Connecting morphology and semantics: What Japanese particles tell us about the meaning of questions

(1) Taroo-ga hon-o kaimasita. ‘Taro bought a book.’
(2) Taroo-ga hon-o kaimasita ka? ‘Did Taro buy a book?’
(3) Taroo-ga nani-ka kaimasita. ‘Taro bought something.’
(4) Taroo-ga nani-o kaimasita ka? ‘What did Taro buy?’

Indefinites are derived forms, almost without exception. Esperanto:

<table>
<thead>
<tr>
<th>wh</th>
<th>some</th>
<th>no</th>
</tr>
</thead>
<tbody>
<tr>
<td>person</td>
<td>kiu</td>
<td>iu</td>
</tr>
<tr>
<td>thing</td>
<td>kio</td>
<td>io</td>
</tr>
<tr>
<td>place</td>
<td>kie</td>
<td>ie</td>
</tr>
</tbody>
</table>

Japanese: dare-ka | someone | ka | or
Sinhala: mokak-də | something | də | or(alt.qs)
Kannada: yaar-oo | someone | -oo | or
Korean: nwukwu-na | anyone | -(i)na | or
| nwukwu-tunci | anyone | -(i)tunci | or
Russian: kto-libo | anyone | libo | or
Basque: edo-nor | anyone | edo | or
Nanay: uj-nuu | someone | -nuu | or

A little project: • The contribution of Q • The contribution of wh-words.

| I. Formalizing meaning | (a fairly standard approach) |

A sentence like (5) 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.

(5) Quinn left.

Suppose: To know the meaning of a sentence is to know its truth conditions.
We can formalize the meaning of (5) as a **function from states of the world to true/false**. It’s a “little machine” that, given a world, will tell you “true” or false”. The little machine depicted below is the “Quinn left” machine.

(6)

\[ \text{true, if } Q \text{ left in } w \]

\[ \text{false, if } Q \text{ did not leave in } w \]

Also assumed: The difference in meaning between *Pat ate the cookie* and *Pat baked the cookie* is a **direct** consequence of the difference in meaning between *ate* and *baked*.

The *Pat ate the cookie* machine and the *Pat baked the cookie* machine share most of the same parts, except for one.

<table>
<thead>
<tr>
<th><strong>Compositionality.</strong></th>
<th>The meaning of the whole is a function of</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• the meanings of its parts</td>
</tr>
<tr>
<td></td>
<td>• the way it is put together.</td>
</tr>
</tbody>
</table>

So, *Quinn* and *left* each **contribute** to the determination of the truth conditions of (5).

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

(7)

What about \[ \text{[left]} \]? You can’t get to truth conditions until you know who left. That is, *left* is a little machine of its own, except it doesn’t spit out “true” and “false”. Rather, it’s a machine that builds machines—give it a person like *Quinn* and it will give you a *Quinn left* machine.

It’s a function that returns a function.
Let’s write (8) in a more compact way.

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

- \( \lambda x \cdot A \) is a function that takes an argument \( x \) and returns \( A \).
- \( \llbracket \text{left} \rrbracket \) 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 \( \text{true} \) if and only iff \( x \) left in world \( w \).

(9) \[ \llbracket \text{left} \rrbracket = \lambda x \lambda w \cdot x \text{ left in } w. \]

[Our goal is to come up with something like (9) 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.

(10) Who broke the toaster?

They do communicate something, though. For example, we know from (10) that (11) is not a possible answer, but (12) is fine.

(11) \( \odot \) It always rains on the Fourth of July. Not a possible answer
(12) \( \ominus \) 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 that takes *propositions* as arguments and returns *true* or *false.*

\( \text{(true if the proposition is a possible answer to the question).} \)

---

**Another way to look at functions that return *true* or *false.***

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

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

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

For any \( a \), (14) returns *true* if \( a \) is in \( S \), *false* otherwise.

The information in (14) is *equivalent* to the information in (13).

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

---

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

The propositions that would be possible answers to (10) are of the form

\( x \text{ broke the toaster}, \) where \( x \) is human.

(e.g., \{Homer broke the toaster, Maggie broke the toaster, Lisa broke the toaster, \ldots\})

It is easier to *conceptualize* of these meanings as sets (of possible answers),
but we will formally *describe* these sets in terms of “member of” functions.
(Functions that assign *true* to every \( p \) which is a member of the set).

To characterize the set containing propositions like \( x \text{ broke the toaster} \) where \( x \) is human:

(15) \( \lambda p . \ \exists x \in \text{humans} . \ p = \lambda w . \ x \text{ broke the toaster in } w \)

‘A proposition \( p \) is in the set if and only if
there is an \( x \)
in the set of humans
such that
\( p \) is \( x \text{ broke the toaster}.‘

\( \text{("\( \lambda p . \)")} \)
\( \text{("\( \exists x \)")} \)
\( \text{("\( \in \text{humans} \)")} \)
\( \text{("\( . \)")} \)
\( \text{("\( p = \lambda w . \) x broke the toaster in } w \)} \)
Questions without *ka*, and “flexible functional application”

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

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

*who-SUBJ* came.POLITE *Q*

‘Who came?’

