Quantification Support with Indeterminate Pronouns

Alastair Butler

*Japan Society for the Promotion of Science and Tohoku University*

ajb129@hotmail.com

This paper is concerned with one possible way of having quantification in a language without determiners, taking Japanese as a case study. After starting with data that suggests deep connections between the ‘morphosyntax’ employed for constituent questions and for quantification, we present a formal account that initially considers sentences without any quantification to see how argument linking is nevertheless achieved when case marking is added to noun phrases. We then look at how quantification can subsequently be added to the system with the aid of indeterminate pronouns (e.g., *dare*, *nani*, *dono*) and particles (*ka* and *mo*). We end by considering the kinds of grammatical effects that arise from this integration.

*Keywords: quantification, indeterminate pronouns, particles, scope control, constituent questions*

1. Introduction

In (1) we see examples of constituent questions from Japanese, with the WH-words *dare* ‘who’, *nani* ‘what’ and *dono* ‘which’, and the question particle *ka* that takes a clause final position.

(1)

a. Dare-ga nani-o kaita ka?
   who-nom what-acc wrote q
   ‘Who wrote what?’

b. Dono hon-ga omosiroi ka?
   which book-nom is-interesting q
   ‘Which book is interesting?’


Kuroda (1965) refers to Japanese WH-words like *dare*, *nani* and *dono* as “indeterminate pronouns” because they have a broader distribution than English WH-words. In (2a), for example, *dare* appears with *dare-mo* ‘everyone’ to form a universal quantifier, and *nani* occurs with *nani-ka* ‘something’ to give an existential quantifier. In (2b) we can see *dono* as part of *dono hon-mo* ‘every book’, thus contributing to a general method for creating instances of restricted universal quantification.

\[(2)\]
\[
a. \text{Dare-mo-ga nani-ka-o kaita.}
\]
who-q-nom what-q-acc wrote
Everyone wrote something.
\[
b. \text{Dono hon-mo-ga omosiroi.}
\]
which book-q-nom is-interesting
‘Every book is interesting.’

Kuroda (1965), Kim (1991), Hagstrom (1998), Shimoyama (2006), among others, observe that it is not only the *nani* part of *nani-ka* that is familiar from questions, arguing that *-ka* should be identified as being the same morpheme as the question particle *-ka*. This is an assumption we will adopt here, to make for a complete convergence in the mechanisms Japanese exploits to code constituent questions and quantification.\(^1\)

The remainder of the paper is structured as follows. In section 2 we begin with a brief sketch of the semantic theory we assume in this paper and then use this theory to introduce a general method for encoding predicates, with

\(^1\) The *-ka* morpheme found in questions and with existential quantification can also, as Kuroda (1965) observed, be used as a marker of disjunction (like English or) between nominal arguments, or between sentences. This has led to accounts (e.g., Gill, Harlow and Tsoulas 2004) that develop the logical connection between existential quantification and disjunction. Also, the *mo* morpheme can be used as a marker of conjunction (like English and), suggesting a further link between the method used to realise universal quantification and conjunction. However, Haspelmath (1997) argues against any such connections on the grounds that these equivalences are exceptional from a typological perspective, and so likely cases of coincidence. In this paper we refrain from taking a stance on this issue.
the consequence that predicates can only be semantically evaluated against environments that support their grammatical requirements and no more. In section 3 we look at sentences without any quantification to see how argument linking is nevertheless achieved when case marking is added to noun phrases. In section 4 we look at how quantification can subsequently be added to the system with semantic encodings for the indeterminate pronouns and the particles -ka and -mo. In section 5 we consider constituent questions and provide an account for their susceptibility to intervention effects. In section 6 we provide a summary of findings.

2. Predicates

Our formal account of quantification in Japanese will be phrased in terms of Scope Control Theory (SCT) (Butler 2007). This provides a small logical language with fine-grained and restricted scope management that combines the static reformulations of Dynamic Semantics by Cresswell (2002) and Dekker (2002) with the Sequence Semantics of Vermeulen (1993). In the appendix to this paper SCT evaluation is defined in terms of providing a translation \((g, e)^\circ\) that takes assignment \(g\) and SCT expression \(e\) and returns a translation formulated according to the conventions of predicate logic notation. For translation an assignment forms a mapping assigning a (possibly empty) sequence of predicate logic variables to each SCT binding name. The resulting translation provides a ‘snapshot’ of the dependencies semantic evaluation establishes.

