# Classics on Questions I:
# Hamblin and Karttunen

Stefan Kaufmann, Northwestern University
*Questions and Inquisitive Semantics*

Kyoto University, September 20, 2009

## 1 Hamblin (1958)

This early essay on some fundamental issues in the semantics of questions contains many highly influential ideas.

### 1.1 Postulates

1. An answer to a question is a statement.
2. Knowing what counts as an answer is equivalent to knowing the question.
3. The possible answers to a question are an exhaustive set of mutually exclusive possibilities.

Some comments.

#### 1.1.1 The form of answers

- In 1., "a statement" means "a declarative sentences." Thus Hamblin assumes that shorter answers (i.e., saying only your name when asked what your name is) are elliptical forms of full sentences.
- This is in line with Hamblin's overall "partitionist" perspective. What about the alternative view that answers are actually short and full-sentence answers are redundant?

  (The structured-meaning approach, where the question is a function that takes its answer as an argument, is more conducive to seeing the rest of the sentence as extraneous material, in line with the fact that it is *backgrounded* in terms of focus.)
- Not so fast: Different questions may have the same "short answers"; cf. (1) and (2a).

  (1) a. In which continent is Luxembourg?
      b. In which continent is Ecuador?

  If knowing that (2a) is the set of answers were all there is to characterizing the question, then (1a,b) would be equivalent. But not if the set of answers is (2b) instead of (2a).

  (2) a. {Europe, Asia, Africa, ...}
      b. {Luxembourg is in Europe, Luxembourg is in Asia, Luxembourg is in Africa, ...}

⇒ The denotation of a question like (1a,b) is not just the characteristic function of some set of continents.

(The structured-meaning approach needs to take that into account as well.)

### 1.1.2 The meaning of questions

- In (2), notice that "knowing what counts as an answer" is distinct from "knowing the answer." To know what the question means, you don't need to know what the answer is.

### 1.1.3 Partitions

- A "possibility" is a proposition (i.e., a set of possible worlds). Thus a question denotes a set of sets of possible worlds.

- Exhaustive in (3) means that the answers jointly exhaust the logical space of possibilities; this is *not* the same as as the "exhaustivity" of answers (see below).

- Hamblin calls (3) a "logically improper" question.

  (3) Have you stopped beating your wife?

  Both possible answers to (3) presuppose that the listener was beating his wife; hence neither is true at those worlds at which he wasn't.

  Alternatively, Hamblin suggests, whether a question as a "proper" one may always be *relative* to some particular set of possible worlds (e.g., the union of its possible answers). He doesn't take that route mainly for simplicity, but it is nowadays often taken:

  – Questions are evaluated relative to some given set of worlds (epistemic states or the common ground)
  – They are only felicitous if this set of worlds is contained in the union of their possible answers.

  Hamblin: A relative question (i.e., one whose answers to not exhaust the logical space) can be converted to a non-relative one by the addition of a "residual" answer — in our terms, one that denies the presupposition of the question, e.g., *'I never have been'* in (3).

- Mutually exclusive: This *is* related to the "exhaustivity" of answers we've been talking about.

  If John and Mary are the people under consideration, then according to Hamblin, (4b) is not an answer, but (4c) is.

  (4) a. Who has a copy of this paper?
  b. John.
  c. John does and Mary doesn't.

  What about semantically weaker answers like (5b)? Well, Hamblin doesn't say that they are not answers — they are just not *proper* answers.

  (5) a. In which continent is Luxembourg?
  b. Either Europe, or Asia, or Africa."

  *Proper* answers are *complete*, i.e., mutually exhaustive.

⇒ Seeds of late developments:

- Question denotations are sets of propositions.
- Distinction between *relative* and *non-relative* questions: The former are evaluated with respect to some restricted set of worlds. This is really the only case that occurs in actual linguistic usage.
- Distinction between *proper* (or *complete*) answers and others: Exhaustivity.

## 1.2 Theorems

Consequences of the above three axioms (keep in mind that he is considering only the case of non-relative questions and complete answers):

1. If a question has only one possible answer, that answer is a tautology.
2. If an any answer to a question is a tautology, it is the only possible answer.
3. Every question has an answer.

## 1.3 Logical relations

- One question *contains* (i.e., entails) another if every answer to the first entails an answer to the second.

  I.e., if one cannot give a complete answer to the first without thereby giving a complete answer to the second. Thus (6a) entails (6b).

  (6) a. What is the latitude and longitude of Ecuador's highest peak?
      b. In which continent is Ecuador?

- Two questions are equivalent if they contain (i.e., entail) each other.
- The *join* of two questions: the question that is asked when the two questions are asked together. Clearly it entails each of them.

