平成１２年度 COE 形成基礎研究費研究成果報告（５）（課題番号　08CE1001）

「先端的言語理論の構築とその多角的な実証（５）
－ヒトの言語を組み立て演算する能力を語彙の意味概念から探る－」（抜刷）
Grant-in-Aid for COE Research Report (5)（No. 08CE1001）
Researching and Verifying an Advanced Theory of Human Language:
Explanation of the human faculty for constructing and computing
sentences on the basis of lexical conceptual features
March, 2001

Japanese PAPPI

Sandiway Fong*

NEC Research Institute, Inc.
4 Independence Way
Princeton NJ, USA
sandiway@research.nj.nec.com

Abstract

PAPPI is a multilingual, Prolog-based parsing system designed for implementing theories in the Principles-and-Parameters framework. This report describes the Japanese version of the system. In particular, it reviews how the core system can be extended to handle a variety of Japanese phenomena, including anti-superiority, indirect passives and potentials, *o*/*ni*-causative and dative subject constructions, and the double-*o* constraint.

1 Introduction

PAPPI is a multilingual parsing engine in the Principles-and-Parameters framework (Chomsky, 1981), initially developed at MIT (Fong, 1991) and greatly expanded upon at NEC. It consists of a core engine written in Prolog, a Horn clause logic-based programming language originally designed for natural language processing (Colmerauer *et al.*, 1973), containing both a module to recover phrase structure and a set of linguistically-motivated primitives for expressing structural constraints imposed by linguistic theory. The basic system written for English implements classic Goverment-and-Binding theory as described in (Lasnik & Uriagereka, 1988), together with selected parameterized extensions for other languages, one of these being Japanese. At Kanda, the Japanese system was revised and extended to cover a variety of linguistic phenemona. This report is divided into two main sections. In section 2, we will review the basic elements of the syntactic theory as implemented in the parser. Next, in the main part of the report, we will describe the additional constraints and parsing machinery necessary to handle the Kanda Extensions. Finally, in section 4, we conclude by discussing the implications of these revisions on PAPPI's computational architecture.

2 Background

At the heart of PAPPI's computational apparatus is a backtracking, all-solutions logic-based engine that (1) takes an input sentence and assigns one or more underspecified parse trees compatible with the input, (2) applies structurally-defined constraints to fill out underspecified structures with syntactic features and filter out ill-formed configurations from a variety of syntactic modules, and (3) transforms surviving parse trees via Quantifier Raising (QR) and Logical Form (LF) movement into syntactic LF trees. The PAPPI engine is designed to pursue all possible computational paths, and thus will return multiple LF parses in both cases of theoretical

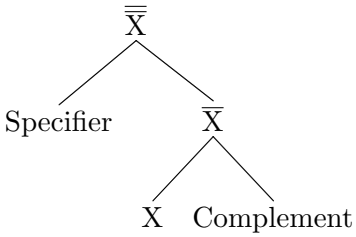
*The author wishes to acknowledge the many contributions of the Kanda COE project members who made the implementation described in this report possible; including Kazuko Inoue, Nobuko Hasegawa, Yukiko Ueda, Kazuma Fujimaki and Yukio Furukawa.

incompleteness and genuine ambiguity (relevant examples will be supplied throughout the report). A brief review of the modules of grammar present in the underlying system, highlighting areas of importance for Japanese, follows.

2.1 $\overline{\text{X}}$-syntax

We assume phrase structure is uniformly described by binary-branching $\overline{\text{X}}$-phrase structure rules of the following form:

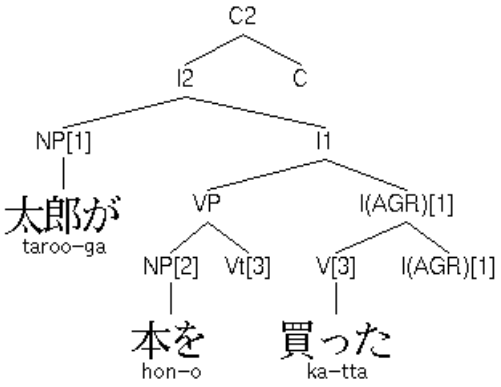
(1) $\overline{\overline{\text{X}}} \rightarrow$ Specifier $\overline{\text{X}}$
$\overline{\text{X}} \rightarrow$ X Complement



