Some consequences of knowing who bought what

Paul Hagstrom
Johns Hopkins University
UPenn 653 Questions seminar, March 3, 1999

Getting to Chapter 6, sections 1 and 4... (about pair list readings of multiple questions)

I. The program (ch. 2 mainly)

1. a. Taroo-ga nani-o katta no?
   Taro-NOM what-ACC bought Q
   ‘What did Taro buy?’

b. Taroo nani-t Q katta   no  ?
   ‘1”

Historical evidence from Premodern Japanese shows Q in this internal position.
(See also Sinhala, Shuri Okinawan—modern languages that do this too)

2. tare-ka mata hanatitibana-ni omoi-identem
   Premodern Japanese
   who-Q againflower.orange-DAT remember-M
   ‘Who will again remember (me) at the time of the mandarin orange flower?’
   (Shin Kokin Wakash [1205]:3, Ogawa 1977:222)

Path effects: Movement of Q sensitive to “relativized minimality”-type effects.

3. a. ?? dareka-ga nani-o nomimasita ka?
   Japanese
   someone-NOM what-ACC drank Q
   ‘What did someone drink?’

   ka dareka  [ nani t_i ] nomimasita
     [ left = up ]

b. nani-o, dareka-ga t_i nomimasita ka?
   what-ACC someone-NOM drank Q
   ‘What did someone drink?’

   ka  [ nani t_i ], dareka  t_i nomimasita
     [ ]

The task: What’s the role of Q (ka, no) wh-words in the semantics of questions?

II. Getting to a proposal for [Q] and for [who] (ch. 5 mainly)

4. Taroo-ga nani-o katta no?
   Taro-NOM what-ACC bought Q
   ‘What did Taro buy?’

Target: [ What did Taro buy? ] = λp ∃x∈things. p = bought(Taro, x)
(at least to a first approximation—essentially Hamblin’s set)

How do we get there compositionally? Clues might come from:

5. Taroo-ga nani-ka-o katta.
   Taro-NOM what-Q-ACC bought
   ‘Taro bought something.’

Hamblin’s (1973) idea, adopted by Rullmann & Beck (1997) and then by me:

6. [ who ] = people_{context} = {John, Mary, … }
   cf. [ John ] = John

7. left ( [ who ] ) = left ( {John, Mary, … } )
   = { left (John), left (Mary), … } = {that John left, that Mary left, … }

8. FLEXIBLE FUNCTIONAL APPLICATION: (cf. Rullmann & Beck 1997)
   Where f and a are sisters, α and µ are types, [ f a ] is:
   (i) f(a) resulting type: µ
      for f type <αµ>, a type α
      function, argument
   (ii) λm ∃x [ m = f(x) ∧ a(x) ]
      resulting type: <µα>
      for f type <αµ>, a type <α> function, (arguments)
   (iii) λm ∃g [ m = g(a) ∧ f(g) ]
      resulting type: <µα>
      for f type <<αµ>α>, a type α. {functions}, argument
   (iv) λm ∃x [ m = g(x) ∧ f(g) ∧ a(x) ]
      resulting type: <µα>
      for f type <<αµ>α>, a type <α> {functions}, {arguments}

otherwise, if (i–iv) are inapplicable, FFA does not yield a meaning for [ f a ].
Incidentally—you don’t need Q to make a question. (Suppose it’s not there below)

Hole: If Kratzer is right about choice functions, this would be non-evidence—but then I’m already presupposing Kratzer’s wrong, so perhaps that’s harmless

We could derive this using the Hamblin-idea above, directly.

(9) Taroo-ga nani-o katta?
Taro-NOM what-ACC bought
‘What did Taro buy?’

(10) Taroo-ga nani-ka-o katta.
Taro-NOM what-Q-ACC bought
‘Taro bought something.’

Well, solve for Q ↓.

Notice: (10) isn’t a question.

[≡ nani] is a set, but it isn’t triggering FFA.

So, [≡ katta] is not getting a set of arguments, just a plain argument.

Proposal: ka is an existential quantifier over choice functions.