  This works a bit like conjunction. But notice that the join of (7a) and (7b), which perhaps roughly corresponds to (7c), is *neither* (7d) *nor* (7e):

  (7) a. Did John read the paper?
      b. Did Mary read the paper?
      c. Did John read the paper and did Mary read the paper?
      d. Did John and Mary read the paper?
      e. Did John or Mary read the paper?

# 2 Hamblin (1973)

This is Hamblin's attempt to make his ideas work in Montague's semantic framework, which at the time was hot off the press. Early in the paper (through Section 7), Hamblin gives a highly readable introduction to some key ideas of Montague's.

## 2.1 Indicatives

Unfortunately, the version of the grammar Hamblin uses is an earlier one than that of PTQ, which Karttunen would later use in his paper. Let us consider a somewhat complex sentence, one with a quantifier:

(8) Every new student meets John.

In this system there is only one interpretation function, $D$, which assigns denotations to both constants and variables (no separate variable assignment $g$). I will write '$D[v/d]$' for the interpretation that differs from $D$ at most in that $D[v/d]_{prn}(v) = d$.

every new student meets John (6)
$D_{fml}(every\ new\ student\ meets\ John)$
$= \bigcap_{\{D'=D[v_0/d]|d\in Dom\}}\Big(-D'_{cmn}(new\ student)[D'_{prn}(v_0)] \cup D'_{fml}(v_0\ meets\ John)\Big)$

new student (5)
$D_{cmn}(new\ student)$
$= D_{adj}(new)[D_{cmn}(student)]$

new$_{adj}$ — $D_{adj}(new)$
student$_{cmn}$ — $D_{cmn}(student)$

$v_0$ meets John (2)
$D_{fml}(v_0\ meets\ John)$
$= D_{2vb}(meets)[D_{prn}(v_0), D_{prn}(John)]$

$v_0$ — $D_{prn}(v_0)$
meets$_{2vb}$ — $D_{2vb}(meets)$
John$_{prn}$ — $D_{prn}(John)$

What the resulting formula says: Given an individual $d$, take union of the set of worlds at which $d$ is not a new student with the set of worlds at which $d$ meets John. You get the set of worlds at which if $d$ is a new student, $d$ meets John. Do this for every $d$ in the domain of individuals. The sentence is true at just those worlds which pass the test for every $d$.

## 2.2 Questions

Consider the following simple question:

(9) Who meets John?

For questions, Hamblin jumps to a different treatment of transitive verbs: Instead of functions from pairs of individuals to propositions, they now denote functions from individuals to functions form individuals to propositions.[1] It's not clear why he doesn't do it this way from the outset. In any case, here is the derivation.[2]

[1] You may remember form your Semantics courses that this reformulation, which is now standard, is called "Currying" or "Schönfinkelization."

[2] In the last step of the derivation below, I use the notational convention that $\gamma(\beta)(\alpha) =_{df} \gamma(\alpha,\beta)$.

who meets John (1Q)
$E_{fml}(\mathit{who\ meets\ John})$
$= E_{1vb}(\mathit{meets\ John})$“‘$E_{prn}(\mathit{who})$
$= \{\alpha(\beta) | \alpha \in E_{1vb}(\mathit{meets\ John}) \wedge \beta \in E_{prn}(\mathit{who})\}$
$= \{D_{2vb}(\mathit{meet})[D_{prn}(\mathit{John})](x) | x \in E_{prn}(\mathit{who})\}$
$= \{D_{2vb}(\mathit{meet})[d_0, D_{prn}(\mathit{John})], D_{2vb}(\mathit{meet})[d_1, D_{prn}(\mathit{John})], D_{2vb}(\mathit{meet})[d_2, D_{prn}(\mathit{John})], \ldots\}$

who
$E_{prn}(\mathit{who})$
$= \{d_0, d_1, d_2, \ldots\}$

meets John (20)
$E_{1vb}(\mathit{meets\ John})$
$= E_{2vb}(\mathit{meet})$“‘$E_{prn}(\mathit{John})$
$= \{\alpha(\beta) | \alpha \in E_{2vb}(\mathit{meet}) \wedge \beta \in E_{prn}(\mathit{John})\}$
$= \{D_{2vb}(\mathit{meet})[D_{prn}(\mathit{John})]\}$

meet$_{2vb}$
$E_{2vb}(\mathit{meet})$
$= \{D_{2vb}(\mathit{meet})\}$

John$_{prn}$
$= E_{prn}(\mathit{John})$
$= \{D_{prn}(\mathit{John})\}$

## 2.3 Remarks

- Notice that the set of propositions derived above is not a partition (they are not mutually exclusive). This is in contrast with Hamblin (1958), but this time it is indeed what he intends:

