A "Maximal Exclusion" Approach to Structural Uncertainty in Dynamic Syntax

Tohru Seraku
Dept. of Japanese Interpretation & Translation
Hankuk University of Foreign Studies
seraku@hufs.ac.kr

Abstract

"Case" and "grammatical function" are central to syntactic theories, but rigorous treatments of these notions in surface-oriented grammars like Dynamic Syntax (DS) are pending. Within DS, it is simply held that a case particle resolves structural uncertainty (i.e., unfixed node) in the course of incremental tree update. We model the relation between "case" and "grammatical function" with special reference to Japanese. In this language, the nominative case particle *ga* normally marks a "subject" NP, but it may mark an "object" NP. Moreover, *ga* may occur more than once within a single clause. We will address these issues by proposing the "maximal exclusion" approach to structural uncertainty.

1 Introduction

"Case" and "grammatical function" are central to any syntactic theories; a number of constructions exhibit unique case-marking patterns and linguistic generalisations are often stated with reference to grammatical function (Keenan and Comrie, 1979). Rigorous accounts of these concepts, however, are pending in "surface-oriented" grammars such as Dynamic Syntax (DS) (Kempson et al., 2001). The aim of this article is to clarify the relation between case and grammatical function in formal-grammar terms, with examples drawn from Japanese.

As will be stated in §2, the case-marking system of Japanese challenges surface-oriented grammars. In particular, DS, which explicates the mechanism whereby a string of words is parsed online and a structure is progressively built up, has not seriously tackled the relation between case and grammatical function (see §3). In this article, we advance the DS formalism from the perspective of "maximal exclusion" so that it models the relation between case and grammatical function in Japanese (see §4). We then apply this account to further data relating to "Major Subject Constructions" (see §5).

2 Case and Grammatical Function

In this article, we construe case and grammatical function in line with Comrie (1989).

Firstly, "case" is a morphological category. In Japanese, a case particle is typically attached to a noun (or a nominalised element).

(1) *Ken-ga* *ringo-o* *tabe-ta*
K-NOM apple-ACC eat-PAST
'Ken ate an apple.'

In (1), *ga* indicates that *Ken* bears a nominative case, while *o* indicates that *ringo* 'apple' bears an accusative case.

Secondly, "grammatical function" refers to a relation which an NP in a sentence has with respect to the predicate in the sentence. Examples include "subject," "object," and so on. These are abstract concepts, and they are identified based on syntactic tests in each language/dialect.

The focus of our enquiry is "subject." Keenan (1975) offers a set of universal "subject"-properties, although "subject" is captured gradably depending upon properties observed. The standard tests for subjecthood in Japanese are as follows (Kishimoto, 2004; Tsujimura, 2013; Tsunoda, 2009):

- α is a subject if it may be a target of a certain "honorification" operation.
- α is a subject if it may be an antecedent of the reflexive anaphor *jibun* 'self.'

Let us illustrate the former property with (2).

(2) *sensei-ga* *ringo-o* *otabeninat-ta*
teacher-NOM apple-ACC eat.HON-PAST
'That teacher ate an apple.'

In (2), the honorific form *otabeninat* 'eat' elevates the referent of *sensei* 'teacher.' *Sensei* is thus said to be a subject of the predicate *otabeninat*.

For some frameworks, grammatical function is a primitive concept. In Lexical-Functional Grammar, SUB, OBJ, etc. are postulated as "attributes" in the attribute-value matrices (Dalrymple, 2001). On the other hand, Dynamic Syntax (DS) dispenses with such primitive concepts; grammatical functions are defined structurally, as in the grammar models that have been developed in Chomsky (1965, 1995), etc. For instance, "subject" is structurally designated as follows: an element on the argument node which is immediately dominated by the root node is said to be a "subject" of the predicate in this structure.

(3) Schematic tree-structure

root

argument (subject) predicate

In DS, no serious attention has been paid to the issue of how case relates to grammatical function,[1] and it has been simply assumed that the nominative particle *ga* marks a subject NP (Cann, et al. 2005; Seraku, 2013). This stipulation may hold of (1)-(2), but it is unsustainable due to the following facts (Kuno, 1973; NKK, 2009; Shibatani, 1978):

- *Ga* may mark an object NP.
- *Ga* may occur several times in a single clause.
- A subject NP may be marked with *ni*, a dative particle (see §4.6).

These properties are not found in all verbs; the *ga*-marking of an object NP, for instance, is normally possible only with "stative" predicates (Koizumi, 2008; Kuno, 1973). The first two properties are illustrated in (4). (See §4-§5 for further data.)