Seven operators serve as the building blocks of SCT expressions:

- \(T(x, i)\) builds a term for binding name \(x\) with integer \(i\);
- \(\text{Use}(x, e)\) supports binding by incrementing an \(x\) usage count;
- \(\text{Hide}(x, e)\) terminates an \(x\) usage count;
- \(\text{Close}(x, e)\) creates fresh bindings for \(x\) based on an \(x\) usage count;
- \(\text{Lam}(x, y, e)\) shifts a single binding from \(x\) to \(y\);
- \(\text{Garb}(n, x, y, e)\) shifts potentially multiple bindings from the binding names of \(x\) to \(y\), leaving exactly \(n\) bindings open for each binding name of \(x\); and
- \(\text{Rel}(x, y, r, e)\) builds a relation from relation name \(r\) with sequence of
arguments \( e \), potentially changing the assignment for each argument based on binding name sequences \( x \) and \( y \).

Having these operators provides a means to gain control over the availability of bindings and enforce binding roles by making evaluations of SCT expressions sensitive to what should and should not be present as a binding. This is achieved in a general way with check, (3). Taking as parameters a binding name \( y \) with a context role, a sequence of binding names \( d \), a relation name \( s \) and a sequence of arguments \( l \), check returns an \( s \) relation over \( l \) that is sensitive to the names of \( d \).

\[
(3) \\
\text{check} = \lambda y. \lambda d. \lambda s. \lambda l. \text{Rel}(d, \text{map}(\lambda x. y) d, s, l)
\]

An example of applying check is illustrated in (4). This creates a semantically vacuous relation that is sensitive to the “\( ga \)” binding name, such that an evaluation of \( f \) against assignment \( g \) is only possible when the number of occurrences of Use(“\( ga \)” , #) inside \( f \) but outside any Hide(“\( ga \)” , #) equals the exact number of bindings open for the “\( ga \)” name in \( g \).

\[
(4) \\
\text{check “c” [“ga”] “” [f] → Rel([“ga”],[“c”], “”, [f])}
\]

This ability to bring about checks is incorporated into \( r \), (5), which, in addition to forming a predicate that places checks on bindings, also creates the required support for the bindings the predicate requires. \( r \) takes four parameters: \( lc \), \( fh \), \( args \), and \( s \). \( lc \) and \( fh \) tell two distinct instances of check which binding names they need to be sensitive to. The first check has vacuous semantic content and checks for binding names that have the role of providing local bindings. This is immediately followed by a \text{foldl} instruction to create the relevant Use support for the local bindings that need to be present. The second check is made sensitive to the binding names that have the role of providing fresh bindings, and has the semantic content of the predicate itself. The remaining parameters are specific to the predicate instance: \( args \) gives the binding names for the arguments of the
predicate; while \( s \) provides the name of the predicate.

(5)
\[
\begin{align*}
\text{foldl} & (\lambda (x, f). \text{Hide}(x, f)) (\text{check} \quad \text{"c" \quad \text{lc} \quad \text{""}} [\text{foldl} (\lambda (x, f). \text{Use}(x, f)) (\text{check} \quad \text{"c" \quad \text{fh \quad s (map (\lambda x. \text{T}(x, 0)) \text{args}) \text{args}})]) \text{lc}
\end{align*}
\]

Note how, in addition to being part of the \text{check} for the \text{lc} bindings, creating \text{Use} requirements from the binding names of \text{args}, \text{foldl} is also used to add \text{Hide} information with the binding names of the \text{lc} parameter. This ensures that the local usage information of the predicate does not adversely affect potentially superordinate predicates with their own checks. Also note how the predicate itself created with the second \text{check} takes as arguments terms constructed with the aid of \text{map} from the binding names of \text{args}.

In order to provide specific instances of predicates we first need to establish values for the \text{lc} and \text{fh} parameters. For the examples of this paper it will be enough to keep to the values of (6). But note that \text{lc} and \text{fh} might be tailored for each expression, to make instrumentation specific to the data that motivates it. This gives the advantage that unnecessary checks need never be made, while also providing the means to adapt to expressions containing binding names from case markers besides \text{ga} and \text{o}.

(6)
\[
\begin{align*}
\text{lc} & = ["h", "ga", "o"] \\
\text{fh} & = ["x"]
\end{align*}
\]

We can now, for example, provide encodings for predicates as in (7).

(7)
\[
\begin{align*}
\text{omosiroi} & = r \text{ lc fh ["ga"]"is interesting"} \\
\text{kaita} & = r \text{ lc fh ["ga", "o"]"wrote"}
\end{align*}
\]
In (8) we illustrate how omosiroi reduces to an SCT expression with the requirements that there should be a single “ga” binding for its single argument, and no open “h”, “o” or “x” bindings.