> “We shall need to regard ‘who walks’ as itself denoting a set, namely, the set whose members are the propositions denoted by ‘Mary walks’, ‘John walks’, . . . and so on for all individuals. Pragmatically speaking a question sets up a choice-situation between a set of propositions, namely, those propositions that count as answers to it.” (p. 48)

- The treatment of quantifiers in this paper is clumsy. Try to see how the derivation for *'Who does every new student meet'* works (“every student meets who”). Or rather, don’t.
- The pointwise function application between sets introduced here (written $E_1$“‘$E_2$) has become influential in a number of areas (focus, the interpretation of pronouns, etc.) When you hear someone speak of “Hamblin-style” semantics, this is what they mean.
- Hamblin says nothing about embedded questions and the verbs that embed them.

# 3 Karttunen (1977)

Main topics of the paper:

- Unified account of *direct* and *indirect* (i.e., embedded) questions
- Classification of question-embedding verbs
- Distribution of interrogative pro-forms (*wh*-words) and their interaction with quantifiers

## 3.1 Syntax

- Categorial grammar: Linguistic expressions have syntactic categories. Two basic categories ($e$ and $t$) and several complex ones: '$A/B$' and '$A//B$' is the category of expressions which, when combined with an expression of category $B$, form an expression of category $A$. Notice that there are no linguistic expressions of type $e$ or $t$.

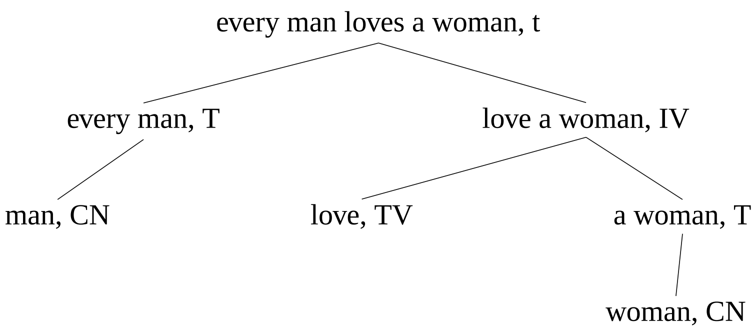
  Things below the line were newly introduced by Karttunen.

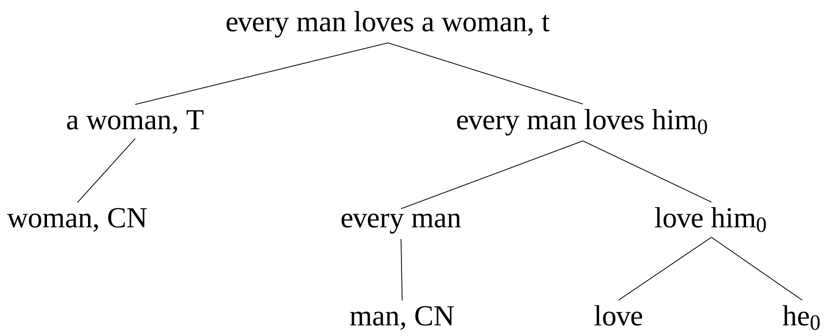
| *type* | *explanation* | *examples* |
|---|---|---|
| $e$ | "entity expressions" (individual expressions) | |
| $t$ | "truth value expressions" (declarative sentences) | |
| $IV = t/e$ | "intransitive verb phrases" (i.e., one-place predicates) | run, walk, rise |
| $T = t/IV$ | "terms" | John, Mary, $\text{he}_0$, $\text{he}_1$ |
| $TV = IV/T$ | "transitive verb phrases" | find, lose, love |
| $IAV = IV/IV$ | $IV$-modifying adverbs | rapidly, slowly, allegedly |
| $CN = t//e$ | common nouns | man, park, fish, unicorn |
| $t/t$ | sentence-modifying adverbs | necessarily |
| $IAV/T$ | $IAV$-making prepositions | in, about |
| $IV/t$ | "sentence-taking verb phrases" | believe that, assert that |
| $IV//IV$ | "$IV$-taking verb phrases" | try to, wish to |
| $Q = t//t$ | indirect questions | whether it is raining, who dates Mary |
| $IV/Q$ | question-embedding verbs | know, remember, tell, wonder |
| $WH = t//IV$ | WH-phrases | who, what, which man |

- Complex expressions are built and combined according to syntactic rules, the arcana of which will not concern us too much.

  One thing worth noting is the way different quantifier scopes are derived syntactically: Since every semantic difference comes from a syntactic difference, there are different ways of deriving sentences containing them.

(10) Every man loves a woman.






### 3.2 Semantics

- Type theory: Syntactic expressions are mapped to semantic objects of different types. Each syntactic type goes with one semantic type.