X ranges over nouns (N), verbs (V), adjectives (A), pre- or post-positions (P), negation (neg), inflection (I) and complementizer (C). Specifier and Complement positions are occupied by maximal projections ($\overline{\overline{\text{X}}}$/XP). The complementizer phrase (CP) defines a clause, the specifier position of which is reserved for *wh*-headed NPs and adverbs. A CP selects for an IP complement. The specifier of IP is reserved for the surface subject. IP selects for a verb phrase (VP), or negP when sentential negation is present. Objects reside in the complement of VP position. (In the basic implementation, the VP-specifier position is unused.) Furthermore, basic word order is accommodated by Specifier-Head and Head-Complement order. For SVO languages like English, the Specifier precedes, and the Complement follows, the Head. For SOV languages like Japanese, the Head follows both the Specifier and Complement. An example of basic Japanese sentential structure is given in (2):

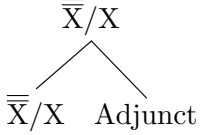
(2)

Taroo-ga	hon-o	katta
Taroo-NOM	book-ACC	bought



Adjunction is also permitted at the maximal and head level:

(3) $\overline{\overline{\text{X}}} \rightarrow \overline{\overline{\text{X}}}$ Adjunct
$\text{X} \rightarrow \text{X}$ Adjunct



Adjuncts are restricted to heads at the X-level, and to maximal projections at the $\overline{\overline{\text{X}}}$-level. The parse in (2) contains an example of head adjunction. The verbal head has adjoined to inflection forming *katta* (buy+PAST).

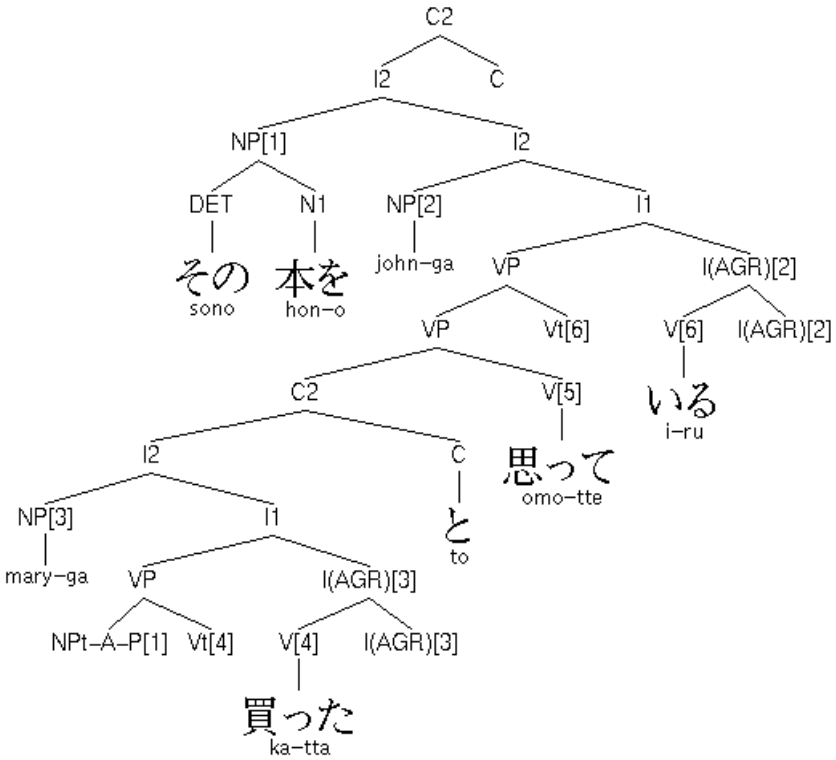
Phrase structure recovery in PAPPI is implemented by a backtracking LR(1) parser (Knuth, 1965), which builds candidate phrase structures in a bottom-up, left-to-right fashion.