(4) *watashi-ga* *ringo-ga* *tabe-tai* (*koto*)[2]
I-NOM apple-NOM eat-want (COMP)
'I want to eat an apple.'

This single clause has two occurrences of *ga*.[3] The second NP *ringo* 'apple' is not a subject because it lacks the "subject-properties," unlike *sensei* in (2) (Koizumi, 2008: 142-5). On the other hand, *ringo* in (4) is characterised as an object NP according to syntactic tests for objecthood (Kishimoto, 2004). Therefore, the simple correspondence between *ga* and "subject" cannot deal with data like (4), as has been a residual issue within DS.

3 Dynamic Syntax (DS)

3.1 Basics

DS models the process whereby the parser takes a string of words and gradually builds up a **semantic** structure. This mapping is direct in that syntactic structure is **not** postulated at any level. Within DS, "dynamic" refers to "online parsing," and "syntax" refers to an abstract system that maps a string onto a semantic structure in a progressive manner (Cann et al., 2005; Kempson et al., 2001, 2011).

For an illustration, the parse of the whole string (5) creates the semantic structure (6).

(5) *Ken-ga* *ne-ta*
K-NOM sleep-PAST
'Ken slept.'

(6) Final state (ignoring tense)

sleep'(*Ken'*) : t

Ken' : e *sleep'* : e→t

Each node conveys information about (i) semantic content such as *Ken'* and (ii) semantic type such as e ("entity" type). The node decorated with *Ken'* is at a "subject" position; a subject node is a type-e daughter of the root node in a propositional tree.

[1] An exception is Nakamura et al. (2009), which will be surveyed in §5. Kiaer (2014) also handles relevant data, but the formal details of her account are not clear.

[2] Without *koto*, (4) would sound better with the topic particle *wa* in place of the first instance of *ga* due to "exhaustivity" (Kuno, 1973). Such meaning disappears in embedded clauses, and scholars thus often put *koto* at the end of sentence. For the interests of brevity, we do not follow this practice in the rest of this article.

[3] *Ga* in *ringo-ga* is interchangeable with the accusative particle *o* in (4). The interchangeability is affected by various factors such as "style" and "transitivity" (Iori, 1995; Noda, 1996: 264-5), with cross-speaker variations (Shibatani, 1978: 230-2).

A tree is binary; a left-hand node is an argument node, and a right-hand node is a functor node. For instance, the right daughter of the root in (6) is a functor node, which takes the type-e content *Ken'* and returns the type-t content *sleep'*(*Ken'*).

A tree update starts with the AXIOM (7).

(7) AXIOM

?t

At this initial stage, there is only a root node, and it is annotated with ?t. ?t is a "requirement" that this node will be decorated with a type-t content. The parser executes **general** and **lexical** actions to meet requirements until no outstanding requirements are left in the tree.

General action. General actions are tree update actions whose applications are not triggered by the parse of a lexical item. If *Ken-ga ne* in (5) is parsed, it yields the semantic tree (8).

(8) Parsing *Ken-ga ne*

?t

Ken' : e   *sleep'* : e→t

As each daughter node is specified for content and type, the parser may perform functional application. This is not lexically triggered, and it is formalised as the general action ELIMINATION. The execution of this action outputs (6). (The tense suffix -*ta* is disregarded in this article.)

Lexical action. Each lexical item encodes a set of actions for tree update. Consider (9).

(9) *ne-ta*
sleep-PAST
'Someone (or a salient person) slept.'

Japanese is a "pro-drop" language; argument NPs may be covert as long as they are retrievable in context. It is then assumed in DS that the parse of a verb projects a propositional template. For instance, *ne* 'sleep' encodes a set of actions to project the propositional template (10).

(10) Parsing *ne*

?t

U : e   *sleep'* : e→t

A subject node is decorated with a metavariable U, a placeholder to be saturated. If Ken is a salient person in context, U is saturated as *Ken'*.

3.2 Structural Uncertainty

Each node is assigned a label for a node position, with the "tree-node" predicate *Tn* which takes a numeral as argument (Cann et al., 2005).

(11) Node-Position Labelling

Tn(0)

Tn(00)   *Tn*(01)

Tn(010)   *Tn*(011)

When a node is assigned a numeral "α," its left daughter is assigned "α0" and its right daughter "α1." Since the root receives "0," its left daughter receives "00" and its right daughter "01."

Let us then introduce LOCAL *ADJUNCTION, a general action to posit a node whose position in a tree is initially uncertain and needs to be resolved within a local structure.