  - Basic types: $e, t$
  - if $a, b$ are types, then $\langle a, b\rangle$ is a type;
  - if $a$ is a type, then $\langle s, a\rangle$ is a type.
  - Model-theoretic interpretation of the types: $D_\tau$ is the is the set of possible denotations of type $\tau$.
    * $D_e$: individuals (universe of the model)
    * $D_t$: $\{0, 1\}$ (truth values)
    * $D_{\langle a,b\rangle}$: $D_b^{D_a}$ (functions with domain $D_a$ and range $D_b$)
    * $D_{\langle s,a\rangle}$: $D_a^W$
      (This is simplified. Montague has $D_a^{I\times J}$, where $I \times J$ are *world-time pairs*.

- Intensional Logic: Pretty much standard; but notice the following:

  - Expressions are generally evaluated relative to some possible world $w \in W$.
    (Montague would say, a world-time pair $\langle i, j\rangle \in I \times J$).
  - The *extension* of an expression at $w$ is its denotation at $w$.
  - The *intension* of an expression at $w$ is a function from posssible worlds to extensions.
  - If $\alpha$ is an expression of type $a$, then ${}^\wedge\alpha$ is of type $\langle s, \alpha\rangle$ (i.e., the intension of $\alpha$).
  - If $\alpha$ is an expression of type $\langle s, a\rangle$, then ${}^\vee\alpha$ is of type $a$ (i.e., the extension of $\alpha$)
  - Montague (and, following him, Karttunen) uses the symbols $\bigwedge$ and $\bigvee$ for $\forall$ and $\exists$, respectively.

- In general, we have the following:

  - $\langle \alpha, t\rangle$: (characteristic functions of) sets of things of type $\alpha$;
  - $\langle s, \langle \alpha, t\rangle\rangle$: properties of things of type $\alpha$
  - $\langle \alpha, \langle \beta, t\rangle\rangle$: relations with domain $D_\alpha$ and range $D_\beta$

Some conventions:

| *Type* | *Explanation* | *Variables* |
|---|---|---|
| $e$ | individuals | $u, v, \ldots$ |
| $\langle e, t\rangle$ | sets of individuals | |
| $\langle s, \langle e, t\rangle\rangle$ | properties of individuals | $M$ |
| $\langle s, e\rangle$ | individual concepts | $x, y, \ldots$ |
| $\langle\langle s, e\rangle, t\rangle$ | sets of individual concepts | |
| $\langle s, \langle\langle s, e\rangle, t\rangle\rangle$ | properties of individual concepts | $P, Q$ |
| $\langle s, t\rangle$ | sets of possible worlds (= propositions) | $p$ |
| $\langle\langle s, t\rangle, t\rangle$ | sets of propositions | $?p$ |

- Some more things:
  - Generally, the interpretation of an expression $\alpha$ is written $\alpha'$ (i.e., a function '$\cdot'$' maps expressions to meanings).
  - You see expressions of the form '$date'_*$' and so on in Karttunen's paper. The star means that the verb denotes a relation between individuals, which is probably you'd expect (without the star, '$date'$' denotes something more complex, a relation between individuals and properties of individual concepts).
  - Each syntactic rule is paired with a rule for semantic composition. Generally, functions apply to *intensions* (see below).
  - There is a lot more to be said about PTQ, but we won't go into it here.

- Type mapping: A function $f$ from syntactic types to semantic types, defined recursively:
  - $f(e) = e$
  - $f(t) = t$
  - $f(A/B) = f(A//B) = \langle\langle s, f(B)\rangle, f(A)\rangle$

  Recall that $A/B$ is an expression which, when combinec with an expression of type $B$, yields an expression of type $A$.

  Likewise, the denotation of an expression of type $A/B$ is a function which, when fed *the intension of* the denotation of a $B$, yields the denotation of an $A$.

  $\Rightarrow$ In particular, for questions: $f(Q) = f(t//t) = \langle\langle s, f(t)\rangle, f(t)\rangle = \langle\langle s, t\rangle, t\rangle$

  This is what Karttunen wants (a set of propositions).

## 3.3 Interpretation of questions

Karttunen derives three kinds of questions (alternative questions, yes/no questions, and wh-questions) from a common semantic ancestor: *Proto-questions*. Proto-questions are in turn derived from declarative sentences (accordingly, their denotations are derived from propositions).

### 3.3.1 Proto-questions

- Proto-question rule (p. 389): If $\phi$ is a declarative sentence, then $?\phi$ is a question.

  If $\phi$ translates to $\phi'$, then $?\phi$ translates to $\lambda p[{}^\vee p \wedge p = {}^\wedge\phi']$.

- Remarks:
  - At the world of evaluation, $\phi'$ refers to a truth value in {0,1}.

