Goal: To formally characterize the contribution of Japanese ka and nani to the meanings of (2) and (3).

The plan: • Discuss a means of formalizing meaning
• Introspect about the meaning of questions
• Formalize a semantics for questions like (2)
• Formalization a semantics for indefinites like nanika in (3)
• Discuss some predictions about questions with multiple question words
• Consider extensions to languages other than Japanese.

I. Formalizing meaning (a fairly standard approach)

A sentence like (4) can be true (if Quinn left) or false (if he didn’t).
It depends on the state of the world—in some “possible worlds” it is true, in others false.
Put another way, Quinn left tells us what the world has to look like for it to be true.

(4) Quinn left.

Suppose: To know the meaning of a sentence is to know its truth conditions.

We can formalize the meaning of (4) as a function from states of the world to true/false.

\[
\begin{array}{c}
\downarrow \text{w} \\
\text{Quinn left} \\
\end{array}
\begin{array}{c}
\text{true, if Q left in } \downarrow \text{w} \\
\text{false, if Q did not leave in } \downarrow \text{w} \\
\end{array}
\]

Compositionality. The meaning of the whole is a function of
• the meanings of its parts
• the way it is put together.
So, Quinn and left each contribute to the determination of the truth conditions of (4).

Suppose [Quinn] (and in general any name) picks out a person in the world.

(6)

What about [left]?

You can’t get to truth conditions until you know who left.

So, suppose left is a function—

given a person (the alleged “leaver”), you get truth conditions (true in worlds where the alleged leaver in fact left).

(7)

Let’s write (7) in a more compact way.

It is useful in semantics to use lambda notation to describe functions.

- \( \lambda x . A \) is a function that takes an argument \( x \) and returns \( A \).
- \([ \text{left} ]\) takes an argument (a person) \( x \) and returns the function in the circle.
- The function in the circle depends on the choice of \( x \).
  - It takes a world \( w \) as an argument,
  - and returns true if and only iff \( x \) left in world \( w \).

(8) \([ \text{left} ] = \lambda x. \lambda w. x \text{ left in } w.\)

[Our goal is to come up with something like (8) for Japanese ka and nani.]
II. Introspecting about the meaning of questions (mainly from Hamblin 1958)

Statements—intuitively—can be either true or false (depending on the world state). But not questions.

(9) Who broke the toaster?

They do communicate something, though. For example, we know from (9) that (10) is not a possible answer, but (11) is fine.

(10) ✗ It always rains on the Fourth of July. Not a possible answer
(11) ✓ Homer broke the toaster. A possible answer

Hypothesis: Questions tell us which propositions are possible answers.

“Given a proposition \( p \), a question \( Q \) determines if \( p \) is or is not a possible answer.”

Sounds like a function.
Sounds like a function which takes propositions as arguments and returns \( \text{true} \) or \( \text{false} \).
\( \text{(true} \) if the proposition is a possible answer to the question).\n
Another way to look at functions that return \( \text{true} \) or \( \text{false} \).

(12) \( S = \{ a_1, a_2, a_3, \ldots, a_n, \ldots \} \).

You can characterize this set with a “member of” function:

(13) \( \lambda a. a \in S \).

For any \( a \), (13) returns \( \text{true} \) if \( a \) is in \( S \), \( \text{false} \) otherwise. The information in (13) is equivalent to the information in (12).

\( \implies \) A function that takes an argument \( a \) and returns \( \text{true} \) or \( \text{false} \) can also be thought of as a set of the \( a \)’s for which the function returns \( \text{true} \).

The meaning of a question is the set of propositions which are its possible answers.

The propositions which would be possible answers to (9) are of the form \( x \) broke the toaster, where \( x \) is human.
\( \text{(e.g., \{Homer broke the toaster, Maggie broke the toaster, Lisa broke the toaster, \ldots\})} \)