(8)

\[
\text{omosiroi} \rightarrow \\
\text{Hide(“o”),} \\
\text{Hide(“ga”),} \\
\text{Hide(“h”),} \\
\text{Rel([“h”, “ga”, “o”], [“c”, “c”, “c”], “”, [} \\
\text{Use(“ga”),} \\
\text{Rel([“x”], [“c”], “is interesting”, [} \\
\text{T(“ga”, 0)]))))
\]

To sum up this section on predicates, let us note that applying \( r \) to form a predicate that restricts dependencies follows from merely declaring conditions for the grammatical dependencies of the predicate. In saying the dependencies that a predicate supports, we are not just saying what the bindings are that the predicate must get, but also what the bindings are that the predicate must not get relative to the potential local and fresh bindings. With this result there is no need for syntax to mark the dependency of what opens a checked binding with whatever needs to be bound, as the only licensed dependencies will be the dependencies that we shall want to see made. At the same time this need not impose overwhelmingly severe constraints, as constraints hold only for checked bindings. If a binding is not checked (e.g., the context binding “c” is never checked) then there are no constraints enforced on the binding. Moreover it is from this fine control over whether a given binding forms part of a given check that the different roles for bindings can emerge. Thus the role of local bindings is created because the bindings of lc are checked with subordination (or termination) but nowhere else, while the role of fresh bindings is created because the bindings of fh are checked with coordination (or termination) but nowhere else.
3. Case marked noun phrases

Let us now consider how we can provide a role for noun phrases with case markers, such as the subject case marker \( ga \) and the direct object case marker \( o \). To have noun phrases that are capable of supporting arbitrary restriction material it is necessary to insulate the restriction from the influence of the containing clause, while maintaining the binding that it is the purpose of the noun phrase to introduce both inside the restriction and outside in the containing clause.

We can accomplish the required insulation of a noun phrase restriction with \( \text{rest} \), (9). This takes three parameters: \( lc \), \( x \) and \( f \). \( lc \) provides the local binding names which informs garbage collection (shifting of bindings) with \( \text{Garb} \); \( x \) provides the binding name that the noun phrase opens in the containing clause; and \( f \) provides the content of the restriction.

\[
(9) \quad \text{rest} = \lambda \, lc \cdot \lambda \, x \cdot \lambda \, f. \\
\text{Garb}(0, \text{"h"}, \text{"c"}, \\
\text{Lam}(x, \text{"h"}, \text{Garb}(0, \text{diff}(lc, \text{["h"]}), \text{"c"}, f)))
\]

Calling \( \text{rest} \) results in the removal of all local bindings, with the exception of the binding given as \( x \), which remains open as an “h” binding. This garbage collection occurs with two ordered calls of \( \text{Garb} \). The first shifts any “h” binding to a “c” binding. Such a binding would be open if the restriction calling \( \text{rest} \) was itself embedded inside another noun phrase restriction. Having this instance of \( \text{Garb} \) ensures there can be no interference from bindings inherited from another noun phrase. This is followed by the shifting of the binding that the noun phrase itself introduces to an “h” binding. The remaining call of \( \text{Garb} \) acts over the names of \( fh \) minus the “h” name (removed with \( \text{diff} \), see the appendix). In short, \( \text{rest} \) has the consequence of ensuring that the binding of the noun phrase itself enters the restriction as the only local binding. It follows that nouns will have to take forms like in (10) which allow for linking to a single open “h” binding, while allowing for no other open binding for the names of \( lc \) and \( fh \).
This setup for nouns is attractive because it provides a uniform way of integrating nouns into the noun phrase in a manner that is regardless of the binding name the noun phrase contributes to the containing clause. The approach also offers the advantage of motivating a lexical distinction between nouns and main predicates of a clause, both of which create predicates, but with main predicates unable to link to or support “h” bindings, which nouns by contrast require. This forces nouns and main predicates into a complementary distribution.

We will suppose that case markers themselves invoke $\text{kp}$, (11). This takes six parameters: $\text{lc}$, $\text{fh}$, $x$, $x$, $l$ and $f$. $\text{lc}$ is passed as a parameter to $\text{rest}$; $\text{fh}$ tells instances of $\text{check}$ which binding names they need to be sensitive to; $l$ provides a sequence of expression and is passed to $\text{rest}$ to provide the content of the restriction; $x$ provides the binding name for the noun phrase; $y$ is used to provide the fresh binding for the noun phrase (so will be a name from $\text{fh}$); and $f$ provides the rest of the containing clause over which the noun phrase takes scope.

(11)

\[
kp = \lambda \text{lc}.\lambda \text{fh}.\lambda x.\lambda y.\lambda l.\lambda f. \\
(\lambda a.\text{if } y = "x" \text{ then Use("x", a) }\text{ else } a) (\\
(\lambda c.\text{Lam}(y, x, c[\text{rest } \text{lc } x (c \ l), f])) (\\
\lambda l.\text{check }"c" \text{ fh }"\land" l))
\]

To get specific instances of case markers, we can continue to assume the definitions for $\text{lc}$ and $\text{fh}$ of (6), and so provide the forms for the case markers $g\alpha$ and $o$, as in (12).