- $^\wedge\phi'$ refers to a function from possible worlds to truth values (i.e., the characteristic function of a proposition).

  $p$ is of the same type.
- $^\vee p$ is the extension of $p$, i.e., its truth value.

$\Rightarrow$ the whole expression is true of those propositions $p$ that are (i) true and (ii) equal to $\phi'$.

- Example: Let *'Mary cooks'* denote '$cook'_*(m)$'. Then

$$?Mary\ cooks = \lambda p[^\vee p \wedge p = {}^\wedge cook'_*(m)]$$

  This is the characteristic function of a set of propositions. This contains just the proposition that Mary cooks if she cooks, and it is empty if she doesn't cook.

#### 3.3.2 (Indirect) alternative questions

(11) a. whether Mary cooks or John eats out
 b. whether Mary likes John or Mary likes Bill

These are derived from proto-questions:

- Alternative question rule: If $?\phi_1, ?\phi_2, \ldots, ?\phi_n$ are questions, then *whether* $\phi_1$ *or* $\phi_2$ *or* ... *or* $\phi_n$ is a question.

  And it translates to $\lambda p[[?\phi_1]'(p) \vee [?\phi_2]'(p) \vee \ldots \vee [?\phi_n]'(p)]$.
- Example:

(12) whether Mary cooks or John eats out

whether Mary cooks or John eats out
$\lambda p[\lambda q[^\vee q \wedge q = {}^\wedge cook'_*(m)](p) \vee \lambda r[^\vee r \wedge r = {}^\wedge eatout'_*(j)](p)]$
$= \lambda p[[^\vee p \wedge p = {}^\wedge cook'_*(m)] \vee [^\vee p \wedge p = {}^\wedge eatout'_*(j)]]$

?Mary cooks
$\lambda q[^\vee q \wedge q = {}^\wedge cook'_*(m)]$

Mary cooks
$cook'_*(m)$

?John eats out
$\lambda r[^\vee r \wedge r = {}^\wedge eatout'_*(j)]$

John eats out
$eatout'_*(j)$

- Remarks:
  - Karttunen points out that the resulting expression is equivalent to

$$\lambda p[^\vee p \wedge [p = {}^\wedge cook'_*(m) \vee p = {}^\wedge eatout'_*(j)]]$$

    which is the characteristic function of a set of proposition: true of just those propositions in $\{cook'_*(m), eatout'_*(j)\}$ that are true (four possibilities: zero, one, or both).
  - Notice that in case Mary cooks and John eats out, the proposition $cook'_*(m) \wedge eatout'_*(j)$ does *not* end up in the denotation. Other accounts differ (e.g. von Stechow, 1991).
  - The presupposition that just one of the propositions is true is not accounted for.

- Regarding the Alternative question rule: Notice that there must be two or more proto-questions. This rule does *not* give us *'whether Mary cooks'*. The next rule does.
- Typo on page 390, second-to-last line: the first $\vee$ should be $\wedge$ (cf. the end of the preceding paragraph)

### 3.3.3 (Indirect) yes/no questions

(13) a. whether Mary cooks
 b. whether John eats out

- Yes/No question rule (p. 391): If $?\phi$ is a question then *whether* $\phi$, *whether or not* $\phi$, *whether* $\phi$ *or not* $\phi$ are also questions.

  And they all translate to $\lambda p[[?\phi]'(p) \vee [\neg\exists q.[?\phi]'(q) \wedge p = {}^\wedge\neg\exists q.[?\phi]'(q)]]$

- Example:

  (14) whether Mary cooks

whether Mary cooks

$$\lambda p[\underline{\lambda q[{}^\vee q \wedge q = {}^\wedge cook'_*(m)]}(p) \vee [\neg\exists r[\underline{\lambda q[{}^\vee q \wedge q = {}^\wedge cook'_*(m)]}(r)] \wedge p = {}^\wedge\neg\exists r[\underline{\lambda q[{}^\vee q \wedge q = {}^\wedge cook'_*(m)]}(r)]]]$$

$$= \lambda q[[{}^\vee q \wedge q = {}^\wedge cook'_*(m)] \vee [\neg\exists r[{}^\vee r \wedge r = {}^\wedge cook'_*(m)] \wedge q = {}^\wedge\neg\exists r[{}^\vee r \wedge r = {}^\wedge cook'_*(m)]]]$$

?Mary cooks

$$\lambda q[{}^\vee q \wedge q = {}^\wedge cook'_*(m)]$$

Mary cooks

$$cook'_*(m)$$

- Remarks:

  - The underlined parts are the three places in which the meaning of the proto-question shows up.
  - In words: "Either $p$ is the true proposition that Mary cooks, or the question denotation is empty (i.e., Mary doesn't cook) and $p$ is the proposition that the question denotation is empty (i.e., that the answer is *'no'*)."