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

*who-SUBJ* came.POLITE

‘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 (17)

Both (16) and (17) 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 came* for *x* a person.

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

Now, if *ka* is actually *missing* from (17),

*ka* is not contributing anything to the meaning.

Let’s concentrate on (17) and try to figure out how to get (18).

As a first step, consider the minimally different sentence (19).

(19) **Taro-ga** kimasita

*Taro-SUBJ* came.POLITE

‘Taro came.’

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

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

What could *dare* mean in (17) to yield the representation like (18) instead of like (20b)?

An idea due to Hamblin (1973):

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

\[ \text{ [ Taro ] } \text{ points to a person in the world.} \]

\[ \text{ [ dare ] } \text{ is a set of things which point to people in the world} \]

for example, \{*Taro, Akira, Hanako, Shigeru, Kazuko, …\}
This creates a type mismatch:
\[ \text{kimasita} \] is a function from people to truth conditions (just like English \[ \text{left} \] was).
It needs a person-type argument, but in (17), it gets a set of person-type things.

\[ \text{(17) dare-ga kimasita ?} \]
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 take bags... it takes quarters.

This suggests a natural way of looking at this:

• 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, \[ \text{kimasita} \] is a function from people to propositions.

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

\text{PAUSE, FOR DRAMATIC EFFECT}

A set of propositions like \text{x came}, one for each member \text{x} of \[ \text{dare} \],
That sounds familiar.
That is in fact what we were after as a representation for (17).
The set of possible answers to (17), propositions of the form \text{x came} for \text{x} a person.
The set for which (18) was a “member of” function:

\[ \text{(18) } \lambda p \exists x \in \text{people. } p = \lambda w . x \text{ came in } w \]

\text{Repeated from before}

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

- Using the idea that \textit{dare} ‘who’ is represented in the semantics as a set, we saw how to get the desired meaning for (17) [the question without \textit{ka}].
- We haven’t addressed the question of why (16) [the question with \textit{ka}] means the same thing—but clearly if we want to figure out what the meaning of \textit{ka} is (assuming it isn’t \textit{meaningless}), this is not the place to look.

V. \textbf{Indefinites formed with \textit{ka}.}

Remember that \textit{dare} and \textit{ka} can appear in a non-question:

\begin{align*}
(21) & \quad \text{dare-\textit{ka}}-\text{ga kimasita.} \\
& \quad \text{who-\textit{Q}}-\text{SUBJ came. POLITE} \\
& \quad \text{‘Someone came.’}
\end{align*}

If \textit{ka} were \textit{meaningless}, we would expect this to be a question—just like (17).

\begin{align*}
(17) & \quad \text{dare-\textit{ga} kimasita ?} & \text{\textit{Repeated from before}} \\
& \quad \text{who-\textit{SUBJ came. POLITE}} \\
& \quad \text{‘Who came?’}
\end{align*}

\textit{ka} is keeping the \textit{set} property of \textit{[dare \textit{]} from turning (21) into a set of propositions. \textit{ka} does play \textit{some} role in the semantics.

The statement (21) is true in any world where someone came—\textit{ that is,} in any world where there is a person \textit{x} such that \textit{x} came.

\begin{align*}
(22) & \quad \exists x \in \text{people. } x \text{ came.}
\end{align*}

In our effort to pin down the meaning of \textit{ka}, consider this minimal pair:

\begin{align*}
(23) & \quad \text{dare-\textit{ka}}-\text{ga paatii-ni kita} & \text{(24) } \exists x \in \text{people } . x \text{ came to the party.} \\
& \quad \text{who-\textit{Q}}-\text{SUBJ party-to came} \\
& \quad \text{‘Someone came to the party.’}
\end{align*}

\begin{align*}
(25) & \quad \text{dare-\textit{mo}}-\text{ga paatii-ni kita} & \text{(26) } \forall x \in \text{people } . x \text{ came to the party.} \\
& \quad \text{who-\textit{MO}}-\text{SUBJ party-to came} \\
& \quad \text{‘Everyone came to the party.’}
\end{align*}

Both \textit{dareka} ‘someone’ and \textit{daremo} ‘everyone’ are formed by appending a particle to \textit{dare} ‘who’

Since the difference between the meanings is that one involves \textit{\textup{∃}} and one involves \textit{\textup{∀}} , the meaning of \textit{ka} probably involves \textit{\textup{∃}} , the meaning of \textit{mo} probably involves \textit{\textup{∀}} .
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, (21) looks like (27). But because ka is “∃”, it is interpreted in two places, one part outside the verb.