(12)

\[
g\alpha = kp \text{ lc } fh "g\alpha" \\
o = kp \text{ lc } fh "o"
\]
Quantification Support with Indeterminate Pronouns

One other ingredient our analysis requires is the means to combine the operations we have now introduced. Following categorial grammar (Ajdukiewicz 1935, Bar-Hillel 1953), we will manage function applications with the infix operators ‘/’ and ‘\’ of (13).

\[(13)\]
\[
a. \ f/a = f \ a \\
b. \ a\f = f \ a
\]

We are now finally in a position to illustrate the evaluation of complete clauses. For example, let us consider the examples of (14) from which we get the evaluation results of (15).

\[(14)\]
\[
a. \ [\text{hon}] \text{\ ga} “x”/\text{omosiroi} \\
b. \ [\text{hon}] \text{\ ga} “x”/([\text{hon}]\text{\ o “x”}/\text{omosiroi}) \\
c. \ [\text{hito}] \text{\ ga} “x”/([\text{hon}]\text{\ o “x”}/\text{kaita})
\]

\[(15)\]
\[
a. \ \exists g: (g, (14a))^o = (\text{book(x)} \land \text{is interesting(x)}) \\
b. \ \forall g: (g, (14b))^o = * \\
c. \ \exists g: (g, (14c))^o = (\text{person(y)} \land (\text{book(x)} \land \text{wrote(y, x)}))
\]

With (15a) we see that an evaluation with \text{omosiroi} is possible when an “x” binding shifted to a “ga” binding by ga need be the only open binding. With (15b) we see that \text{omosiroi} is impossible with an additional “o” binding, while such an environment will support \text{kaita}, (15c).

Note that the results of (15) produce predicate logic translations with only free variables. This illustrates how no quantification is involved in establishing argument links, with the “x” bindings that are shifted by the case markers needing to be already present in the environment against which evaluation occurs. This begs the question of how it can be guaranteed that the relevant bindings are in place. One answer is that because evaluation is always with respect to an environment, all that matters is that \text{some} environment can be found to support evaluation. Another answer is to assume
that evaluation of an expression occurs with respect to an “\(x\)” closure inherited from the discourse, which would essentially follow the idea from Discourse Representation Theory (Kamp and Reyle 1993) that there is existential closure at the level of discourse. An instance of \(\text{Close}(\text{"x"}, f)\) will create fresh bindings for as many instances of \(\text{Use}(\text{"x"}, \#)\) as occur in \(f\) outside the scope of any \(\text{Hide}(\text{"x"}, \#)\). Note that an instance of \(\text{Use}(\text{"x"}, \#)\) will have been introduced as part of each case marker that shifts an “\(x\)” binding. We can see the results of such existential closure with (16), where evaluation succeeds against the empty assignment when possible. Also note how existential quantifiers now appear in the resulting predicate logic translations to leave no free variables.

\[
\begin{align*}
\text{(16)} \\
\text{a. } & (\lambda, \text{Close}(\text{"x"}, (14a)))^\circ = \exists x (\text{book}(x) \land \text{is interesting}(x)) \\
\text{b. } & \forall g: (g, \text{Close}(\text{"x"}, (14b)))^\circ = * \\
\text{c. } & (\lambda, \text{Close}(\text{"x"}, (14c)))^\circ = \exists x y (\text{person}(y) \land (\text{book}(x) \land \text{wrote}(y, x)))
\end{align*}
\]

Let us end this section by considering how we can add the possibility of relative clauses to the current account. This we can do with \(\text{relc}\), (17), which brings about a shift of the “\(h\)” binding to an \(x\) binding, where \(x\) is an open parameter.

\[
\text{(17)} \\
\text{relc} = \lambda x \lambda f. \text{Lam}(\text{"h"}, x, f)
\]

Having \(\text{relc}\), we can, for example, create the expression (18), where the binding of the head noun \(\text{hon}\) serves as the subject in the main clause, as determined by \(\text{ga}\), and as the object in the relative clause, as determined by the shift of \(\text{relc} \text{ “o”}\). This allows for the evaluation of (19), where \(g\) is an assignment with two “\(x\)” bindings, but no other bindings.

\[
\text{(18)} \\
[\text{relc “o” ([hito]\text{ga “x”/kaita)}, \text{hon}]\text{ga “x”/omosiroi}]
\]
(19)  
\[(g, (18))^o = (((\text{person}(x) \land \text{wrote}(x, y)) \land \text{book}(y)) \land \text{is interesting}(y))\]

To sum up this section, marking noun phrases with case overtly codes the name under which the noun phrase binds in the clause, here “ga” (subject binding) or “o” (direct object binding). Despite the different binding names of noun phrases inside the clause, it was still possible for all nominals to link in a uniform way to their noun phrase restriction, via an “h” binding.