  Q: Why is the second disjunct so complicated? Why not just refer to the complement of the proposition that Mary cooks?

  A: Because if Mary doesn't cook at the world of evaluation, that complement is not recoverable from the denotation of the proto-question. Recall that $\lambda q[{}^\vee q \wedge q = {}^\wedge cook'_*(m)]$ is (the characteristic function of) a set of propositions, nothing more. It is either singleton (if Mary cooks) or empty (if Mary doesn't). In neither case does it contain the negation of $cook'_*(m)$. But the fact that it is empty tells you that the answer is *'no'*.

  - The resulting expression is complicated, but equivalent to

$$\lambda q[{}^\vee q \wedge [q = {}^\wedge cook'_*(m) \vee q = {}^\wedge\neg cook'_*(m)]]$$

    because of what I just outlined above (together with additional assumptions such as the Law of Excluded Middle): The proto-question denotes the empty set iff $cook'_*(m)$ is false, in which case $\neg cook'_*(m)$ is true. I will use this notation below.

  - Typo on page 392, (28b): The hat on $\hat{p}$ should be round, as in (28a).

### 3.3.4 Question embedding

(15) know whether John walks

- Question embedding rule (p. 392): If $\delta$ is a quesetion-embedding transitive verb and $\phi$ is a question (not a proto-question!), then $\delta\phi$ is a verb phrase.

  Its translation is $\delta'({}^\wedge\phi')$.

- Example:

  (16) Bill knows whether John walks.

Bill knows whether John walks

$$\begin{aligned} &\lambda P[{}^\vee P({}^\wedge b)]({}^\wedge know'({}^\wedge\lambda p[{}^\vee p \wedge [p = {}^\wedge walk'_*(j) \vee p = {}^\wedge\neg walk'_*(j)]])) \\ &= [{}^\vee{}^\wedge know'({}^\wedge\lambda p[{}^\vee p \wedge [p = {}^\wedge walk'_*(j) \vee p = {}^\wedge\neg walk'_*(j)]])]({}^\wedge b) \\ &= know'({}^\wedge\lambda p[{}^\vee p \wedge [p = {}^\wedge walk'_*(j) \vee p = {}^\wedge\neg walk'_*(j)]])({}^\wedge b) \\ &\equiv know'({}^\wedge b, {}^\wedge\lambda p[{}^\vee p \wedge [p = {}^\wedge walk'_*(j) \vee p = {}^\wedge\neg walk'_*(j)]]) \end{aligned}$$

Bill

$\lambda P[{}^\vee P({}^\wedge b)]$

know whether John walks

$know'({}^\wedge\lambda p[{}^\vee p \wedge [p = {}^\wedge walk'_*(j) \vee p = {}^\wedge\neg walk'_*(j)]])$

know

$know'$

whether John walks

$\lambda p[{}^\vee p \wedge [p = {}^\wedge walk'_*(j) \vee p = {}^\wedge\neg walk'_*(j)]]$

?John walks

$\lambda q[{}^\vee q \wedge q = {}^\wedge walk'_*(j)]$

John walks

$walk'_*(j)$

- Remarks:

  – The last step holds because of Montague's notation convention whereby $\gamma(\beta)(\alpha)$ is rewritten as $\gamma(\alpha,\beta)$.

  – Following Karttunen's suggestion, let's not worry too much about intensionality. Then the formula is true iff Bill stands in the knowing-relation to a certain set of propositions, containing either the proposition that John walks or the proposition that John doesn't walk (by the above equivalence of the emptiness of the set with this proposition), depending on what the facts are.

### 3.3.5 WH-questions

(17) a. who dates Mary
 b. which girl sleeps

- Wh-phrases are interpreted as generalized quantifiers, in the same way as existentially quantified NPs. They combine via quantifying-in with proto-questions which contain free variables that end up bound as a result of the quantification. They can also combine with (non-proto) question which already contain one wh-phrase, but still have free variables.

- direct translation for "standalone" wh-words like *'who'*, *'what'* (ignoring animacy):

  who
  $$\lambda P \exists x[{}^{\vee}P(x)]$$

- wh-words which combine with common nouns (recall that this includes basic nouns as well as those modified by adjectives) are introduced syncategorematically, just like the determiner *'a(n)'*

  which cat
  $$\lambda P \exists x[cat'(x) \wedge {}^{\vee}P(x)]$$

  cat
  $$cat'$$

- But even though now *'which cat'* is semantically equivalent to *'a cat'*, there is no danger of confusion because they belong to different *syntactic* categories ($t/IV$ and $t//IV$, respectively). (This has certain consequences — cf. Section 2.12, especially (65) vs. (66).)