(12) LOCAL *ADJUNCTION

?t, *Tn*(0)

Ken' : e, $<\uparrow_{01*}>$(*Tn*(0))

In $<\uparrow_{01*}>$(*Tn*(0)), "1*" is an arbitrary succession of "1" (including none). $<\uparrow_{01*}>$(*Tn*(0)) means: if you go up from an argument node by one node (and optionally keep going up through functor nodes), you will reach the root node, as marked with *Tn*(0) (Blackburn and Meyer-Viol, 1994). In (12), the dashed line visually displays structural uncertainty. $<\uparrow_{01*}>$(*Tn*(0)) indicates that this node is at some argument position within a local structure although the exact position is uncertain at this point.

Structural uncertainty may be fixed in two ways: (i) the general action of UNIFICATION (see §4.2) or (ii) lexical actions encoded in a case particle. As for (ii), it has previously been held that the parse of a case particle resolves an unfixed node (Cann et al., 2005; Seraku, 2013). The nominative particle *ga*, for instance, has been assumed to resolve an unfixed node as a "subject" node. (This analysis is similar to the "constructive case" analysis within LFG (Nordlinger, 1998).)

This past DS analysis of case particles, however, encounters the problem mentioned in the paragraph around (4). In the next section, we will abandon this previous view of case particles, and propose an alternative approach.

4 A "Maximal Exclusion" Approach

4.1 Informal Sketch

It has been held in DS that a case particle **uniquely determines** a landing site for an unfixed node (Cann et al., 2005). In this article, we propose that a case particle reduces the range of landing sites by **maximally excluding** potential sites modulo the limitations imposed by each case particle.

(13) Proposal: General Claim

a. A case particle excludes all landing sites for an unfixed node but a few candidates.

b. Such "candidates" differ depending on the type of a case particle.

Thus, a case particle may not immediately resolve an unfixed node. If the number of potential landing sites is reduced to one, however, it will amount to immediate resolution. (13) is consonant with the central DS view: a tree is gradually built up, with various constraints posited by general and lexical actions constraining the way the tree grows.

Concerning (13)b, we assume (14) for *ga*.

(14) Proposal: Nominative Particle *Ga*

a. *Ga* excludes all but a subject node **and** an object node.

b. If the above exclusion has already occurred, further exclusion occurs: exclude all but a subject node **or** an object node (not both).

(14) will be illustrated in §4.2-§4.5 (and formalised in the Appendix). Further, other case particles than *ga* are briefly discussed in §4.6.

4.2 Nominative Particle (Part I)

Suppose the parser processes the string (15). At the time of parsing *Ken*, the tree (16) has been built up. (Other *Tn*-statements than *Tn*(0) are omitted in this and subsequent tree displays.)

(15) *Ken-ga ne-ta*
K-NOM sleep-PAST
'Ken slept.'

(16) Parsing *Ken*

?t, *Tn*(0)

Ken' : e, $\langle\uparrow_{01*}\rangle(Tn(0))$

$\langle\uparrow_{01*}\rangle(Tn(0))$ specifies the set of constraints (17).

(17) $\{\langle\uparrow_0\rangle(Tn(0)), \langle\uparrow_{01}\rangle(Tn(0)), \langle\uparrow_{011}\rangle(Tn(0)) \ldots\}$

Recall that $\langle\uparrow_0\rangle(Tn(0))$ refers to a subject position, $\langle\uparrow_{01}\rangle(Tn(0))$ refers to an object position, and so on. Thus, (17) indicates that an unfixed node may be fixed at **any** argument position within a local tree.

The next element is *ga*. According to (14)a, *ga* excludes all but a subject and an object node.

(18) Parsing *Ken-ga*

?t, *Tn*(0)

Ken' : e, $\langle\uparrow_{0(1)}\rangle(Tn(0))$

"(1)" in $\langle\uparrow_{0(1)}\rangle(Tn(0))$ means that the presence of "1" is optional, as delineated in (19).

(19) $\{\langle\uparrow_0\rangle(Tn(0)), \langle\uparrow_{01}\rangle(Tn(0))\}$

Unlike (17), (19) indicates that an unfixed node may be fixed at a subject or an object node (but not other nodes). In this way, the parse of *ga* tightens the constraint $\langle\uparrow_{01*}\rangle(Tn(0))$ to $\langle\uparrow_{0(1)}\rangle(Tn(0))$.

The rest of the process is as usual: the parse of *ne* 'sleep' yields the tree (20) (cf., (10)).

