Some consequences of knowing who bought what

Syntax/Semantics of Questions, March 23, 1999

Purportedly, this is about Hagstrom (1998), §5.5, §6.1, §6.4. But first, getting to that…

I. The program (ch. 2 mainly)

(1) a. Taroo-ga nani-o katta no? [reminder: no = $ka$]
   Taro-NOM what-ACC bought Q
   "What did Taro buy?"

(2) kimi-wa [ dare-ga kaita hon-o ] yomi-masi-ta ka ?
   you-TOP who-NOM wrote book-ACC read-POL Q
   'Who did you read books that $t$ wrote?' (Nishigauchi 1990:40)

(3) kimi-wa [ dare-ga kaita hon-o ] t ka yomi-masi-ta ka ?

Evidence included: • Premodern Japanese, Sinhala, Shuri Okinawan show Q
in its (purportedly) pre-movement position
• These languages respect islands between Q and the edge.
• "Intervention effects" along movement path (e.g., *dareka)

Evidence included: • Intervention effects go away when you embed in an island.
• Sinhala, Premodern Japanese, Shuri Okinawan Q appears (on the surface) just outside the island.

The task: What's the role of Q (ka, no) in the semantics of questions?

II. Getting to a proposal for [ $Q$ ] and for [ who ] (ch. 5, §§1–4)

(This was stuff that was more or less in my Swarthmore job talk, but worth doing again? )

(4) Taroo-ga nani-o katta no? Japanese
   Taro-NOM what-ACC bought Q
   'What did Taro buy?'

Target: $[\text{What did Taro buy?}] = \lambda p \exists x e \text{things. } p = \text{bought}(\text{Taro}, x)$
(at least to a first approximation—essentially Hamblin’s set)

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

(5) [ who ] = people$_{\text{ont}}$ = {John, Mary, … } cf. [ John ] = John

(6) left ( [ who ] ) =
   = [ left ( John, Mary, … ) ]
   = [ that John left, that Mary left, … ]

(7) FLEXIBLE FUNCTIONAL APPLICATION: (cf. Rullmann & Beck 1997)
Where $f$ and $a$ are sisters, $\alpha$ and $\mu$ are types, $f\alpha$ is:

(i) $f(a)$ resulting type: $\mu$
   for $f$ type $\langle \alpha\mu \rangle$, a type $\alpha$
   function, argument

(ii) $\lambda m \exists x [ m = f(x) \land a(x) ]$
   resulting type: $\langle \mu \rangle$
   for $f$ type $\langle \alpha\mu \rangle$, a type $\langle \alpha \rangle$
   function, (arguments)

(iii) $\lambda m \exists g [ m = g(a) \land f(g) ]$
   resulting type: $\langle \mu \rangle$
   for $f$ type $\langle \langle \alpha\mu \rangle, \alpha \rangle$, a type $\alpha$.
   (functions), argument

(iv) $\lambda m \exists g [ m = g(x) \land f(g) \land a(x) ]$
   resulting type: $\langle \mu \rangle$
   for $f$ type $\langle \langle \alpha\mu \rangle, \alpha \rangle$, a type $\langle \alpha \rangle$
   (functions), {arguments}

otherwise, if (i–iv) are inapplicable, FFA does not yield a meaning for $f\alpha$.

Incidentally—you don’t need Q to make a question. (Suppose it’s not there below)

(8) Taroo-ga nani-o katta ? Japanese
   Taro-NOM what-ACC bought Q
   'What did Taro buy?'

We could derive this using the Hamblin-idea above, directly.
So then what’s Q doing?

(9) Taroo-ga nani-ka-o katta. Japanese
   Taro-NOM what-Q-ACC bought Q
   'Taro bought something.'

Well, solve for [ $Q$ ].

Notice: (9) 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.

\[ \text{ka} = \lambda p, \lambda w \exists f : p, f(w) \]  

where \( p \) is a proposition-but-for-a-choice-function \(<<\text{et,e}>>\)  
\( w \) is a possible world \(<\text{et,e}>>\), \( f(D) \in D \).  
so \( \text{[ka]} \) itself is of type \(<<\text{et,e}>,<\text{et,e}>>>, \)  
or, more perspicuously, \(<\text{cp,p}>>\)  
(\( p \) is the type of a proposition \(<\text{et,e}>>\))  
and \( c \) is the type of a choice function \(<\text{et,e}>>\))

Possible alternative:  choice function variable bound by \( \exists \)-closure at some point.  
But:  QVE evidence that nanika acts like hotondo-no ronbun and not like ronbun (i.e. like a quantifier not a variable).

\[ [\text{ka}]=\lambda p, \lambda w \exists f : p, f(w) \]

[Hole: If Kratzer is right about choice functions (i.e. they’re never bound, they’re just left as free variables), this would be non-evidence—then I’m already presupposing Kratzer’s wrong (by my definition of [ka]), so perhaps that’s harmless]

Type mismatch—FFA can’t help. This is the “quantifier in object position” problem. QR.