It takes a set (type <et>) and returns an individual (type e)—a choice function. (plausibly) The meaning of (10) is (with this choice function thing in mind):

(12) $\exists f . \text{ bought } (\text{Taro}, f (\equiv \text{nani}))$

‘There is a function $f$ such that Taro bought the element $f$ picks from $\equiv \text{nani}$.’

‘There is an $x$ you can pick from $\equiv \text{nani}$ such that Taro bought $x$.’

Possible alternative:

[έ€] is responsible for turning the set $\equiv \text{nani}$ into a single argument for $\equiv \text{katta}$.

Syntactic assumption about the landing site of ka in questions.

ka moves (clitic-like?) to adjoin to $\text{C interrogative}$—forming a constituent.

Proposal: $\text{C interrogative}$ should take ka as its first argument, then the IP as its second argument, and should return a set of propositions.

[έ€] $\exists p . f (\equiv \text{fp})$ for some $f$.

somewhat inelegant, uses vacuous $\lambda w$ to get T from tautology, F from contradiction)
We do not fear sets, ontologically. (\(p = \langle st,pt,\ldots\rangle\))
So, how about: A PL question is a set of questions \(\langle pt,t\rangle\)? So:

(25) **Multiple Question Recognition**  
If the semantic value of an utterance is of type \(\langle pt,t\rangle\) (a set of questions), 
then the utterance is a (pair-list multiple) question. 
To respond: For each member set \(A\), respond to \(A\).  
(via SQR)

Ok, so we have an idea about • single questions, • PL multiple questions.
How about SP multiple questions? They **behave** like single questions. You pick one proposition, altogether.
So we’ll suppose they’re \(\langle pt\rangle\) and handled by Single Question Recognition.

Though I have no real explanation for the English fact, we suppose that

\[ \text{Who just bought what?} \equiv \{ p : p = \text{that } x \text{ just bought } y \text{ for } x \text{ a person } y \text{ a thing} \}\]

A set of propositions, a “Single Question,” pick one and you specify a pair.

### III. Ok, now we’re ready for Chapter 6. Part I: The Armchair

(22) Q. Who bought what? 
   A. ?? John bought a book. 
   A’. ??SP

(23) Q. Who just bought what? 
   A. ?? John just bought a book. 
   A. ??PL 
   A’. SP

**Armchair theorizing**—What ought to be the difference between SP and PL questions? 

\[ [\text{Who left?}] \equiv \{ \text{that John left, that Mary left, that Bill left, that Sue left, \ldots} \} \text{ \langle pt\rangle} \]

(24) **Single Question Recognition**  
If the semantic value of an utterance is of type \(\langle pt\rangle\) (a set of propositions) 
then the utterance is a single question. 
To respond: (a) one proposition from the set is selected, 
or (b) the presupposition (that there is an answer) is denied.

The PL meaning of *who bought what?* seems to be a lot like:

**What did John buy?**  
**What did Sue buy?**  
**What did Bill buy?**  
**What did Mary buy?**  
‘For everyone \(x\) who is contextually relevant, what did \(x\) buy?’

---

**Take-home point:**  

\(ka\) is responsible for **existential quantification over choice functions.** 
\(C_{\text{pl-recognize}}\) is responsible for abstraction over propositions.
(31) \[ \text{katta nani} \{ \text{dare} \} = \{ \text{bought-candy}, \text{bought-gum}, \ldots \} \{ \text{dare} \} \]  

\[ \text{FFA (iv)} \Rightarrow = \{ \text{bought-candy} \{ \text{John} \}, \text{bought-candy} \{ \text{Mary} \}, \ldots, \text{bought-gum} \{ \text{John} \}, \text{bought-gum} \{ \text{Mary} \}, \ldots \} \]  

\[ = \{ \text{that John bought candy, that Mary bought candy, \ldots, that John bought gum, that Mary bought gum, \ldots} \} \]  

'the set of propositions like \( y \) bought \( x \) for \( x \) a thing, \( y \) a person.'

That's what we expect as the representation for a SP multiple question.