2.2 Movement

Phrase structure recovery may introduce underspecified empty categories. These empty categories may be instantiated as traces during the computation of movement. PAPPI directly implements overt NP movement, as in direct passives and subject-raising, and overt *wh*-movement, as in English *wh*-word fronting, by computing the possible chain configurations holding between overt NPs and empty categories. Subjacency, limiting the number of bounding nodes separating each link in a chain, is also implemented. For *wh*-in-situ languages like Japanese, overt *wh*-movement is deactivated via a `WhInSyntax` parameter. Covert movement of *wh*-phrases to specifier of CP scope positions will still occur to satisfy interface requirements.

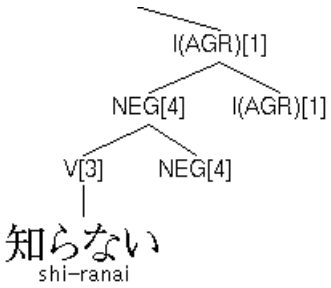
Scrambling imposes a heavy computational burden on parsing. PAPPI contains a special mechanism to handle cases of non-vacuous scrambling efficiently (Fong, 1994). In particular, argument NPs may be (leftwards) adjoined to VP forming an A-position, or IP, forming either an A or $\overline{\text{A}}$-position. An example of scrambling taken from (Saito, 1985) is given in (4):

(4) sono hon-o John-ga Mary-ga katta to omotteiru (koto)
that book John-NOM Mary-NOM bought COMP thinks
John thinks that Mary bought that book



Finally, head movement in the extended verbal projection is also implemented. Verbs may raise to inflection (I), through negation (neg), and on to complementizer (C/COMP) by compound head adjunction, e.g. for subject-auxiliary inversion in English and V2 in Germanic. In the Japanese implementation, verbs may raise through negation as far as inflection. An example of compound head adjunction is given in (5). The verbal head has adjoined to neg and inflection producing *shiranai* (know+neg+NPAST):

(5) shiranai
know+neg+NPAST



2.3 Case theory

We assume a basic configurational theory of Case assignment, where Cases are associated with particular positions. Sentential subjects receive nominative (NOM) Case from I(AGR) present in tensed clauses. Direct and indirect objects receive accusative (ACC) and dative/genitive (DAT/GEN) Case from verbs, respectively. NP-internal subjects and objects receive genitive (GEN) Case. Subjects of sentential complements may also receive Case exceptionally from higher verbs marked as Exceptional Case Markers (ECM). In Japanese, Case is realized on overt NPs by case particles such as *-ga* (NOM), *-o* (ACC), *-ni* (DAT) and *-no* (GEN), e.g. see (2), (4) (shown earlier) and (7) (below).[1] According to the Case Filter, all overt NPs must receive Case. Hence, for scrambling, we assume Case transmission operates. For empty NPs, traces of NP movement do not receive Case, whereas *wh*-traces must.[2] Japanese poses further problems for Case theory. For example, the Dative Subject Construction, to be discussed in section 3.6, apparently allows both subjects and objects to exhibit a variety of Case particles. The basic model will be revised in due course.

2.4 θ-theory

We assume lexical entries for main verbs contain θ-grids specifying the θ-roles to be assigned. Sample entries for *kau* (buy), used in (2), and *omou* (think), used in (4), are shown in (6).

(6)
```
lexicon(ka, v,[morph(kau,base(u)),  grid([agent],[theme]),eng(buy)]).
lexicon(omo,v,[morph(omou,base(u)), grid([agent],[proposition]),eng(think)]).
```

In both cases, the external role (`agent`) will be assigned to the subject. The internal roles (`theme`/`proposition`) will be assigned by the verb to the direct object. In (4), the θ-role `proposition` is assigned to the clausal complement headed by *to* (COMP). Finally, we assume the distribution of θ-roles obeys the θ-criterion; that is, argument chains and θ-roles are in one-to-one correspondence.