It is easier to conceptualize these meanings as sets (of possible answers), but we will formally describe these sets in terms of “member of” functions. (Functions which assign \( \text{true} \) to every \( p \) which is a member of the set).
To characterize the set containing propositions like \( x \) broke the toaster where \( x \) is human:

\[
\lambda p . \exists x \in \text{humans} . p = \lambda w . x \text{ broke the toaster in } w
\]

\[
\text{‘A proposition } p \text{ is in the set if and only if } (\lambda p . \)”} \\
\text{there is an } x \\
\text{in the set of humans} \\
\text{such that} \\
\text{\( p \) is } x \text{ broke the toaster.’} \\
\text{\( (\exists x) \)} \\
\text{\( (“\in \text{humans”) \)} \\
\text{\( (“.”) \)} \\
\text{\( (“p = \lambda w . x \text{ broke the toaster in } w”) \)}
\]

**IV. Questions without *ka*, and “flexible functional application”**

It turns out that you can ask questions in Japanese without that *ka* morpheme:

\[
\text{dare-ga kimasita ka?} \\
\text{who-SUBJ came.POLITE Q} \\
\text{‘Who came?’}
\]

\[
\text{dare-ga kimasita ?} \\
\text{who-SUBJ came.POLITE} \\
\text{‘Who came?’}
\]

*Note:* There are certain restrictions on this “*ka-drop*” (Yoshida & Yoshida 1997) 
It can never happen in embedded questions. 
It can only happen in yes/no questions with certain verbs. 
*Point:* It’s systematic enough to suspect it isn’t just “phonology” 
*That is,* it is not unreasonable to think that *ka* is in fact missing in (16)

Both (15) and (16) mean ‘Who came?’ 
The meaning of ‘Who came’ is rendered as the set of its possible answers— 
i.e. the set of propositions of the form \( x \text{ came} \) for \( x \) a person.

\[
\lambda p . \exists x \in \text{people} . p = \lambda w . x \text{ came in } w
\]

Now, if *ka* is actually missing from (16), *ka* is clearly not contributing anything to the meaning.

Let’s concentrate on (16) and try to figure out how to get (17). 
As a first step, consider the minimally different sentence (18).

\[
\text{Taroo-ga kimasita} \\
\text{Taro-SUBJ came.POLITE} \\
\text{‘Taro came.’}
\]

\[
\text{a. } [kimasita] = \lambda x . \lambda w . x \text{ came in } w. \\
\text{b. } [kimasita] ([ Taroo ]) = \lambda w . \text{Taro came in } w.
\]

What could *dare* mean in (16) to yield the representation like (17) instead of like (19b)?
An idea due to Hamblin (1973):

Suppose what (or nani) is not a single thing (like, say, Taro), but a set.

[ Taroo ] points to a person in the world.
[ dare ] is a set of things which point to people in the world
for example, {Taro, Akira, Hanako, Shigeru, Kazuko, … }

This creates a type mismatch:
[ kimasita ] is a function from people to truth conditions (just like English [ left ] was)
It needs a person-type argument, but in (16),
it gets a set of person-type things.

(16)  dare-ga kimasita ?  Repeated from before
     who-SUBJ came.POLITE
     ‘Who came?’

So what do we do? Consider a “real life” analogy:
Suppose we think of a vending machine is a function—
a function that given a quarter returns a gumball.

What if you arrive with a bag of quarters?
The machine does not bags… it takes quarters.

This suggests a natural way of looking at this:
(it’s a lot like mapcar)

- You have a function that can apply to individuals
- You have a set of individuals

so
- Apply the function to each individual in the set (separately).
- When you are done, you have a set of results (instead of just one).

We can refer to this as flexible functional application.
(see also Rooth 1985, Rullmann & Beck 1997)

In our specific example, [ kimasita ] is a function from people to propositions.

Given [ Taro ], it yields the proposition Taro came.
So, Given [ dare ], it yields a set of propositions, one for each member of [ dare ].
Specifically, the propositions like x came, for each x in the set [ dare ].

PAUSE, FOR DRAMATIC EFFECT

That sounds familiar.
That is in fact what we were after as a representation for (16).
The set of possible answers to (16), propositions of the form x came for x a person.
The set for which (17) was a “member of” function:

(17)  λp ∃x∈people. p = λw . x came in w  Repeated from before

Conclusion:
If we suppose that, unlike [ Taro ], [ dare ] is a set, a reasonable approach to
type mismatch resolution automatically yields the desired representation.
Where we are:

- Using the idea that *dare* ‘who’ is represented in the semantics as a set, we saw how to get the desired meaning for (16) [the question without *ka*].
- We haven’t addressed the question of why (15) [the question with *ka*] means the same thing—but clearly if we want to figure out what the meaning of *ka* is (assuming it isn’t meaningless), this is not the place to look.