- WH-quantification rule (p. 398, (47)): I want to ignore the antiquated syntactic arcana. So let's just say:

  If $\alpha$ is a wh-phrase and $\phi$ is a (proto-)question with an (unbound) pronoun $PRO_n$ and which does not begin *'whether'*, then the result of putting $\alpha$ and $\phi$ together in the appropriate way is a question.

  And it translates as $\lambda p[\alpha'({}^{\wedge}\lambda x_n[\phi'(p)])]$ (where $x_n$ is the translation of $PRO_n$).

- Notice the semantic difference with ordinary quantifiers like *'a cat'*: Here $\phi$ would be proposition-denoting, and the translation would be $\alpha'({}^{\wedge}\lambda x_n[\phi'])$.

- Example:

(18) who dates Mary

who dates Mary

$$\begin{aligned}
&\lambda p[\lambda P \exists x[{}^{\vee}P(x)]({}^{\wedge}\lambda x_0[\lambda q[{}^{\vee}q \wedge q = {}^{\wedge}date'_*({}^{\vee}x_0, m)](p)])] \\
&= \lambda p[\lambda P \exists x[{}^{\vee}P(x)]({}^{\wedge}\lambda x_0[{}^{\vee}p \wedge p = {}^{\wedge}date'_*({}^{\vee}x_0, m)])] \\
&= \lambda p[\exists x[[{}^{\vee\wedge}\lambda x_0[{}^{\vee}p \wedge p = {}^{\wedge}date_*({}^{\vee}x_0, m)]](x)]] \\
&= \lambda p[\exists x[\lambda x_0[{}^{\vee}p \wedge p = {}^{\wedge}date_*({}^{\vee}x_0, m)](x)]] \\
&= \lambda p[\exists x[{}^{\vee}p \wedge p = {}^{\wedge}date'_*({}^{\vee}x, m)]]
\end{aligned}$$

who
$$\lambda P \exists x[{}^{\vee}P(x)]$$

?he$_0$ dates Mary
$$\lambda q[{}^{\vee}q \wedge q = {}^{\wedge}date'_*({}^{\vee}x_0, m)]$$

he$_0$ dates Mary
$$date'_*({}^{\vee}x_0, m)$$

- Remarks:

  – In case you wonder what's beneath this tree: By Montague's meaning postulates and notational conventions, $date'_*({}^\vee x_0, m)$ is equivalent to $date'(x_0, {}^\wedge\lambda P[{}^\vee P({}^\wedge m)])$ (since *'date'* is an extensional verb). And the latter we get as follows:

he$_0$ dates Mary
$\lambda Q[{}^\vee Q(x_0)]({}^\wedge date'({}^\wedge\lambda P[{}^\vee P({}^\wedge m)]))$
$= [{}^{\vee\wedge} date'({}^\wedge\lambda P[{}^\vee P({}^\wedge m)])](x_0)$
$= date'({}^\wedge\lambda P[{}^\vee P({}^\wedge m)])(x_0)$
$\equiv date'(x_0, {}^\wedge\lambda P[{}^\vee P({}^\wedge m)])$

he$_0$
$\lambda Q[{}^\vee Q(x_0)]$

date Mary
$date'({}^\wedge\lambda P[{}^\vee P({}^\wedge m)])$

date
$date'$

Mary
$\lambda P[{}^\vee P({}^\wedge m)]$

  – The result is a set of propositions, one for each person who dates Mary, saying that person dates Mary. If Mary is undated, the set is empty.
  – As Karttunen says, the propositions in this set *jointly* (not individually) constitute a true and complete answer to the question.
  – But notice that the set contains no proposition saying of some person that he or she does *not* date Mary. Thus at worlds in which there is such a person, the answer is only weakly exhaustive. An interpretation that would always yield a strongly exhaustive answer would be the following instead of the above:

(19) $\lambda p \exists x[{}^\vee p \wedge [p = {}^\wedge date'_*({}^\vee x, m) \vee p = {}^\wedge\neg date'_*({}^\vee x, m)]]$

This, Karttunen says, would be wrong:

i. The following are not equivalent:

(20) a. Bill wonders who dates Mary.
b. Bill wonders who doesn't date Mary.

but would be equivalent if each denoted a strongly exhaustive answer.
[Good point. Same thing with *'knows'*, right?]

ii. (21) would entail that John knows of every individual in existence:

(21) John knows who dates Mary.

which is bizarre.
[Weak argument. There is always some contextually restricted domain of discourse.]