Pick one, you specify a pair.

All is well.

Ok, now for (26a). (26a) has Q, and a PL reading. (26a) needs to be a set of questions.

(26a) \[ \text{dare-ga nani-o katta no?} \]  

\[ \text{Japanese} \]

\[ \text{who-NOM what-ACC bought Q} \]

'Who bought what?' (PL, SP)

Q contributes "\( \exists \! f \)" so we have a choice function—that's what is different.

What does \( \exists \! f \) do?

In simple single questions it, together with \( C_{\text{interrogative}} \) provides a set of propositions.

We can also get sets of propositions in single questions just with FFA.

~FFA: If an argument \( x \) gives you a representation of type \( \sigma \),

using \( \{ x_1, x_2, x_3, \ldots \} \) as that argument gives you a representation of type \( \sigma \).

\begin{tabular}{|l|}
\hline
\textbf{Idea:} & So, if we had a question with an individual argument  \\
& (e.g. with \textit{What did John buy?})  \\
& We can get a set of those things (that is, a set of questions) by replacing  \\
& the argument with a set.  \\
& (e.g. in \textit{Who bought what?})  \\
\hline
\end{tabular}

That is, we can get a set of questions (a PL question) using \textbf{both} means of getting sets:

- Flexible Functional Application
- \textit{kaz} + \( C_{\text{interrogative}} \)

Let's try it. Replace \textit{Taroo} with \textit{dare}…

(32) \[ \text{Taroo-ga nani-o katta no?} \]  

\[ \text{Japanese} \]

\[ \text{Taro-NOM what-ACC bought Q} \]

'What did Taroo buy?'

\[ \{ \text{that Taroo bought } \alpha, \text{that Taroo bought } \beta, \ldots \} \text{ if things}_{\text{conex}} = \{ \alpha, \beta, \ldots \} \]

(33) \[ \text{dare-ga nani-o katta no?} \]  

\[ \text{who-NOM what-ACC bought Q} \]

'Who bought what?'

We expect a set of questions like \textit{what did \( x \) buy?} for the \( x \)'s in \textit{[ dare ]}.

\[ \{ \{ \text{that } \alpha \text{ bought } \alpha, \text{that } \beta \text{ bought } \beta, \ldots \} \} \text{ if things}_{\text{conex}} = \{ \alpha, \beta, \ldots \} \text{ and people}_{\text{conex}} = \{ A, B, \ldots \} \]

This of course is what we’d hoped for—that’s the pair-list question \textit{who bought what?}

A technical obstacle: To do this, we need to specify how we interpret (34):

(34) \[ \lambda \! f \cdot \{ \text{Taro bought } f(\text{things}_{\text{conex}}), \text{Hanako bought } f(\text{things}_{\text{conex}}), \ldots \} \]

What we want is for this to work just like FFA—that is, we want it to yield:

(35) \[ \{ \lambda \! f. \text{Taro bought } f(\text{things}_{\text{conex}}), \lambda \! f. \text{Hanako bought } f(\text{things}_{\text{conex}}), \ldots \} \]

So we need to state a rule to handle this.

(36) \[ \text{FLEXIBLE } \lambda \text{-ABSTRACTION} \]

\[ [ \lambda \! x . \Phi ] \approx \lambda \! A. \exists! \phi \in \Phi. \text{A} = [ \lambda \! x . \phi[^{x=A}] ] \text{ for some } x \text{ not occurring in } \phi. \]

where \( \Phi \) is a set (type <\text{set}>), b) the result is composable.

This will get us from (34) to (35).

Incidentally, (26a) (=33) also has a SP reading.

Suppose syntactically in this case Q moves from someplace outside both \textit{wh}-words.

(37) \[ \text{[ dare nani katta ] t}_0 \text{ Q} \]

Then FFA will yield a set of propositions \( x \) bought \( y \) below \( t_0 \), and the choice function introduced by Q will choose among them. \( \lambda \! a. \exists! \phi. a = f(\Phi) \) characterizes A (like \( \lambda \! a. \text{as A} \)).
V. Answerhood—boarding up a back door

(38) **Single Question Recognition**
If the semantic value of an utterance is of type <pt> (a set of propositions) then the utterance is a single question.
To respond: (a) one proposition from the set is selected, or (b) the presupposition (that there is an answer) is denied.

