{John}} \land \langle \text{Li} \rangle_{\text{Mary}} \land \text{walk} & = \lambda p. p \subseteq \lambda w. \text{walk}(j) \land \text{walk}(m) \\
\langle \text{Li} \rangle_{\text{John}} \lor \langle \text{Li} \rangle_{\text{Mary}} \land \text{walk} & = \lambda p. p \subseteq \lambda w. \text{walk}(j) \lor \text{walk}(m)
\end{align*} \]

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

\[ \langle \text{Lift(John)} \rangle_{\tau} = \lambda P \langle e, T \rangle. P(j) \quad \text{type } \langle e T, T \rangle \]

- We can now interpret John and Mary walk and John or Mary walks.

\[ \begin{align*}
\langle \text{Lift(John) and Lift(Mary) walk} \rangle_{\tau} & = \text{walk}(j) \land \text{walk}(m) & \text{type } T \\
\langle \text{Lift(John) or Lift(Mary) walks} \rangle_{\tau} & = \text{walk}(j) \lor \text{walk}(m) & \text{type } T
\end{align*} \]

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

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

- We can define type-shifted versions of the inquisitive operators $!$ (noninquisitive closure) and $? \hat{\text{?}}$ (noninformative closure):

\[ \begin{align*}
\lambda p. \langle \text{E walk} \rangle_{\tau} & = \lambda P \langle e, T \rangle. P(j) \\
\lambda p. \langle \text{E walk} \rangle_{\tau} & = \lambda P \langle e, T \rangle. P(j)
\end{align*} \]
\[
(13) \quad ? \overset{\text{def}}{=} \lambda P. P \cup \neg P \quad \text{type } \langle \tau, \tau \rangle \\
(14) \quad ! \overset{\text{def}}{=} \neg \circ \neg \quad \text{type } \langle \tau, \tau \rangle 
\]

- We assume that any assertion must contain ! at its root.

- This has the following effect (Ciardelli, Groenendijk, and Roelofsen, 2013): Where \( A \sqcup B \) denotes the set of all states that entail \( A \) or entail \( B \), \(! (A \sqcup B)\) denotes the set of all states that entail \( A \lor B \), including those that do not entail one of the disjuncts.

\[
(15) \quad \llbracket \llbracket ! [\text{John or Mary walk}] \rrbracket_g \rrbracket_g \\
= \lambda p. p \subseteq \lambda w. \text{walk}(j)(w) \lor \text{walk}(m)(w) 
\]

- Finally, we can define inquisitive quantifiers:

\[
(16) \quad a. \quad \exists x \phi \overset{\text{def}}{=} \lambda p. \exists x \phi(p) \\
 b. \quad \forall x \phi \overset{\text{def}}{=} \lambda p. \forall x \phi(p) 
\]

4 \textbf{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 \textit{interrogative nucleus}. The complement of \( Q \) is the \textit{abstract}.

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

- The abstract is of type \( \langle e^n, T \rangle \): e.g. \( \langle e, T \rangle \) for single-\textit{wh} questions, \( \langle e, eT \rangle \) for double-\textit{wh}-questions.

- We could naively assume that \textit{wh}-phrases like \textit{who} are identity functions:

\[
(18) \quad \text{e.g. } [\llbracket \text{who} \rrbracket_g \text{ in subject position } = \lambda P_{et} \lambda x_e. P(x) 
\]

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

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

\[
(19) \quad \text{Whom do you want Mary to invite?} 
\]

- Here, \textit{want} expects a proposition, so \textit{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):

(20)  a. Who thinks that who walks?
     b. Who will be offended if we invite whom?

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

(21)  a. Who walks? \( \sim [Q \ [1 \ [\text{who, walks}]]) \]
     b. Who loves whom? \( \sim [Q \ [1 \ [2 \ [\text{who, loves whom,}]]) \]
     c. Who thinks that who walks? \( \sim [Q \ [1 \ [2 \ [\text{who, thinks [that who, walks]]})] \]

• Sometimes the abstract will be noninquisitive:

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

• Sometimes it will be inquisitive:

(23)   Who walks or talks?
     a. \([Q \ [1 \ [\text{who, [walks or talks]}]]) \]
     b. \([[[1 \ [\text{who, [walks or talks]}]])] = \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:

(24)  a. ?\(\exists x. \text{walk}(x)\)
     b. ?\(\exists x \exists y. \text{love}(y)(x)\)

• 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:

(25)   ?\(\exists x. \text{walk}(x) \lor \text{talk}(x)\)

• 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:

(26)   ?\(\exists x. (\text{walk}(x) \lor \text{talk}(x))\)

• 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 !.

(27) Would you like coffee↑, or tea↓?
    ≠ Is it the case that you would like either coffee or tea?