(20) Parsing *Ken-ga ne*

?t, *Tn*(0)

U : e, $\langle\uparrow_0\rangle(Tn(0))$ *sleep'* : e→t

Ken' : e, $\langle\uparrow_{0(1)}\rangle(Tn(0))$

The intransitive verb *ne* creates a subject node, which is marked with $\langle\uparrow_0\rangle(Tn(0))$. UNIFICATION, then, merges this subject node with the unfixed node. (UNIFICATION is a general action to combine a description of an unfixed node with that of a fixed node of the same type; see §3.2.)

(21) UNIFICATION

?t, *Tn*(0)

Ken' : e, $\langle\uparrow_0\rangle(Tn(0))$ *sleep'* : e→t

ELIMINATION (i.e., functional application) outputs the final state; see (6) in §3.1.

4.3 Nominative Particle (Part II)

Let us turn to example (22).

(22) *Ken-ga ringo-o tabe-ta*
K-NOM apple-ACC eat-PAST
'Ken ate an apple.'

After *Ken-ga* is processed (see (18)), the parse of *ringo-o* engenders (23). (The parse of *o* resolves an unfixed node at an object position; see §4.6.)

(23) Parsing *Ken-ga ringo-o*

?t, *Tn*(0)

Ken′ : e, $<\uparrow_{0(1)}>(Tn(0))$?(e→t)

apple′ : e

The parse of *tabe* ‘eat’ then builds a propositional template, as in (24).

(24) Parsing *Ken-ga ringo-o tabe*

?t, *Tn*(0)

Ken′ : e, $<\uparrow_{0(1)}>(Tn(0))$ U: e ?(e→t)

apple′ : e *eat′* : e→(e→t)

The parse of *tabe* creates a subject node. This node is compatible with the constraint $<\uparrow_{0(1)}>(Tn(0))$ of the unfixed node. Thus, UNIFICATION may be run, merging the description of the unfixed node with that of the subject node. After ELIMINATION is run, the final state emerges.

(25) UNIFICATION + ELIMINATION

eat′(*apple′*)(*Ken′*) : t, *Tn*(0)

Ken′ : e *eat′*(*apple′*) : e→t

apple′ : e *eat′* : e→(e→t)

4.4 Nominative Particle (Part III)

Let us then examine (26), repeated from (4).

(26) *watashi-ga ringo-ga tabe-tai*
I-NOM apple-NOM eat-want
‘I want to eat an apple.’

The parse of *watashi-ga* is as usual, and the parse of the next item *ringo* ‘apple’ yields (27). (*Sp′* is informally used for the content of *watashi* ‘I.’)

(27) Parsing *watashi-ga ringo*

?t, *Tn*(0)

Sp′ : e, $<\uparrow_{0(1)}>(Tn(0))$ *apple′* : e, $<\uparrow_{01*}>(Tn(0))$

In (27), the exclusion stated in (14)a occurs. Thus, according to (14)b, the parser excludes all potential landing sites for an unfixed node but a subject or an object position. If the parser chooses to exclude all but an **object** position, $<\uparrow_{01}>(Tn(0))$ is posited at the unfixed node for *ringo*. That is, the unfixed node for *ringo* is resolved as the object node.

(28) Parsing *watashi-ga ringo-ga*

?t, *Tn*(0)

Sp′ : e, $<\uparrow_{0(1)}>(Tn(0))$?(e→t)

apple′ : e, $<\uparrow_{01}>(Tn(0))$

The remainder of the parse process is as outlined in the last subsection.

At the stage (27), the parser could have excluded all but a **subject** position as a landing site for the unfixed node for *ringo*. If this exclusion happened, the unfixed node for *watashi* would be licensed at an object position, giving rise to the interpretation ‘An apple wants to eat me.’ This tree update itself is legitimate, but the resulting interpretation would be blocked on semantic grounds.

In this respect, noteworthy is (29).

(29) *Ken-ga Naomi-ga sukida*
K-NOM N-NOM like
a. ‘Ken likes Naomi.’
b. ‘Naomi likes Ken.’

The parse of *Ken-ga Naomi* outputs (30).

(30) Parsing *Ken-ga Naomi*

?t, *Tn*(0)

Ken′ : e, $<\uparrow_{0(1)}>(Tn(0))$ *Naomi′* : e, $<\uparrow_{01*}>(Tn(0))$

In parsing *ga* in *Naomi-ga*, if the parser chooses to put $<\uparrow_{01}>(Tn(0))$ at the unfixed node for *Naomi*, the node is resolved as the object node. This leads to the “a”-interpretation. If $<\uparrow_{0}>(Tn(0))$ is posited at the unfixed node for *Naomi*, the node is resolved as the subject node, and the “b”-reading arises.