4. Adding quantification

This section forms the core of what we want to show in this paper, which is how quantification only becomes possible with the supporting presence of an indeterminate pronoun, and this without any change to the set up already established with case marked noun phrases. Indeterminate pronouns can be introduced as in (20), as providing nouns much like (10), but with the addition of providing a Use(“q”, #) instruction.

(20)  
dare = Use(“q”, r lc fh [“h”] “person”)  
nani = Use(“q”, r lc fh [“h”] “thing”)  
dono = \lambda f.\text{Use(“q”, f)}

In addition to the indeterminate pronouns, we also require some means to create instances of quantification. This is accomplished with the particles ka, (21), and mo, (22). Each takes three parameters: k, l and f. When a particle is combined with a case marker, e.g., ga or o: k will be the case marker; l will be a sequence of expressions, providing the restriction for a noun phrase; and f will provide the material of the nuclear scope.

(21)  
ka = \lambda k.\lambda l.\lambda f.  
\text{Hide(“q”, Close(“q”, k “q” l (Hide(“q”, f)))))}
(22)  
\[ \text{mo} = \lambda\; k, \lambda\; l, \lambda\; f. \]
\[ (\lambda\; \text{neg} \cdot \text{neg} (\]
\[ \text{Hide}("q", \]
\[ \text{Close}("q", k"q") \; 1 \; (\text{neg} (\text{Hide}("q", f))))))) (\]
\[ \lambda\; f.\text{Rel}(\text{nil}, \text{nil}, "\neg", [\]
\[ \text{Hide}("x", \text{Close}("x", f))))] \]

Providing sufficient \text{Use} ("q", #) support is found in the restriction \[1\], \text{ka} has the consequence of introducing instances of existential quantification with \text{Close} ("q", #); while, \text{mo}, because of the presence of two applications of negation, will create instances of universal quantification. Note that occurrences of \text{Use} ("q", #) inside \[f\] will fail to contribute their presence because \text{Hide} ("q", #) scopes over \[f\].

We can illustrate applications of \text{ka} and \text{mo} with (23), which lead to the evaluation results of (24).

(23)

a. \[\text{[dare]}(\text{mo}/\text{ga})/([\text{nani}])(\text{ka/o}/\text{kaita})\]
b. \[\text{[hito]}(\text{mo}/\text{ga})/([\text{nani}])(\text{ka/o}/\text{kaita})\]
c. \[\text{[dono hito]}(\text{mo}/\text{ga})/([\text{nani}])(\text{ka/o}/\text{kaita})\]

(24)

a. \( (\lambda, (23a))^p = \neg \exists x \; (\text{person}(x) \land \neg \exists y \; (\text{thing}(y) \land \text{wrote}(x, y))) \)
b. \( \forall g : (g, \text{Garb}(0, ["q"], "c", (23b)))^p = * \)
c. \( (\lambda, (23c))^p = \neg \exists x \; (\text{person}(x) \land \neg \exists y \; (\text{thing}(y) \land \text{wrote}(x, y))) \)

These results illustrate how the presence of an indeterminate pronoun in the restriction of a noun phrase to which \text{mo} or \text{ka} are attached is an absolute requirement without which evaluation fails, as the case of (23b/24b) hows.\(^2\) The reason for this is that a case marker occurring with \text{no} or \text{ka}

\(^2\) More specifically, (24b) shows that evaluations of (23b) fail against all assignments that contain no "q" bindings. An evaluation of (23b) would technically be possible if the starting state of the assignment contained a single "q" binding. But the point of a quantified binding is that it must be created (by a quantificational
Quantification Support with Indeterminate Pronouns

will be primed to shift a “q” binding (as opposed to an “x” binding), but a “q” binding will only be created by the \text{Close}(“q”, #) that \text{mo} and \text{ka} contain given that sufficient \text{Use}(“q”, #) support is found for its creation, and such support only arises with the presence of an indeterminate pronoun in the local noun phrase restriction.

While the particles \text{mo} and \text{ka} and the indeterminate pronouns in (23a) and (23b) have a very close relationship, essentially a head-complement relation, the analysis does not require such close proximity. Interestingly, as noted by Kuroda (1965), Hoji (1985), Nishigauchi (1990), Watanabe (1992), Takahashi (1999), among others, Japanese data shows they can be separated from each other. For example, consider the examples of (25) from Takahashi (1999). In (25a) the particle and the indeterminate pronoun are together, with the complex serving as the subject in the relative clause. Of interest is (25b): while the indeterminate pronoun is confined in the relative clause, the particle is attached to the head noun of the relative clause, so that they apparently are not in the head-complement relation. Takahashi (1999) calls cases like (25b) “split QP sentences.”

(25)

a. \text{[[Dare mo-ga kaita] hon] -ga omosiroi}
   person q-NOM wrote book -NOM is-interesting
   ‘The book that everyone wrote is interesting.’

b. \text{[[Dare-ga kaita] hon] mo omosiroi}
   person-NOM wrote book q is-interesting
   lit. ‘Every book that a person wrote is interesting.’