V. Indefinites formed with *ka*.

Remember that *dare* and *ka* can appear in a non-question:

(20) **dare-ka-ga** kimasita.

*who-Q-SUBJ* came.*POLITE*

‘Someone came.’

If *ka* were *meaningless*, we would expect this to be a question—just like (16).

(16) **dare-ga** kimasita?

*who-SUBJ* came.*POLITE*

‘Who came?’

*ka* is keeping the *set* property of [[dare]] from turning (20) into a set of propositions. *ka* does play *some* role in the semantics.

The statement (20) is true in any world where someone came—

*that is*, in any world where there is a person *x* such that *x* came.

(21) $\exists x \in \text{people. } x \text{ came.}$

In our effort to pin down the meaning of *ka*, consider this minimal pair:

(22) **dare-ka-ga** paatii-ni kita

*who-Q-SUBJ* party-to *came*

‘Someone came to the party.’

(23) $\exists x \in \text{people. } x \text{ came to the party.}$

(24) **dare-mo-ga** paatii-ni kita

*who-MO-SUBJ* party-to *came*

‘Everyone came to the party.’

(25) $\forall x \in \text{people. } x \text{ came to the party.}$

Both *dareka* ‘someone’ and *daremo* ‘everyone’ are formed by appending a particle to *dare* ‘who’.

Since the difference between the meanings is that one involves “∃” and one involves “∀”, the meaning of *ka* probably involves “∃”, the meaning of *mo* probably involves “∀”.

Sidebar on quantifiers (“Quantifier Raising”):

∃ and ∀ are quantifiers; they need to be interpreted outside the predicate.

*That is*, “John bought ∃x” is interpreted as “∃x . John bought x.”

**Stipulation:** A quantifier in argument position is interpreted discontinuously, the quantifier outside the predicate, and the variable in argument position.
Idea: On the surface, (20) looks like (26). But because *ka* is “∃”, it is *interpreted in two places*, one part outside the verb.

(20) **dare-ka-ga** kimasita.

Repeated from before

who-Q-SUBJ came.POLITE

‘Someone came.’

(26)

Surface

![Surface diagram]

Now: [kimasita] is a function (like [left]) that needs a person-type argument.
And: We hypothesized before that [dare] is a set of person-type things.
So: “(ka)” must be a function from sets of individuals to individuals
(since applying [ (ka) ] to [ dare ] provides the argument for [ kimasita ])

Proposal: [ (ka) ] is a *choice function.*

It takes a set as its argument and returns a *member* of that set.

So, in (27), it takes [ dare ] as its argument, returns a member
(which will necessarily be a person).

Proposal: [ ka ] ≈ ∃f (for *f* a choice function)

(“=” because I give a more complicated version in my thesis—
but complicated in ways that are not important here)

The way (27) works:
• *ka* is “∃f” so “∃f” is interpreted outside the predicate (at the position of “ka”),
  the variable *f* is interpreted at the position of “(ka)’’
• (27) translates to (28):

(28) ∃f . f(people) came.

‘There is a way of choosing a member from the set of people
such that the chosen person came.’

Now, before we said that this sentence should mean (21):

(21) ∃x∈people. x came.

Repeated from before

But (28) and (21) are true under exactly the same conditions.
• Any world in which there is a person who came is one in which
  there is a way to pick a person from the set of people
  such that the chosen person came and vice-versa.
So (28) and (21) *characterize the same proposition.* (i.e., it’s ok to stick with (28)).
VI. Back to questions —tying up a loose end

So if *ka* is not meaningless, how come you can ask questions either with or without *ka*?

Answer (has to be): In this context, *ka* turns out to have no effect.

(29) **dare-ga** **kimasita** **ka?**  
who-SUBJ came.POLITE Q  
‘Who came?’

(30) **dare-(ka)-ga** **kimasita** **ka**  
This is responsible for the movement.

(31)

The effect of moving *ka* over $\sqcup C \sqcap$ is that the ‘ $\exists f$ ’ winds up outside the proposition:

(32) $\lambda p \exists f . p = \lambda w . f(\text{people}) \text{ came in } w$.