4.5 Nominative Particle (Part IV)

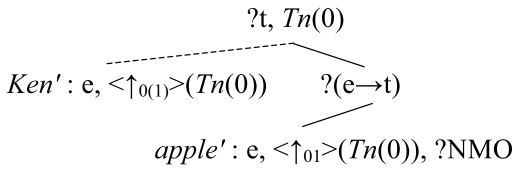
The proposed account is still not complete. The *ga*-marking of an object NP is usually allowed only by stative predicates (see §2). Thus, (31), where *kat* ‘buy’ is an action verb, is ungrammatical.

(31) **Ken-ga ringo-ga kat-ta*
K-NOM apple-NOM buy-PAST
Int. ‘Ken bought an apple.’

The account developed thus far does not rule (31) out because the possibility of the *ga*-marking of an object NP is dependent on the type of predicate.

We thus assume that if *ga* marks an object NP, this case-marking fact is recorded, which will be checked by a forthcoming predicate. In (31), when *ringo-ga* is parsed, it puts ?NMO at an object node.

(32) Parsing *Ken-ga ringo-ga*



?NMO (Nominative Marking of Object) must be checked by a predicate that allows the *ga*-marking of an object NP. (This constraint is encoded in the entries for stative predicates.) In (31), *kat* 'buy' disallows such *ga*-marking. ?NMO is thus not met, and (31) becomes ungrammatical.

The above idea is summarised in (33); see also the Appendix for formalisation.

(33) Proposal: Record of Object Marking
If *ga* excludes all but an object node, the object node is annotated with ?NMO.

4.6 Other Case Particles

According to our general proposal (13), a case particle excludes all landing sites for an unfixed node but a few candidates, and such candidates are encoded in each particle. Below, we touch on the accusative particle *o* and the dative particle *ni*.

The accusative particle *o* typically marks an NP which bears the semantic role "theme"; see *ringo* 'apple' in (22). The accusative particle *o* may also mark an NP bearing the semantic role "path" (34) or "departure site" (35) (NKK, 2009: 67-70).

(34) *Ken-ga* *sono-yama-o* *koe-ta*
K-NOM that-mountain-ACC pass-PAST
'Ken passed that mountain.'

(35) *Ken-ga* *ie-o* *de-ta*
K-NOM house-ACC leave-PAST
'Ken left a house.'

In the light of the "double-*o* constraint" (Harada, 1973), Shibatani (1978: 289-92) shows that the *o*-marked NPs as in (34)-(35) have the grammatical function of "object." Setting aside complex issues,[4] we thus hold that *o* always marks an object NP.

(36) Proposal: Accusative Particle *O*
O excludes all but an object node.

(36) amounts to immediately resolving an unfixed node as an object node. So, as far as *o* is concerned, our "maximal exclusion" approach converges with the "unique-determination" approach (Cann et al., 2005; Seraku, 2013).

The dative particle *ni* usually marks an indirect-object NP (37), but in some environments, *ni* may mark a subject NP (38).

(37) *Ken-ga* *Naomi-ni* *ringo-o* *age-ta*
K-NOM N-DAT apple-ACC give-PAST
'Ken gave an apple to Naomi.'

(38) *Ken-ni* *eigo-ga* *wakaru*
K-DAT English-NOM understand
'Ken understands English.'

From the "maximal exclusion" perspective, then, we assume (39).

(39) Proposal: Dative Particle *Ni*
a. *Ni* excludes all but a subject node **and** an Indirect Object (IO) node.
b. If such exclusion has already been present, further exclusion occurs: exclude all but a subject node **or** an IO node (not both).

Two caveats are in order. First, the *ni*-marking of a subject NP is not possible with all predicates, and the possibility of such *ni*-marking must be encoded in each predicate (Shibatani, 1978: 224).[5] Second, although *ni* appears in other contexts (NKK, 2009), *ni* in these environments would be characterised as postpositions, such as *ni* 'at' and *ni* 'to.'

In this section, we have re-considered the role of case particles in structure building from the angle of "maximal exclusion."[6]

[4] First, *o* may mark an adverbial element (Mihara, 1994). This use of *o* would be an instance of the postposition *o*. Second, *o* is said to appear in "small clauses" or "ECM" constructions, but their theoretical status is contentious (Kawai, 2008; Kuno, 1976).

[5] The set of predicates allowing "SUB-*ni* OBJ-*ga*" is a proper subset of the set of predicates allowing "SUB-*ga* OBJ-*ga*" (Kuno, 1973: §4). ("SUB" means a subject NP, and "OBJ" an object NP.) For predicates allowing the *ni*-marking of SUB, we assume: if *ni* excludes all but a subject node, the subject node is annotated with ?DMS (Dative Marking of Subject); cf., (33).