\footnote{Note that previous examples have shown that in Japanese the subject and object are marked by the nominative \text{ga} and the accusative marker \text{o}, respectively. When they are attached to by quantificational particles in split QP sentences, these markers sometimes drop, as noted by Nishigauchi (1990). Thus, although the complex noun phrase functions as the subject in (25b), it is not accompanied by the nominative marker. We will not touch on why this should be the case here, and we will assume in the forms that we present for SCT evaluation that a \text{ga} marking is actually present.}
Using the method for introducing relative clauses presented at the end of section 3, we are able to encode (25a) as the SCT expression (26a), and (25b) as (26b). As (27a) and (27b) show, both expressions have evaluations, but they differ in the state of the assignment required to support evaluation. For (27a), \( g \) must contain a single “\( x \)” binding, but no other binding. This is reflected in the predicate logic translation that has a single free variable, namely \( x \). In addition, the translation contains the \( y \) variable bound to give a universal interpretation (owing to the presence of the negations) with a narrow scope, reflecting the embedded placement of the \( mo \) particle inside the relative clause. In contrast, an evaluation of (27b) requires the empty assignment, with all bindings in the resulting translation bound as a consequence of the high placement of \( mo \). Being quantificational, the primary contribution of \( mo \) is a “\( q \)” closure, but “\( x \)” closures also happen to be contributed with the presence of the negations that determine the quantificational force of the created bindings.

\[
\begin{align*}
(26) & \\
a. & [relc \text{"o"} ([dare]([mo/ga]/kaita), hon]ga \text{"x"}/omosiroi) & \\
    b. & [relc \text{"o"} ([dare]ga \text{"x"}/kaita), hon]([mo/ga]/omosiroi)
\end{align*}
\]

\[
\begin{align*}
(27) & \\
a. & \exists g: (g, (26a))^\circ = (\neg \exists y (person(y) \land \neg wrote(y, x)) \land book(x)) \land \text{is interesting}(x) & \\
    b. & (\lambda, (26b))^\circ = \neg \exists x \exists y (((person(x) \land wrote(x, y)) \land book(y)) \land \neg \text{is interesting}(y))
\end{align*}
\]

Note that the selectional restrictions seen with (23) persist with the split QP construction. Thus, in contrast to (25b), (28) is ungrammatical.

\[
\]

person-NOM wrote book \( q \) is-interesting

With (29) and (30) we can see that (28) gives rise to an SCT expression that has no evaluation.
5. Constituent questions and intervention

In the previous section we looked at the way quantification works in Japanese by employing indeterminate pronouns and particles. In this section we sketch how this same mechanism is used to establish constituent questions. As observed in section 1, in constituent questions indeterminate pronouns take on the role of WH-phrases and the -ka particle seemingly acts as a clause final question marker. We want to maintain the ka operation introduced as (21), so at best ka can only be part of a question forming operation, contributing a “q” closure. We will assume that the remainder is carried out by ?, (31).

(31)
\[ ? = \lambda \text{oper.} \lambda f. \]
\[ \text{Rel}(\text{nil}, \text{nil}, “\text{QUEST}”, [ \]
\[ \text{oper}(\lambda q \lambda f. \lambda f’. f)( \]
\[ \text{Rel}(\text{nil}, \text{nil}, “\text{EXIST}”, [ \]
\[ \text{Hide(“x”, Close(“x”, f))} )) ) ( \]
\[ \text{Rel}(\text{nil}, \text{nil}, “\text{”, nil})) ] \]

We can see examples of applying ? with ka in (32). These differ with: (32a) illustrating the binding of two indeterminate pronouns by ka; (32b), one; and (32c), none.

(32)
\[ a. [\text{dare}] ga “q”/[[\text{nani}] o “q”/kaita)(ka\?) \]
\[ b. [\text{hito}] ga “x”/[[\text{nani}] o “q”/kaita)(ka\?) \]
\[ c. [\text{hito}] ga “x”/[[\text{hon}] o “x”/kaita)(ka\?) \]
The resulting translations of (33) show how existential quantifiers within the direct scope of QUEST bind arguments that involve indeterminate pronouns, while existential quantifiers within the direct scope of EXIST bind arguments that do not involve indeterminate pronouns. While the end point of the current analysis is translation into expressions of predicate logic notation, we can expect the values for existentially quantified bindings created within the immediate scope of QUEST to be placed under question with an application of existential disclosure (see e.g., Dekker 1993, Honcoop 1998). Similarly, existential bindings under the influence of EXIST receive ordinary existential readings with existential disclosure.