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

(28) $\llbracket \text{who}_i \rrbracket_g = \lambda P_{(e,T)}!P(g(i))$

(29) type $\langle eT, T \rangle$

In nonsubject position, we resolve type mismatches by type-shifters on the verb (Hendriks, 1993).
5 The $Q$ operator

- The $Q$ operator maps abstracts to inquisitive propositions.
- As is well known, there are several relevant candidate propositions:

\[
\begin{align*}
\text{(30)} & \quad \langle e, eT \rangle \\
& \quad \lambda y \lambda x. \text{love}(x)(y) \\
& \quad \lambda x. \text{love}(x)(g(1)) \\
& \quad !\text{love}(g(2))(g(1)) \\
& \quad \lambda P_{(e, T)} !P(g(1)) \\
& \quad \lambda Q_{(e, T, T)} \lambda x. Q(\lambda y. \text{love}(y)(x)) \\
& \quad \lambda P_{(e, T)} !P(g(2)) \\
& \quad \lambda y \lambda x. \text{love}(y)(x) \\
\end{align*}
\]

\[
\begin{align*}
\text{(31)} & \quad \text{John knows who is tall.} \\
& \quad \text{a. John knows of some } x \text{ that } x \text{ is tall.} \\ & \quad \text{b. John knows of every tall } x \text{ that } x \text{ is tall.} \\ & \quad \text{c. John knows of every } x \text{ whether } x \text{ is tall.}
\end{align*}
\]

\[
\begin{align*}
& \quad \text{mention-some} \\
& \quad \text{weakly exhaustive} \\
& \quad \text{strongly exhaustive}
\end{align*}
\]

- Groenendijk and Stokhof (1984) take (31c) as basic, which makes it hard to model (31a) and (31b) (Heim, 1994; Beck and Rullmann, 1999).
- InqB takes (31a) as basic, so one can model (31b) and (31c) through additional operations (Theiler, 2014).
- We assume that exhaustification optionally takes place within the interrogative nucleus; the precise “flavor” of exhaustivity is determined higher up (Theiler, 2014).
- We base the meaning of $Q$ on the inquisitive existential $\exists$ 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 ? to noninquisitive meanings.)

\[ Q^\eta \]_g = \lambda P_{(e^n,T)} . \exists x^n . P(x^n) \quad \text{type } \langle e^nT, T \rangle

- Some special cases:

  a. \[ Q^\eta \]_g (for non-wh questions) = \lambda P_T . ?P \quad \text{type } \langle T, T \rangle
  
  b. \[ Q^1 \]_g (for single-wh questions) = \lambda P_{(e,T)} . ?P(x) \quad \text{type } \langle eT, T \rangle
  
  c. \[ Q^2 \]_g (for double-wh questions) = \lambda R_{(e,eT)} . ?P(x) ?R(x)(y) \quad \text{type } \langle (e,eT), T \rangle

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

\[ Q^\eta_{exh} \]_g = \lambda R_{(e^n,T)} . \lambda P . \forall q \subseteq p . (\lambda x^n . R(x^n)(q)) = \lambda x^n . R(x^n)(p) \quad \text{type } \langle e^nT, T \rangle

- Some special cases:

  a. \[ Q^\eta_{exh} \]_g (for non-wh questions)
     = \lambda P_T . ?P = \lambda P_T . P \lor \neg P \quad \text{type } \langle T, T \rangle
  
  b. \[ Q^1_{exh} \]_g (for single-wh questions)
     = \lambda P_{(e,T)} . \lambda P . \forall q \subseteq p . (\lambda x . P(x)(q)) = \lambda x . P(x)(p) \quad \text{type } \langle eT, T \rangle
  
  c. \[ Q^2_{exh} \]_g (for double-wh)
     = \lambda R_{(e,eT)} . \lambda P . \forall q \subseteq p . (\lambda y \lambda x . R(y)(x)(q)) = \lambda y \lambda x . R(y)(x)(p) \quad \text{type } \langle (e,eT), T \rangle

6 Which-questions

- Following Groenendijk and Stokhof (1984), we assume that which-questions interpret their noun in situ. (We give here a non-presuppositional account but we are working on a presuppositional extension.)

\[ \text{which}_i \]_g = \lambda N_{(e,T)} \lambda P_{(e,T)} . ![N(g(i)) \land P(g(i))] \quad \text{type } \langle eT, \langle eT, T \rangle \rangle

- As for who, in nonsubject position we resolve type mismatches by type-shifters on the verb.
(37) \[ \lambda p. \forall q \subseteq p. ((\lambda x. q \in ![\text{murderer}(x) \land \hat{\text{tall}}(x)]) = (\lambda x. p \in ![\text{murderer}(x) \land \hat{\text{tall}}(x)]) \]

\[
\begin{array}{ccc}
\lambda P_{(e,T)} \cdot \lambda p. \forall q \subseteq p. & (\lambda x. P(x)(q) = \lambda x. P(x)(p)) \\
\langle e, T \rangle & \langle e, T \rangle & 1 \\
\lambda x. ![\text{murderer}(g(1)) \land \hat{\text{tall}}(g(1))] & ![\text{murderer}(g(1)) \land \hat{\text{tall}}(g(1))] & \langle e, T \rangle
\end{array}
\]

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

- Following Groenendijk and Stokhof (1984) we have given what amounts to a symmetric account. Of course, we know this can’t be the whole story:

(38) From Higginbotham (1996):
   a. Which men are bachelors?
   b. #Which bachelors are men?

- We are currently studying presuppositional extensions of inquisitive semantics that would allow us to import accounts such as Rullmann and Beck (1998) that capture the contrast between these questions in terms of the presuppositions of the *which*-phrase.

References