[6] Case particles also appear in head-internal relatives (Kuroda, 2005). Within DS, this construction has been analysed in Seraku (2013), and our account of *ga*, *o*, and *ni* is compatible with Seraku's analysis.

5 Further Issues

Turning back to multiple occurrences of *ga*, let us explore MSC (Major Subject Construction) of the type (40) (Kuroda, 1992: 248). Noda (1996: 257-9) mentions other kinds of MSC, but (40) represents the most discussed type of MSC.

(40) *Ken-ga* *imouto-ga* *yasashii*
K-NOM younger.sister-NOM sweet
'Ken's younger sister is sweet.'

The first *ga*-marked item *Ken*, often called "major subject," acts as a possessor NP of the second *ga*-marked item *imouto* 'younger sister.' In fact, some scholars claim to derive (40) from (41), where *no* in *Ken-no* is a genitive case particle (e.g., Kuno's (1973: §3) "subjectivisation").

(41) *Ken-no* *imouto-ga* *yasashii*
K-GEN younger.sister-NOM sweet
'Ken's younger sister is sweet.'

5.1 Previous DS Account

In DS, Nakamura et al. (2009) focusses on the type of MSC shown in (40). (They do not address the data in §4.4-§4.5.) Their analysis is as follows:

- *Ga* does not resolve structural uncertainty, but just lets the parser return to the root node.
- Before a second *ga*-marked item is parsed, the general action of GENERALISED ADJUNCTION sets an unfixed ?t-node, under which a second *ga*-marked item is parsed.
- A second *ga*-marked item is a relational noun which creates a complex structure, into which the unfixed node for the first *ga*-marked item is incorporated by means of UNIFICATION.

In their analysis, while an unfixed node for the first *ga*-marked item requires that it be fixed in a **local** tree, an unfixed node introduced by GENERALISED ADJUNCTION requires that it be fixed **anywhere** in the whole tree. Presumably to avoid this problem, Nakamura et al. (2009: 114) resort to "structural abduction" (Cann et al., 2005: 256). But such an abduction step cannot occur in their proposed tree, since it ends up identifying the unfixed node for the first *ga*-item with that for the second *ga*-item, leading to inconsistency of node descriptions. Thus, their analysis is formally illegitimate.

5.2 Alternative DS Account

Our alternative account holds that *ga* is ambiguous between *ga* (14) and *ga* for "major subject" which we will propose by utilising Seraku and Ohtani's (2016) analysis of the genitive particle *no*.

Let us illustrate the analysis of *no* with (42). The parse of *Ken-no* derives the tree state (43).

(42) *Ken-no hon*
K-GEN book
'Ken's book' ('a book which Ken possesses,' 'a book which Ken wrote,' etc.)

(43) Parsing *Ken-no*

Ken' : e $\curvearrowright$ $U_{R(Ken', U)}$: e

$U_{R(Ken', U)}$ must be saturated with a semantic content in relation R to *Ken*. R is contextually specified as a "possession" relation, for example. The curved arrow represents a "LINK" relation (Cann et al., 2005: Ch. 3). LINK connects two structures, given a shared term like *Ken'*. When the next item *hon* 'book' is parsed, the tree is updated into (44).

(44) Parsing *Ken-no hon*

Ken' : e $\curvearrowright$ $book'_{POSS(Ken', book')}$: e

$book'_{POSS(Ken', book')}$ denotes a book which stands in a possession relation to Ken.[7]

A metavariable $U_{R(Ken', U)}$ is used in (43) since *Ken-no* itself may denote an entity.

(45) *Ken-no/*-ga*
K-GEN/-NOM
'Ken's'

For *Ken-no*, $U_{R(Ken', U)}$ is saturated pragmatically (rather than by the parse of *hon* 'book' as in (42)).

Another notable point is that *imouto* 'younger sister' in (40) is a relational noun which takes an individual x and denotes the sister(s) of x. We view "relational nouns" broadly so as to include nouns for which a relation can be contextually set out.

(46) *Ken-ga* *ie-ga* *goukada*
K-NOM house-NOM gorgeous
'Ken's house is gorgeous.'

We will thus define the actions encoded in *ga* (for major subjects) by reflecting the following:

[7] Formally, terms are expressed in the epsilon calculus: (ε, x, *book'*(x)&*poss'*(x)(*Ken'*)) for $book'_{POSS(Ken', book')}$.

- A post-*ga* NP must be overtly present.
- A post-*ga* NP is a "relational" noun (at least, for the type of MSC illustrated in (40)).