Also note how (32c) is a valid expression with an evaluation despite the fact that ka does not create a quantified binding. This is because ka is applied at the level of the clause. It is only when attached to a case marker that ka must create a quantified binding for that case marker, as seen with the selectional restrictions of (23) and (28).

In addition to accounting for the possibility of ((multiple) constituent) questions, the current account predicts observations made by Hagstrom (1998) and Shimoyama (2006) that it is impossible to have a quantified particle intervene between an indeterminate pronoun and the instance of what should be its binding particle. This can be seen with (34) and the evaluations of (35), which shows (34a) to be invalid, in contrast to (34b) and (34c) where there is no particle intervention.
Quantification Support with Indeterminate Pronouns

(35)
a. \( \forall g: (g, (34a))° = * \)
b. \((\lambda, (34b))° = \text{QUEST}(\exists x \text{EXIST}((\text{thing}(x) \land \exists y (\text{person}(y) \land \text{wrote}(y, x)))) \)
c. \((\lambda, (34c))° = \text{QUEST}(\exists x \text{EXIST}((\text{person}(x) \land \exists y (\text{thing}(y) \land \text{wrote}(x, y)))) \)

In (34a) the intervening particle was a second instance of \( \text{ka} \). The example of (36a) shows that the same result of invalidity obtains when the intervening particle is \( \text{mo} \).

(36)
a. \([\text{dare}](\text{mo/ga}))/([\text{nani}]\text{o “q’/kaita})(\text{ka}\?) \)
b. \([\text{nani}]\text{o “q’/}[\text{dare}](\text{mo/ga}/\text{kaita})(\text{ka}\?) \)
c. \([\text{dare}]/\text{ga “q’/}[\text{nani}]\text{(mo/o)/kaita})(\text{ka}\?) \)

(37)
a. \( \forall g: (g, (36a))° = * \)
b. \((\lambda, (36b))° = \text{QUEST}(\exists x \text{EXIST}((\text{thing}(x) \land \neg \exists y (\text{person}(y) \land \neg \text{wrote}(y, x)))) \)
c. \((\lambda, (36c))° = \text{QUEST}(\exists x \text{EXIST}((\text{person}(x) \land \neg \exists y (\text{thing}(y) \land \neg \text{wrote}(x, y)))) \)

6. Concluding remarks

In this paper we have seen how a mechanism of quantification can be established with indeterminate pronouns and particles. Integration of the mechanism into a language came without any need to disrupt or change what was in place to ensure successful argument linking without quantification. More specifically, we saw indeterminate pronouns contribute usage information to trigger the creation of a “q” (quantified) binding. Particles themselves provided the means to create as many “q” bindings as they found support for. Having this set up resulted in a flexible means for establishing quantification, with the particles essentially taking on the role of scope marking operations to facilitate a high scope placement when
required, while placement of an indeterminate pronoun could thereafter be used to direct exactly where the contribution of the quantification created by the particle would be made available for integration into the clause. This directing role for the indeterminate pronoun, which is susceptible to intervention effects, became most apparent when the indeterminate pronoun and the particle were displaced, particularly when particles take up clausal positions, as in cases of constituent questions where indeterminate pronouns act as in-situ WH-phares.

**Appendix: SCT evaluation**

In this appendix we present the constrained evaluation procedure of Scope Control Theory we have assumed in this paper by way of a translation routine from expressions of a language with operators \(T, \text{Use}, \text{Hide}, \text{Close}, \text{Lam}, \text{Garb}\) and \(\text{Rel}\), into formulas of a predicate logic notation. The idea is that translation returns a ‘snapshot’ of the dependencies evaluation establishes.

Extensive use is made of sequences and operations on sequences. Notably:

- \([x_0, ..., x_{n-1}]\): a sequence with \(n\) elements, \(x_0\) being frontmost.
- \(x\): abbreviation for a sequence.
- \(\text{nil}\): the empty sequence.
- \(x_i\): the \(i\)-th element of a sequence, e.g. \([x_0, ..., x_{n-1}]_i = x_i\) where \(0 \leq i < n\).
- \(|x|\): the sequence length, e.g. \(|[x_0, ..., x_{n-1}]| = n\).
- \(\text{cons}(y, [x_0, ..., x_{n-1}]) = [y, x_0, ..., x_{n-1}]\).
- \(\text{snoc}(y, [x_0, ..., x_{n-1}]) = [x_0, ..., x_{n-1}, y]\).
- \(\text{foldl}\ f\ b\ [x_0, x_1, ..., x_{n-1}] = f (x_{n-1}, ..., f (x_1, f (x_0, b)))\).
- \(\text{map}\ f\ [x_0, x_1, ..., x_{n-1}] = [f(x_0), f(x_1), ..., f(x_{n-1})]\).
- \(\text{diff}(l, l')\) returns the elements of sequence \(l\) that are not in sequence \(l'\).