(21) 
\begin{align*}
\text{dare-ka-ga kimasita.} & \\
\text{who-Q-SUBJ came.POLITE} & \\
\text{‘Someone came.’}
\end{align*}

(27) 
\begin{align*}
\text{dare} & \\
\text{ka} & \\
\text{kimasita}
\end{align*}

Now: \[
\text{[kimasita]}\] is a function (like \[
\text{[left]}
\]) that needs a person-type argument.

And: We hypothesized before that \[
\text{[dare]}
\] is a set of person-type things.

So: “(ka)” must be a function from sets of individuals to individuals (since applying \[
\text{[(ka)]}
\] to \[
\text{[dare]}
\] provides the argument for \[
\text{[kimasita]}
\])

Proposal: \[
\text{[(ka)]}
\] is a choice function. It takes a set as its argument and returns a member of that set.

So, in (28), it takes \[
\text{[dare]}
\] as its argument, returns a member (person).

Proposal: \[
\text{[(ka)]} \equiv \exists f
\] (for \(f\) a choice function)

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

The way (28) works:

- \(ka\) is “\(\exists f\)” so “\(\exists f\)” is interpreted outside the predicate (at the position of “\(ka\)”), the variable \(f\) is interpreted at the position of “\((ka)\)”
- (28) translates to (29):

(29) \[
\exists f. f(\text{people}) \text{ came.}
\]

‘There is a way of choosing a member from the set of people such that the chosen person came.’
(29) \( \exists f . f(\text{people}) \text{ came.} \)

'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 (22):

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

Repeated from before

But (29) and (22) 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 (29) and (22) characterize the same proposition. (i.e., it's ok to stick with (29)).

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.

(30) \( \text{dare-} \text{ga kimasita } ka? \)

\begin{tabular}{l}
who-\text{SUBJ} \hspace{1cm} \text{came.} \hspace{1cm} \text{POLITE} \hspace{1cm} \text{Q} \\
\end{tabular}

‘Who came?’

(31) \( \text{dare-}(ka) \text{-ga kimasita } ka \)

This is responsible for the movement.

(32) \hspace{1cm} \text{QP} \hspace{1cm} \text{ka} \hspace{1cm} \text{C}

\hspace{1cm} \text{(syntactic proposal from Hagstrom 1998)}

The effect of moving \( ka \) over \( [C] \) is that the ‘\( \exists f \)’ winds up outside the proposition:

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

What this says:

‘A proposition \( p \) is in the set if and only if there is 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.’

\( (\lambda p \)’)

\( (\exists f \)’

\( (\cdot \cdot \cdot \)’

\( (p = \)’

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

So it characterizes a set of propositions of the form \( x \) 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 (18) from before (for the variant without \( ka \))

(18) \( \lambda p \exists x \in \text{people}. \ p = \lambda w. x \) 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 (34–35).

(34) \( \text{dare-ga} \ \text{kimasita} \ \text{ka?} \)
    \( \text{who-SUBJ} \ \text{came.POLITE} \ \text{Q} \)
    ‘Who came?’

(35) \( \text{dare-ga} \ \text{kimasita} \ ? \)
    \( \text{who-SUBJ} \ \text{came.POLITE} \)
    ‘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.

(36) \( \text{dare-ga} \ \text{nani-o} \ \text{kaimasita} \ \text{ka?} \)
    \( \text{who-SUBJ} \ \text{what-OBJ} \ \text{bought.POLITE} \ \text{Q} \)
    ‘Who bought what?’

(37) \( \text{dare-ga} \ \text{nani-o} \ \text{kaimasita} \ ? \)
    \( \text{who-SUBJ} \ \text{what-OBJ} \ \text{bought.POLITE} \)
    ‘Who bought what?’

(36), with \( ka \), can be answered with a list of pairs.
(37), 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.

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 conjunction with C).

In (37) there is no *ka*, so we have one means available.
We can only get a set of propositions (a single pair question).
In (36), 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).

To show how this works can get very technical, but here’s the idea:

(38) *What did John buy?*
comes out (with the help of Q moving from beside *what*) to be

(39) \( \lambda p \, \exists f . \ p = \lambda w . \text{John bought } f(\text{things}) \text{ in } w. \)
or

(40) \( \lambda p \, \exists f . \ p = \lambda w . \text{bought}(f(\text{things}))(\text{John}) \text{ in } w. \)

and if we now substitute in *who* for *John*, we are giving *bought* a set when it wants just a single argument.

(41) \( \lambda p \, \exists f . \ p = \lambda w . \text{bought}(f(\text{things}))(\llbracket \text{who} \rrbracket) \text{ in } w. \)

So we get a set of (41)s, one for each thing in \( \llbracket \text{who} \rrbracket \). A set of questions:
{What did John buy?, What did Mary buy?, What did Howard buy?, …}
or

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

(true of any Q that, for some person \( x \), is a set of propositions that can, by picking a choice function \( f \), be written as ‘\( x \) bought that which \( f \) picks from the set of things’)

References mentioned