Our contention is that the parse of *Ken-ga* in (40) yields the tree (47).

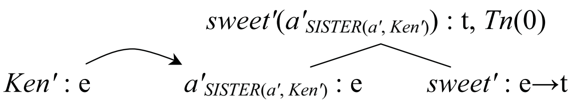
(47) Parsing *Ken-ga*

$?\exists x.Fo(x_{R(x,\ Ken')})$ requires that this node will be decorated with a content in relation *R* to *Ken′*. (*Fo* is a "formula" predicate (Cann et al., 2005).) This requirement lacks a metavariable U, and data such as *Ken-ga* in (45) are ruled out. The requirement is fulfilled by the parse of *imouto* 'younger sister,' as shown in (48). $a'_{SISTER(a',\ Ken')}$ denotes an individual *a′* who is in a sister relation to Ken.[8]

(48) Parsing *Ken-ga imouto*

The rest of the parse process is as outlined in §4.2. The final state is given in (49).

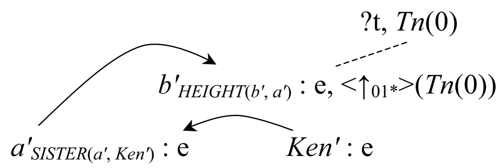
(49) Parsing *Ken-ga imouto-ga yasashii*



Note that the tree update triggered by the parse of a major subject may occur more than once. For instance, the parse of *Ken-ga imouto-ga se* in (50) gives rise to (51), where $b'_{HEIGHT(b',\ a')}$ represents the height of the individual who is the sister of Ken.[9]

(50) *Ken-ga* *imouto-ga* *se-ga* *takai*
K-NOM sister-NOM height-NOM high
'Ken's younger sister's height is high.'

(51) Parsing *Ken-ga imouto-ga se*



[8] Formally, (ι, x, *sister′*(*Ken′*)(x)).
[9] Formally, (ι, x, *height′*(ι, y, *sister′*(*Ken′*)(y))(x)).

Further, $a'_{SISTER(a',\ Ken')}$ is composed reflecting the order in which *Ken* is first parsed and then *imouto* 'younger sister' follows. Consider (52).

(52) * *imouto-ga* *Ken-ga* *yasashii*
sister-NOM K-NOM sweet
Int. 'Ken's younger sister is sweet.'

(52) is ruled out since *Ken* cannot denote a relation, unlike *imouto*, which denotes the relation *SISTER* so that composite terms like $a'_{SISTER(a',\ Ken')}$ are created.

As a residual issue, *ga* may be used as a genitive particle, but such examples are archaic (Frellesvig, 2011). Although our treatment of *ga* (for major subjects) allows (53), it is not obvious if we should posit further constraints to block such examples. (It is also notable that in many Ryukyuan languages, the nominative particles have the genitive-marking function, too (Tohyama and Seraku, in press).)

(53) *warera-ga michi*
we-GEN road
'Our road' (with an archaic flavour)

6 Conclusion

We have presented a maximal-exclusion approach to structural uncertainty. It is an open issue if this approach is applicable to data on languages other than Japanese (Koizumi, 2008: 142). It would also be essential to explore if the proposed view of case may be incorporated into other "realistic" grammar models (Sag and Wasow, 2011).

Appendix. Entries for Case Particles

A lexical entry specifies a set of actions to be run in conditional format (Cann et al., 2005). For space reasons, the entry for *ga* alone is presented here.

IF e, $<\uparrow_{01*}>(Tn(0))$
THEN IF $<\uparrow_{01*}><\downarrow_{1*0}>(Tn(U), \exists x.Tn(x), <\uparrow_{0(1)}>(Tn(0)))$
THEN `put`($<\uparrow_{0}>(Tn(0))/<\uparrow_{01}>(Tn(0))$, ?NMO)
ELSE `put`($<\uparrow_{0(1)}>(Tn(0))$)
ELSE `abort`

Acknowledgments

This article largely benefitted from the comments provided by the three anonymous PACLIC referees, Ruth Kempson, and Akira Ohtani. This work was supported by the Hankuk University of Foreign Studies Research Fund of 2016.

References

Blackburn, P. and Meyer-Viol, W. 1994. Linguistics, logic and finite trees. *Logic Journal of the IGPL* 2 (1), 3-29

Cann, R., Kempson, R., and Marten, L. 2005. *The Dynamics of Language*. Oxford: Elsevier.

Chomsky, N. 1965. *Aspects of the Theory of Syntax*. Cambridge, MA: MIT Press.

Chomsky, N. 1995. *Minimalist Program*. Cambridge, MA: MIT Press.