Translation is with respect to an assignment of a (possibly empty) sequence of predicate logic variables (scopes) to each binding name: \(g : \text{Name} \rightarrow \text{Var}^*\). The empty assignment, \(\lambda\), assigns \(\text{nil}\) to every binding name. We also require relations on assignments with pairs \((g, h)\) taking us...
from g to h or vice versa.

• \((g, h) \in \text{pop}_x\) iff h is just like g, except that, \(g(x) = \text{cons}((g(x))_0, h(x))\).

• \((g, h) \in \text{shift}(\text{op})_{x,y}\) iff \(\exists k: (h, k) \in \text{pop}_y\) and k is just like g, except that, \(g(x) = \text{op}((h(y))_0, k(x))\).

For \(\text{shift}(\text{op})\) the operation \(\text{op}\) needs to be specified, with suitable candidates being \(\text{cons}\) and \(\text{snoc}\), to give \(\text{shift}(\text{cons})\) and \(\text{shift}(\text{snoc})\). Relations are iterated when augmented with a positive superscript, e.g., \(\text{pop}^3_x\) iterates \(\text{pop}_x\) three times.

We now define a ‘usage count’ operation \(x(e)\). This formally defines the contribution of \(\text{Use}\) and \(\text{Hide}\), returning a count of the number of times \(\text{Use}(x, \#)\) occurs in expression e outside the scope of any \(\text{Hide}(x, \#)\).

\[
\begin{align*}
\text{x(Use}(y, e)) &= x(e) + 1 \text{ if } x = y, x(e) \text{ otherwise} \\
x(\text{There}(y, e)) &= x(e) - 1 \text{ if } x = y, x(e) \text{ otherwise} \\
x(\text{Hide}(y, e)) &= 0 \text{ if } x = y, x(e) \text{ otherwise} \\
x(\text{T}(y, i)) &= 0 \\
x(\text{Close}(y, e)) &= x(e) \\
x(\text{Lam}(y, z, e)) &= x(e) \\
x(\text{Garb}(n, y, z, e)) &= x(e) \\
x(\text{Rel}(y, z, r, e)) &= \sum_{i=0}^{x(e)-1} x(e_i)
\end{align*}
\]

A formal definition for the other operators is given below in terms of translation with respect to a scope sequence assignment, g.

\[
\begin{align*}
\text{(g, T}(x, i))^0 \text{ return } (g(x))_i, \text{ provided } 0 \leq i < |g(x)|; \text{ otherwise return } *.
\text{(g, Use}(x, e))^0 \text{ return } (g, e)^0.
\text{(g, Hide}(x, e))^0 \text{ return } (g, e)^0.
\text{(g, Close}(x, e))^0 \text{ if } x(e) = 0 \text{ return } (g, e)^0 \text{ else } \exists h: (h, g) \in \text{pop}_{x(e)}^0 \text{ \text{return } } \exists(h(x))_{x(e)-1}(h, e)^0, \text{ provided } (h, e)^0 \neq *; \text{ otherwise return } *.
\text{(g, Lam}(x, y, e))^0 \text{ given } \exists h: (g, h) \in \text{shift}(\text{cons})_{x,y} \text{ return } (h, e)^0; \text{ otherwise return } *.
\end{align*}
\]
• \((g, \text{Garb}(n, x, y, e))^\circ\) if \(|x| = 0\) return \((g, e)^\circ\) else \(\exists h_0...h_{|x|}: h_0 = g\) and for \(0 \leq i < |x|\), \((h_i, h_{i+1}) \in \text{shift}(\text{cons})_{x, y, b(x)}^{|x|} \cdot n\) return \((h_{|x|}, e)^\circ\); otherwise return *.

• \((g, \text{Rel}(x, y, r, e))^\circ\) return \(r((0, g, x, y, e)^\circ, ..., (|e|-1, g, x, y, e)^\circ)\), provided for \(0 \leq i < |e|\), \((i, x, y, e)^\circ \neq *\); otherwise return *.

• \((n, g, x, y, e)^\circ\) if \(|x| = 0\) return \((g, e)^\circ\) else \(\exists h_0...h_{|x|}: h_0 = g\) and for \(0 \leq i < |x|\), \((h_i, h_{i+1}) \in (\text{pop}_{|x|}^{|x|} \cdot n \cdot x)^\circ, \text{shift}(\text{snoc})_{x, y, n}^{|x|}, \Sigma_{k=0}^{n} x(e_k), \Sigma_{k=|e|-1}^{n} x(e_k))\) return \((h_{|x|}, e)^\circ\); otherwise return *.

Acknowledgements

For helpful comments and thought-provoking remarks, I would like to thank Kei Yoshimoto, two anonymous reviewers and the audience at the SPA/ICCS Workshop on Quantificational Phenomena in East Asian Languages.

References


Quantification Support with Indeterminate Pronouns