Which one do you choose?
Unless the answers are exhaustive & mutually exclusive (à la Groenendijk & Stokhof 1984, Hamblin 1958), we can’t just say “the true one.”

A potential problem case: **Who** can be plural
So **Who left?** can be answered **John, Bill, and Mary left.**

Assuming that’s a single question and we picked one proposition, the proposition

(39) \[ \text{that John}\bar{\cap}\text{Bill}\bar{\cap}\text{Mary left} \]

must have been an option—in the set of propositions from which we picked.
Easy enough:

(40) \[ [ \text{Who } ] \text{ contains individuals and is closed under } \ominus. \]
\[ [ \text{who } ] = \{ \text{John, Mary, Bill, John}\bar{\cap}\text{Mary, John}\bar{\cap}\text{Bill, Bill}\bar{\cap}\text{Mary, … } \} \]

So a proposition of the form \( \text{that John}\bar{\cap}\text{Bill}\bar{\cap}\text{Mary left} \) is of the form \( x \text{ left for } x\in[\text{who }] \).

Now, if we ask **Who left?** and it is true that John, Bill, and Mary left, we want to pick:
\( \text{that John}\bar{\cap}\text{Bill}\bar{\cap}\text{Mary left} \)
even though
\( \text{that John left} \)
\( \text{that Bill left} \)
\( \text{that Bill}\bar{\cap}\text{Mary left} \)
…
are all true as well.

(41) \[ \text{answer}(Q)(w) = \text{the unique } p \in Q \text{ such that} \]
\[ p(w) \text{ and } \forall q \in Q, q(w) \rightarrow p \subseteq q \]

The one proposition in the set of propositions which is:
• true
and • which entails all the other true propositions.

This doesn’t mistakenly license lists in cases where we want only single-pair answers because there isn’t an entailment relationship between the members of the list, so the uniqueness of answer isn’t satisfied if more than one pair is true.

(42) \[ [ \text{Who bought what } ] = \{ \text{John bought gum, Mary bought coffee, John}\bar{\cap}\text{Mary bought gum}\bar{\cap}\text{coffee, … } \} \]
\( \text{John}\bar{\cap}\text{Mary bought gum}\bar{\cap}\text{coffee} \) does not imply that \( \text{John bought gum} \)
and \( \text{Mary bought coffee} \).
And there is no proposition in there like \( \text{John bought gum and Mary bought coffee} \), (which is what would be needed to get something like a list out)

Essentially: “Pick the maximally informative answer from the set Q”

Note: answer1 (Heim 1994, Beck & Rullmann 1996, 1997) does not have the right effect. Answer1 will let lists “in the back door.”
\[ \text{answer1}(Q)(w) = \ominus(Q(w)) \]

Intersecting the true propositions \( \text{John bought gum and Mary bought coffee} \) will give you the proposition \( \text{John bought gum and Mary bought coffee} \) which wasn’t in the Q set to begin with. To express that proposition, you have to say a list (“John bought gum and Mary bought coffee”)—leaving no way to restrict a multiple question to a SP reading.

VI. Paving the road to hell

I’ve been pretty cavalier about my (non-)use of intensionality, and there are possibly serious & difficult questions that need to be addressed.

Reinhart 1997, in her discussion of choice functions, says choice functions are drawn from G. These are functions which, given an intensional predicate yields an individual of which the predicate is true in the utterance world. This is purportedly needed for **Who wants to marry which millionaire?** where millionaires are not in want-worlds but in the actual world. Notice—the evaluation world is non-local, so we can’t just encode it in the denotation of *which millionaire* (unless it’s always the actual world—and I bet it isn’t).