What this says:

‘A proposition $p$ is in the set if and only if

given a choice function $f$
such that

$p$ is the proposition

$x \text{ came }$ where $x$ is the person chosen by $f$

from the set of people.

$\lambda w . f(\text{people}) \text{ came in } w$’

So it characterizes a set of propositions of the form $x \text{ came }$ where $x$ is a person  
(or, more accurately, where $x$ can be chosen from the set of people).

That’s the same set that was characterized by (17) from before  
(for the variant without *ka*)

(17) $\lambda p \exists x \in \text{people}. p = \lambda w . x \text{ came in } w$  
Repeated from before

And *that* is why the questions with and without *ka* mean the same thing:

Two mechanisms (flexible functional application, movement of *ka* to $C$)  
converge on the same abstract representation.
VII. In support of the apparent “redundancy”, ka in multiple questions

So, we saw that presence vs. absence of ka didn’t seem to affect the meaning of (33–34).

(33) **dare**-ga **kimasita** ka?
**who**-**SUBJ** came.**POLITE** Q
‘Who came?’

(34) **dare**-ga **kimasita** ?
**who**-**SUBJ** came.**POLITE** Q
‘Who came?’

And we just saw why—
two different mechanisms converge on the same set of possible answers.
Do we really need two different mechanisms?
In fact, yes.
It turns out that ka does have an observable effect in multiple questions.

(35) **dare**-ga **nani-**o kaimasita ka?
**who**-**SUBJ** **what**-**OBJ** bought.**POLITE** Q
‘Who bought what?’

(36) **dare**-ga **nani-**o kaimasita ?
**who**-**SUBJ** **what**-**OBJ** bought.**POLITE** Q
‘Who bought what?’

(35), with ka, can be answered with a list of pairs.
(36), without ka, can only be answered with a single pair.

First: What is a “list of pairs” question?

Proposal: A “list of pairs” question is actually a set of questions.
(and you answer each one)

Who bought what?
when answered like
John bought coffee, Mary bought pizza, Howard bought carrots, …
is interpreted as a set of questions
{What did John buy?, What did Mary buy?, What did Howard buy?, …}
each of which you answer.

Let’s see how the proposed analysis of ka actually predicts (35) vs. (36).
The idea: We have two ways of getting sets.

One is flexible functional application, which deals with sets of arguments. The other is moving ka outside the proposition.

In (36) there is no ka, so we have one means available.
We can only get a set of propositions (a single pair question).

In (35), we have ka so we can use both—
We can get a set of propositions (a question) and then a set of questions (a pair list question).

How it works (requires more background than we covered in the talk—see Appendix)

VIII. Beyond Japanese

Ok, so what do Japanese particles tell us about the meaning of questions?

- wh-words (e.g., dare ‘who’) are interpreted as sets.
- “normal” questions involve an \( \exists \) quantifier over choice functions (ka).

Japanese is not alone—lots of languages form ‘someone’ from ‘who’ and a particle.

<table>
<thead>
<tr>
<th>Language</th>
<th>Particle</th>
<th>‘who’</th>
<th>‘someone’</th>
</tr>
</thead>
<tbody>
<tr>
<td>German</td>
<td>wer</td>
<td>‘who’</td>
<td>irgend-wer</td>
</tr>
<tr>
<td>Icelandic</td>
<td>hver</td>
<td>‘who’</td>
<td>ein-hver</td>
</tr>
<tr>
<td>Latin</td>
<td>quis</td>
<td>‘who’</td>
<td>ali-quis</td>
</tr>
<tr>
<td>Romanian</td>
<td>cine</td>
<td>‘who’</td>
<td>cine-va</td>
</tr>
<tr>
<td>Mod. Greek</td>
<td>pijos</td>
<td>‘who’</td>
<td>ká-pjos</td>
</tr>
<tr>
<td>Bulgarian</td>
<td>koj</td>
<td>‘who’</td>
<td>njá-koj</td>
</tr>
<tr>
<td>Serbo-Croatian</td>
<td>ko</td>
<td>‘who’</td>
<td>ne-ko</td>
</tr>
<tr>
<td>Polish</td>
<td>kto</td>
<td>‘who’</td>
<td>ktoš</td>
</tr>
<tr>
<td>Russian</td>
<td>kto</td>
<td>‘who’</td>
<td>kto-to</td>
</tr>
<tr>
<td>Hungarian</td>
<td>ki</td>
<td>‘who’</td>
<td>vala-ki</td>
</tr>
<tr>
<td>Hebrew</td>
<td>mi</td>
<td>‘who’</td>
<td>mi-šehu</td>
</tr>
<tr>
<td>Kannada</td>
<td>yaaru</td>
<td>‘who’</td>
<td>yaar-oo</td>
</tr>
<tr>
<td>Korean</td>
<td>nwukwu</td>
<td>‘who’</td>
<td>nwukwu-nka</td>
</tr>
<tr>
<td>Basque</td>
<td>nor</td>
<td>‘who’</td>
<td>nor-bait</td>
</tr>
<tr>
<td>Sinhala</td>
<td>kauru</td>
<td>‘who’</td>
<td>kau-dô</td>
</tr>
<tr>
<td>English?</td>
<td>where</td>
<td>somewhere</td>
<td></td>
</tr>
</tbody>
</table>