Comrie, B. 1989. *Language Universals and Linguistic Typology*, 2nd edn. Chicago, IL: The University of Chicago Press.

Dalrymple, M. 2001. *Lexical Functional Grammar*. New York: Academic Press.

Frellesvig, B. 2011. *A History of the Japanese Language*. Cambridge: Cambridge University Press.

Harada, S.-I. 1973. Counter equi-NP deletion. *Annual Bulletin Research Institute of Logopaedics and Phoniatrics* 7, 113-48.

Iori, I. 1995. "Ga shitai" to "o shitai." *Nihongo Kyoiku* 86, 52-64.

Kawai, M. 2008. Alleged small clauses in Japanese. *Toronto Working Papers in Linguistics* 28, 89-105.

Keenan, E. 1975. Towards a universal definition of "subject." In Li, C. (ed.) *Subject and Topic*. New York: Academic Press, pp. 303-33.

Keenan, E. and Comrie, B. 1977. Noun phrase accessibility and universal grammar. *Linguistic Inquiry* 8 (1), 63-99.

Kempson, R., Gregoromichelaki, E., and Howes, C. (eds.) 2011. *The Dynamics of Lexical Interfaces*. Stanford, CA: CSLI Publications.

Kempson, R., Meyer-Viol, W., and Gabbay, D. 2001. *Dynamic Syntax*. Oxford: Wiley-Blackwell.

Kiaer, J. 2014. *Pragmatic Syntax*. London: Bloomsbury Publishing.

Kishimoto, H. 2004. Transitivity of ergative-marking predicates in Japanese. *Studies in Language* 28 (1), 105-36.

Koizumi, M. 2008. Nominative object. In Miyagawa, S. and Saito, M. (eds.) *The Oxford Handbook of Japanese Linguistics*. Oxford: Oxford University Press, pp. 141-64.

Kuno, S. 1973. *The Structure of the Japanese Language*. Cambridge, MA: MIT Press.

Kuno, S. 1976. Subject raising. In Shibatani, M. (ed.) *Syntax and Semantics*, Vol. 5. New York: Academic Press, pp. 17–41.

Kuroda S.-Y. 1992. What can Japanese say about government and binding? In Kuroda, S.-Y. (ed.) *Japanese Syntax and Semantics*. Dordrecht: Kluwer, pp. 40-52.

Kuroda, S. Y. 2005. *Nihongo-kara Mita Seisei Bunpou*. (Generative grammar from the viewpoint of Japanese) Tokyo: Iwanami Shoten.

Mihara, K. 1994. *Nihongo-no Tougo Kouzou*. (Syntactic structure of Japanese) Tokyo: Shohakusha.

Nakamura, H., Yoshimoto, K., Mori, Y., and Kobayashi, M. 2009. Multiple subject construction in Japanese. In Hattori, H. et al. (eds.) *New Frontiers in Artificial Intelligence (LNAI 5447)*. Dordrecht: Springer, pp. 103-18.

NKK (Nihongo Kijyutsubunpou Kenkyuukai). 2009. *Gendai Nihongo Bunpou*, Vo. 2. (The grammar of modern Japanese) Tokyo: Kuroshio Publishers.

Noda, T. 1996. "*Wa*" *to* "*Ga*." ("Wa" and "ga") Toyko: Kuroshio Publishers.

Nordlinger, R. 1998. *Constructive Case*. Stanford, CA: CSLI Publications.

Sag, I. and Wasow, T. 2011. Performance-compatible competence grammar. In Borsley, R. and Borjars, K. (eds.) *Non-transformational Syntax*. Oxford: Wiley-Blackwell, pp. 359-77.

Seraku, T. 2013. *Clefts, Relatives, and Language Dynamics*. DPhil thesis, University of Oxford.

Seraku, T. and Ohtani, A. 2016. The word-order flexibility in Japanese novels. In Ogata, T. and Akimoto, T. (eds.) *Computational and Cognitive Approaches to Narratology*. Hershey, PA: IGI Global, pp. 213-44.

Shibatani, M. 1978. *Nihongo-no Bunseki*. (The analysis of Japanese) Tokyo: Taishukan Publishing Company.

Tohyama, N. and Seraku, T. in press. Towards a description of the case system of Yoron Ryukyuan. *International Journal of Okinawan Studies*.

Tsujimura, N. 2013. *An Introduction to Japanese Linguistics*, 3rd edn. Oxford: Wiley-Blackwell.

Tsunoda, T. 2009. *Sekai-no Gengo-to Nihongo*, 2nd edn. (The languages of the world and Japanese) Tokyo: Kuroshio Publishers.