(43) \[ G = \{ f: \forall P_{\subseteq\text{who}}, [f(P) \in P(w_0) ] \} \]

But \[ \text{who } \] wasn’t intensional the way I defined it. And if \[ \text{who } \] is of type \( <s,et> \) instead of \( <et> \), then it changes the character of FFA.
(I cannot claim to have fully absorbed the handout, but) Romero (1998; SALT 8) argued that Reinhart’s G is insufficient, and we need to allow for intensional choice functions of type \(<<\text{se},t>\)\text{,se}>, that is a straightforward choice function on individual concepts.

Just looking at the type, it seems like we might be able to maintain the system I was suggesting but say that \{\text{ who}\} does not denote a set of individuals, but rather a set of individual concepts. It makes things a little harder to think about, but it might be closer to right?

Perhaps fruitful discussion occurs here…

Would this also address the *Which subway line goes to the airport?* case?

What, after all, is the concept of *The Red Line*?

[That is, what unifies it across possible worlds—where it might be *Blue* in some worlds, or even not exist? Maybe transparency will avoid the problem here…?]

VII. Long-distance list readings (section 6.4)

Dayal (1996) points out that the Baker-ambiguities arise in a “wh-triangle” configuration.

(44) Who knows where we bought what?
   a. John knows where we bought what.
   b. John knows where we bought the beer, Mary knows where we bought the paper goods, and Bill knows where we bought the chips.

   [ Wh … [ wh .. wh ] ]

Same in other languages (Japanese, Bulgarian)—Wh-triangles permit long-distance lists. Embedded clause has to be a multiple question, not a single question.

Likely: embedded pair-list reading is responsible for this in some way.

Observe: If SP and PL are different (<pt>, <pt,t>) question embedding verbs can take either.

(45) a. Dale knows who killed Laura Palmer.  \text{know}_1, embeds <pt>
   b. Dale knows which deputy likes which donut. \text{know}_2, embeds <pt,t>

What would happen if you use \text{know}_1 in a question embedding a PL question? Well, FFA would apply.

(46) \text{know}_1, ( \{ \text{Where did we buy } \alpha? , \text{Where did we buy } \beta? , \ldots\} )

= \{ \text{know}_1(\text{Where did we buy } \alpha?) , \text{know}_1(\text{Where did we buy } \beta?) , \ldots\} 

Combine this with a subject wh-word wrapped with Q (the choice function), you get:

(47) \{ \text{Who knows where we bought } \alpha? , \text{Who knows where we bought } \beta? , \ldots\} 

So, we have a different question to answer for each thing bought. So, we should have to provide a pairing for each bought thing—and that seems to be right (as Dayal in fact observes).

Two complications:

One  • Declarative complementizers seem to stop FFA propagation.

(48) John knows where we bought what
cannot be a question. Matrix declarative is incompatible with sets of propositions.

(49) Who said that John knows where we bought what?
   [no LDL reading]

   Dayal: QR is clausebound.
   Here: Declarative clauses stop FFA propagation.

Incidentally, maybe plurality propagation stops at clauses too.

Pluralization is kind of like FFA

\text{left}(\text{John@Bill}) is true iff \text{left}(\text{John}) is true and \text{left}(\text{Bill}) is true.

[Beck (1998, SALT 8): different depends on partitions of pluralities which may not be able to penetrate clause boundaries:

* Brett and Karen said that John read books that came to different conclusions.
  * Brett and Karen read books that came to different conclusions.

If this is crosslinguistically true (we’d hope) it predicts that you can’t drop Q in a Japanese matrix question if the wh-word is embedded. I bet that’s false.

Two  • Relied on wrapping subject wh-word with Q.

Perhaps in general Q cannot be merged with an embedded clause complement.

So the subject is the first applicable wh-word to be merged?

Background assumption: Q usually merges with the lowest wh-word.

[I interpret that as “merge Q as soon as possible” on a bottom-up derv.]

VIII. So…?

• PL questions a) are sets of questions, b) arise using both FFA and “\exists!”+C\text{indefinite},
• Matches up nicely (and not by accident) with the proposed syntax of Q-movement.
• Lots of remaining issues…