References mentioned
APPENDIX: How the pair-list readings and single-pair readings relate to *ka*.

(36) **dare**-ga **nani**-o kaimasita ?

Repeated from before

who-SUBJ what-OBJ bought.POLITE

‘Who bought what?’

There is no *ka*, so we need flexible functional application (stated below, we need (iv))

(37) FLEXIBLE FUNCTIONAL APPLICATION (taken from Rullmann & Beck 1997)

\[ \lambda f a = (\text{where } f \text{ and } a \text{ are sisters}) \]

(i) \( f(a) \)

(ii) \( \lambda m \exists x. [m = f(x) \land a(x)] \)

(iii) \( \lambda m \exists g. [m = g(a) \land f(g)] \)

(iv) \( \lambda m \exists g \exists x. [m = g(x) \land f(g) \land a(x)] \)

whichever is defined

(38) \( \langle \langle s, t \rangle, t \rangle \)

\( \langle e, t \rangle \)

\( \text{dare} \)

\( \text{nani} \)

\( \text{kaimasita} \)

\( \langle e, e, \langle s, t \rangle \rangle \)

Results in a set of propositions like *x bought y* for every value of *x* in \( \text{[dare]} \) and every value of *y* in \( \text{[nani]} \).

(39) \( \lambda p \exists x \in \text{people} \exists y \in \text{things}. p = \lambda w. x \) bought *y* in *w*.

Picking an answer simultaneously picks a value of *x* and *y*—it’s a single pair answer.

(35) **dare**-ga **nani**-o kaimasita **ka**?

Repeated from before

who-SUBJ what-OBJ bought.POLITE Q

‘Who bought what?’

In (35), the structure is different—*ka* has moved from next to one of the *wh*-words.

(40) \( \langle \langle s, t \rangle, t \rangle \)

\( \langle e, t \rangle \)

\( \text{dare} \)

\( \text{nani} \)

\( \text{kaimasita} \)

\( \langle e, e, \langle s, t \rangle \rangle \)

\( \text{ka} \)

\( \langle e, \langle s, t \rangle \rangle \)

\( \langle e \rangle \)

\( \langle e, \langle e, \langle s, t \rangle \rangle \rangle \)

Artificially suspending FFA, we have:

(41) \( \lambda p \exists f. p = \lambda w. \text{[dare]} \) bought \( f \)(things) in *w*.

In the spirit of FFA, we want to end up with one of these for each member of \( \text{[dare]} \). This requires allowing \( \lambda \)-abstraction to also be “flexible” (see Hagstrom 1998).

(42) \[ \text{[} \lambda i. \varphi \text{]} = \lambda A. \exists \phi. A = \{ \lambda x. \varphi \}^{\text{[}i \rightarrow x\text{]}} \land \forall x. \varphi^{\text{[}i \rightarrow x\text{]}} \in \varphi^{\text{[}i \rightarrow x\text{]}} \]

where a) \( \varphi \) is a set (type \( < \mu t > \)), b) the result is composable.

In the end we get something characterizable as:

(43) \( \lambda Q \exists x \in \text{people}. Q = \lambda p \exists f. p = \lambda w. x \) bought \( f \)(things) in *w*. 