Take-home point:  ka is responsible for existential quantification over choice functions.  
\( \text{C_interrogative} \) is responsible for abstraction over propositions.

\[ \text{CP} \]

\[ \text{IP} \]

\[ \text{C_interrogative} \]

\[ \text{ka} \]

Compositional implication:  \( \text{C_interrogative} \) should take ka as its first argument,  
then the IP as its second argument,  
and should return a set of propositions.

\[ [\text{C_interrogative}]=\lambda Q \lambda p, \lambda w \exists f : p, f(w) \]

(somewhat inelegant, uses vacuous \( \lambda w \) to get \( T \) from tautology, \( F \) from contradiction)

(15)  
| a.  | \[ t_w = f \]  |
| b.  | \[ nani \equiv \text{things}_{\text{context}} \]  |
| c.  | \[ nani t_w = f(\text{things}_{\text{context}}) \]  |
| d.  | \[ \text{[katta]} = \lambda x \lambda y \lambda w. \text{bought}(y, x) \text{in} w \]  |
| e.  | \[ \text{[katta nani t_w]} = \lambda y \lambda w. \text{bought}(y, f(\text{things}_{\text{context}})) \text{in} w \]  |
| f.  | \[ \text{[katta nani t_w]} (\text{[Taroo]} ) = \lambda w. \text{bought}(\text{[Taroo]}, f(\text{things}_{\text{context}})) \text{in} w \]  |
| g.  | \[ \text{[katta nani t_w]} = \lambda y \lambda w. \text{bought}(\text{[Taroo]}, f(\text{things}_{\text{context}})) \text{in} w \]  |
| h.  | \[ \text{[ka]} = \lambda p, \lambda w \exists f : p, f(w) \]  |
| i.  | \[ \text{[ka]} = \lambda p, \lambda w \exists f : p, f(w) \]  |

'Somewhat inelegant, uses vacuous \( \lambda w \) to get \( T \) from tautology, \( F \) from contradiction'
We do not fear sets, ontologically. (p = \text{st}, \text{single question} = \text{pt}, \ldots)
So, how about: A PL question is a set of questions (\text{pt}, \text{t})? So:
\[
\text{á} = \text{fl} = \text{fw}. \text{bought} (\text{Taro}, f(\text{things context})) \text{ in w} \text{ <cp>}
\]
That is, (20c) = \{\text{that Taro bought} \alpha, \text{that Taro bought} \beta, \ldots\} 
if \text{things context} = \{ \alpha, \beta, \ldots\}

III. More flexible functional application—islands
\underline{§5.5}

(I actually motivated FFA via the island structure in (3), not with the \text{ka-drop} case)

(3) kimi-wa [dare-ga kaita hon-o] tsa, yomi-masi-ta \text{ka} ?

The point: \text{if ka} isn’t moving from inside the island (but caveat: ch. 4, 8), the choice function can’t help do composition. You need to compose \text{dare}, a set, on its own. Same deal, though. (See p. 141 if necessary).

IV. Ok, now Chapter 6. Multiple Questions Part I: The Armchair

(21) a. \[ \text{\{Taroo nani t\alpha} \text{ katta } \text{\}} = \lambda f \lambda w . \text{bought}'(\text{Taroo}, f(\text{things context})) \text{ in w} \text{ <cp>}
\]
b. \[ \text{\{ka \}} = \lambda p, \lambda w \exists f . \text{p}(f(w) \text{ <cp,p>}
\]
c. \[ \text{\{C\text{\}} = \lambda Q \lambda p, \lambda k . Q \lambda k \lambda w \left\{ \text{p=p(g)} \right\} \neq \emptyset \text{ <cp,p>, <cp,pt>}
\]
d. \[ \text{\{C\text{\alpha, ka \}} = \lambda p, \lambda k \lambda w \exists f . \text{p}\left( f(w) \right) \neq \emptyset \text{ <cp,pt>}
\]
e. \[ \left\{ (19) \right\} = \lambda p \exists f . \text{p} = \lambda w . \text{bought}'(\text{Taro}, f(\text{things context})) \text{ in w} \text{ <pt>}
\]

Armchair theorizing—What ought to be the difference between SP and PL questions?
\[ \text{[Who left?]} = \left\{ \text{that John left, that Mary left, that Bill left, that Sue left, \ldots} \right\} \text{ <pt>}
\]

(22) Q. Who just bought what?
A. ?? \text{John just bought a book, Sue just bought a record, …} 
\text{??SP}
A'. ?? \text{John just bought a book.} 
\text{??PL}

(23) \underline{Single Question Recognition}

If the semantic value of an utterance is of type \text{<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.