- Typo on page 396, (39b): Left bracket missing after $\wedge$

### 3.3.6 Multiple wh-questions

(22) who dates which girl

who dates which girl

$$\begin{aligned}
&\lambda q[\lambda P\exists y[girl'(y) \wedge {}^\vee P(y)]({}^\wedge\lambda x_1[\lambda p[\exists x[{}^\vee p \wedge p = {}^\wedge date'_*({}^\vee x, {}^\vee x_1)]](q)])]\\
&= \lambda q[\lambda P\exists y[girl'(y) \wedge {}^\vee P(y)]({}^\wedge\lambda x_1[\exists x[{}^\vee q \wedge q = {}^\wedge date'_*({}^\vee x, {}^\vee x_1)]])]\\
&= \lambda q[\exists y[girl'(y) \wedge [{}^\vee{}^\wedge\lambda x_1[\exists x[{}^\vee q \wedge q = {}^\wedge date'_*({}^\vee x, {}^\vee x_1)]]](y)]]\\
&= \lambda q[\exists y[girl'(y) \wedge \lambda x_1[\exists x[{}^\vee q \wedge q = {}^\wedge date'_*({}^\vee x, {}^\vee x_1)]](y)]]\\
&= \lambda q[\exists y[girl'(y) \wedge \exists x[{}^\vee q \wedge q = {}^\wedge date'_*({}^\vee x, {}^\vee y)]]]\\
&\equiv \lambda q[\exists y\exists x[girl'(y) \wedge {}^\vee q \wedge q = {}^\wedge date'_*({}^\vee x, {}^\vee y)]]
\end{aligned}$$

which girl

$\lambda P\exists y[girl'(y) \wedge {}^\vee P(y)]$

who dates $\text{him}_1$

$$\begin{aligned}
&\lambda p[\lambda P\exists x[{}^\vee P(x)]({}^\wedge\lambda x_0[\lambda q[{}^\vee q \wedge q = {}^\wedge date'_*({}^\vee x_0, {}^\vee x_1)](p)])]\\
&= \lambda p[\lambda P\exists x[{}^\vee P(x)]({}^\wedge\lambda x_0[{}^\vee p \wedge p = {}^\wedge date'_*({}^\vee x_0, {}^\vee x_1)])]\\
&= \lambda p[\exists x[[{}^\vee{}^\wedge\lambda x_0[{}^\vee p \wedge p = {}^\wedge date'_*({}^\vee x_0, {}^\vee x_1)]](x)]]\\
&= \lambda p[\exists x[\lambda x_0[{}^\vee p \wedge p = {}^\wedge date'_*({}^\vee x_0, {}^\vee x_1)](x)]]\\
&= \lambda p[\exists x[{}^\vee p \wedge p = {}^\wedge date'_*({}^\vee x, {}^\vee x_1)]]
\end{aligned}$$

girl

$girl'$

who

$\lambda P\exists x[{}^\vee P(x)]$

?$\text{he}_0$ dates $\text{him}_1$

$\lambda q[{}^\vee q \wedge q = {}^\wedge date'_*({}^\vee x_0, {}^\vee x_1)]$

$\text{he}_0$ dates $\text{him}_1$

$date'_*({}^\vee x_0, {}^\vee x_1)$

- Remarks:
  - The last step holds just by virtue of propositional logic.
- Notice: (42) on page 397 should better be *'who dates which girl'*, not *'Who dates which girl?'*.
- Typo on page 397, (45), 2nd line: *'which dates $him_1$'* should be *'who dates $him_1$'*

## 3.4 Other stuff

Most of the rest of the paper is concerned with constraints on readings. Much of it illustrates strengths and weaknesses of the syntax Karttunen uses. In particular, he argues that some constraints on what syntacticians would call movement are accounted for by his semantics. We won't go into this (for now).

## 3.5 Some more things to remember

- In Karttunen's system, the denotations are not partitions (far from it). The propositions in the set are neither jointly exhaustive (only true answers are included) nor mutually exclusive (they are all true).
- Intensions vs. extensions:

| *Object* | *Extension* | *Intension* |
|---|---|---|
| (decl.) sentence | $t$ (truth value) | $\langle s, t\rangle$ (proposition) |
| (proto-)question | $\langle\langle s, t\rangle, t\rangle$ (set of propositions) | $\langle s, \langle\langle s, t\rangle, t\rangle\rangle$ (function from worlds to sets of propositions) |

## References

Hamblin, C. 1958. Questions. *Australasian Journal of Philosophy*, 36:159–168.

Hamblin, C. 1973. Questions in Montague English. *Foundations of Language*, 10:41–53.

Karttunen, L. 1977. Syntax and semantics of questions. *Linguistics and Philosophy*, 1:3–44.

von Stechow, A. 1991. Focusing and backgrounding operators. In Abraham, W., editor, *Discourse Particles.*, pages 37–84. John Benjamins.