The PL meaning of who bought what? seems to be a lot like:
\text{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?’

V. Chapter 6… Multiple Questions Part II: The effect of Q on PL readings

(25) a. \[ \text{dare-ga nani-o katta no?} \text{ Japanese}
\]
\text{who-NOM what-ACC bought Q} \text{ ‘Who bought what?’ (PL, SP)}

b. \[ \text{dare-ga nani-o katta ?} \text{ Japanese}
\]
\text{who-NOM what-ACC bought Q} \text{ ‘Who bought what?’ (*PL, SP)}

Start with (25b), Look, no Q. No choice function. Wh-words are sets. We’ll need FFA.
Let’s see how it evaluates under FFA:

(26) a. \[ \text{\{nani\} = things context} = \{ \text{candy, gum, coffee, soda, milk, \ldots} \} \text{ <pt>}
\]
b. \[ \text{[dare} \} = \text{people context} = \{ \text{John, Mary, Bill, Sue, \ldots} \} \text{ <pt>}
\]

(27) Compositionally:
\[
\begin{array}{c}
\text{dare} \\
\text{nani} \\
\text{katta}
\end{array}
\]

(28) \[ \text{katta} = \lambda x \lambda y . y \text{ bought x} \text{ (type <e,<et>> or so)}
\]

(29) \[ \text{[katta nani]} = \text{[katta]([nani])} \text{ FFA (iii) \implies}
\]
\[ = \{ \text{[katta](candy), [katta](gum), \ldots} \}
\]
\[ = \lambda P \exists x \text{ things context} . P = \text{[katta]}(x) \]
‘the set of predicates like bought x for x a contextually relevant thing.’
(30) \[ \text{katta nani } (\text{[ dare ]}) = \{ \text{bought-candy', bought-gum', ... } \} (\text{[ dare ]}) \]

\[ \text{FFA (iv) } = \{ \text{bought-candy' ([ John ]), bought-candy' ([ Mary ]), ... } \}
\text{bought-gum' ([ John ]), bought-gum' ([ Mary ]), ... } \]
\[ = \{ \text{that John bought candy, that Mary bought candy, ... } \}
\text{that John bought gum, that Mary bought gum, ... } \]

'the set of propositions like \text{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 (25a). (25a) has Q, and a PL reading. (25a) needs to be a set of questions.

(25a) \text{dare-ga nani-o katta no?}
\text{who-NOM what-ACC bought Q}
'Who bought what?' (PL, SP)

Q contributes \text{"∃f"}, so we have a choice function—that’s what is different.
What does \text{∃f} do? In simple single questions it, together with \text{C_interrogative} provides a set of propositions.

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

~FFA: If an argument \text{x} gives you a representation of type \text{σ},
using \{x, x, x, ... \} as that argument gives you a representation of type \text{σt}.

| Idea: So, if we had a question with an individual argument (e.g. with \text{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 \text{Who bought what?}) |
|---|

That is, we can get a set of questions (a PL question) using both means of getting sets:
• Flexible Functional Application
• \text{ka+ C_interrogative}.

Let’s try it. Replace \text{Taroo} with \text{dare}...

(31) \text{Taroo-ga nani-o katta no?}
\text{Taro-NOM what-ACC bought Q}
'What did Taroo buy?'

\{ \text{that Taroo bought } \alpha, \text{ that Taroo bought } \beta, ... \} \text{ if things}_{\text{context}} = \{ \alpha, \beta, ... \}

(32) \text{dare-ga nani-o katta no?}
\text{who-NOM what-ACC bought Q}
'Who bought what?'

We expect a set of questions like \text{what did x buy?} for the \text{x}'es in \text{[ dare ]}.

\{ \{ \text{that A bought } \alpha, \text{ that A bought } \beta, ... \}, \{ \text{that B bought } \alpha, \text{ that B bought } \beta, ... \}, ... \}
\text{if things}_{\text{context}} = \{ \alpha, \beta, ... \} \text{ and people}_{\text{context}} = \{ A, B, ... \}

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

A technical obstacle: To do this, we need to specify how we interpret (33):

(33) \text{λx . \{ Taroo bought f(things}_{\text{context}}, Hanako bought f(things}_{\text{context}}, ... \}}

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

(34) \{ \text{λx.Taroo bought f(things}_{\text{context}}, λx.Hanako bought f(things}_{\text{context}}, ... } \}

So we need to state a rule to handle this (note: this is revised, better...)

(35) \text{FLEXIBLE λ-ABSTRACTION}
\[ [ \lambda x . \Phi ]^{t_0} = \lambda A \exists \phi \in \Phi . A = [ \lambda x . [ \Phi ]^{t_0 x} ] \]
for some \text{x} not occurring in \phi.

where a) \Phi is a set (type <m't>), b) the result is composable.

This will get us from (33) to (34).

Incidentally, (25a) (=32) also has a SP reading.
Suppose syntactically in this case \text{Q} moves from someplace outside both \text{wh}-words.

(36) \text{[ dare nani katta ]}_{t_0} \text{ Q}

Then FFA will yield a set of propositions \text{x bought y} below \text{t}_0, and the choice function introduced by \text{Q} will choose among them. \text{λa∃x.a=f(A)} characterizes \text{A} (like \text{λa.a∈ A}).
And, check this out: (it was hidden in chapter 3)

(37) Taroo-ga [dare-ga nani-o katta toki-ni] okotta no?
Taroo-NOm who-NOm what-ACC bought when got.angry Q
‘Taroo got angry when who bought what?’ (*PL, SP)

Just as we expect, if Q can’t be coming from “between” the wh-words, no PL reading.

VI. Long-distance list readings

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

(38) 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. Here’s my story:

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

(39) a. Dale knows who killed Laura Palmer. know₁ embeds <pt>
   b. Dale knows which deputy likes which donut. know₂ embeds <pt,t>

What would happen if you use know₁ in a question embedding a PL question? Well, FFA would apply.

(40) know₁, { (Where did we buy α?, Where did we buy β?, …) }
    = { know₁(Where did we buy α?), know₁(Where did we buy β?), … }

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

(41) { Who knows where we bought α?, Who knows where we bought β?, … }

So, we have a different question to answer for each bought thing. 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.

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

(43) 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 left (John@Bill) is true iff left(John) is true and left(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.]

A probably-false prediction:
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.

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.]

VII. Answerhood—boarding up a back door

(We talked about this a little bit earlier on, but I forget in what context…)

(44) 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?
(And remember, proposition # utterance, proposition # sentence)

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

\( (45) \quad \text{that John} \bar{\text{Bill}} \bar{\text{Mary left}} \)

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

Easy enough:

\( (46) \quad [ \text{Who} ] \text{ contains individuals and is closed under } \oplus. \)

\[ [ \text{who} ] = \{ \text{John, Mary, Bill, John} \bar{\text{Mary}}, \text{John} \bar{\text{Bill}} \bar{\text{Mary}}, \ldots \} \]

So a proposition of the form \( \text{that John} \bar{\text{Bill}} \bar{\text{Mary left}} \) 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:

\( \text{that John} \bar{\text{Bill}} \bar{\text{Mary left}} \)

even though \( \text{that John left} \)
\( \text{that Bill left} \)
\( \text{that Bill} \bar{\text{Mary left}} \)
\( \ldots \)

are all true as well.

\( (47) \quad \text{answer}(Q)(w) \) is the unique \( p \in Q \) 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:

\( \cdot \) true
\( \cdot \) 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.

\( (48) \quad [ \text{Who bought what} ] = \{ \text{John bought gum, Mary bought coffee, John+Mary bought gum+coffee, ...} \} \)

\( \text{John+Mary bought gum+coffee does not imply that John bought gum} \)
\( \quad \text{and Mary bought coffee.} \)

And there is no proposition in there like \( \text{John bought gum and Mary bought coffee,} \)
\( \quad \text{(which is what would be needed to get something like a list out)} \)

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

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

\[ \text{answer1}(Q)(w) = \cap(Q(w)) \]

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.

\textbf{VIII. 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: choice functions are drawn from \( G \)—given an intensional predicate yields an individual of which the predicate is true in the utterance world. \( \text{(Who wants to marry which millionaire?)} \text{ Notice—the evaluation world is non-local, so we can’t just encode it in the denotation of which millionaire} \) (unless it’s always the actual world).

\[ (49) \quad G = \{ f: \forall P_{\text{ext}}, \{ f(P) \in P(w_0) \} \} \]

But [ who ] wasn’t intensional the way I defined it. And if [ who ] is of type \( \text{<s,et>} \) instead of \( \text{<et>} \), then it changes the character of FFA.

Romero (1998; SALT 8) argued that Reinhart’s \( G \) is insufficient, and we need to allow for intensional choice functions of type \( \text{<<se,t>,se>} \), 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 [ 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?

\[ \text{But cf. the presumably prior discussion of the which millionaire cases too—i.e. doesn’t Which kid wants to meet which of Santa’s reindeer? allow the reindeer to exist only in the belief-worlds of the kids?} \]

\textbf{IX. So...?}

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