2.5 Other principles

PAPPI contains many other linguistic principles relevant to Government-and-Binding theory such as the Empty Category Principle (ECP) in which an empty category must either be governed by a lexical head, e.g. a trace in object position, or be governed by an antecedent, e.g subject or adjunct traces. We briefly mention two additional sub-systems used in the Japanese grammar.

[1]The current system does not address the issue of the topic marker *-wa*. There is no special topic phrase or distinguished Case position such as the specifier of CP.

[2]To accommodate scrambling, we define NP movement to exclude cases of A-movement to adjunct positions.

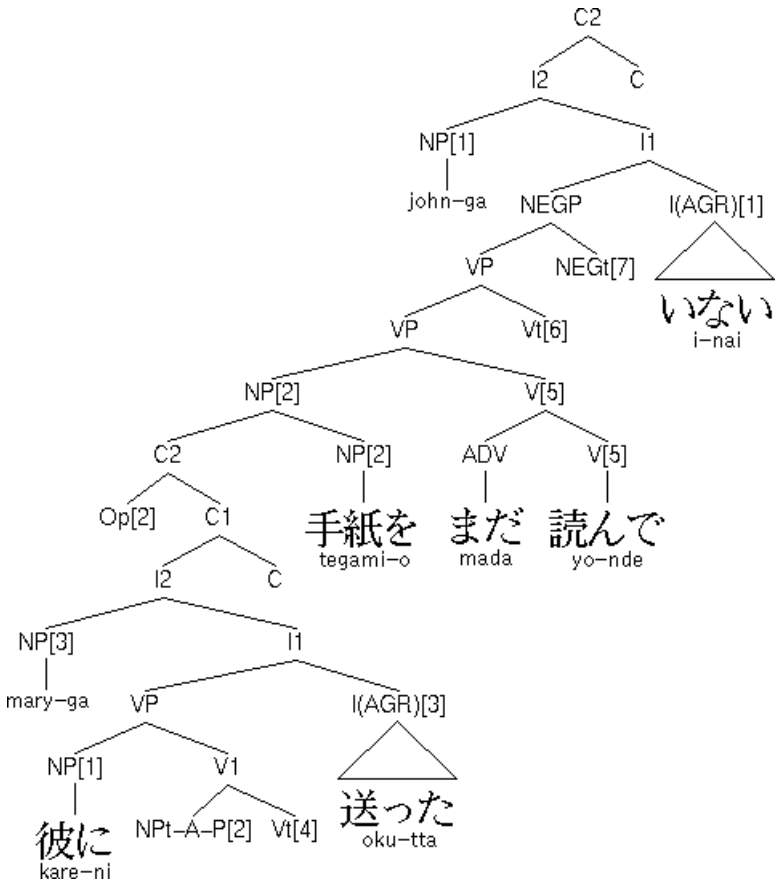
2.5.1 Binding Theory

We assume a simple set of Binding conditions operating over overt anaphors, e.g. *otagai* (each other) and *zibunzisin* (himself), and pronouns, e.g. *kare* (him) and *karera* (they), as well as empty NPs such as traces and pro (empty pronoun). Based on the notion of a Governing Category (GC) domain, anaphors and pronouns must be A-bound (Condition A) or not A-bound (Condition B) in their GC (provided one exists), respectively.[3] Referential expressions, e.g. proper nouns and variables, must remain A-free (Condition C). In (7), from (Saito, 1985), *kare* (him) is A-bound by (and coreferential with) the matrix subject *John*, which is outside GC(*kare*), namely, the relative clause.

(7)

John$_i$-ga	Mary-ga	kare$_i$-ni	okutta	tegami-o	mada	yonde inai
John$_i$-NOM	Mary-NOM	him$_i$-DAT	sent	letter-ACC	still	be reading+NEG

John has not yet read the letter Mary sent to him

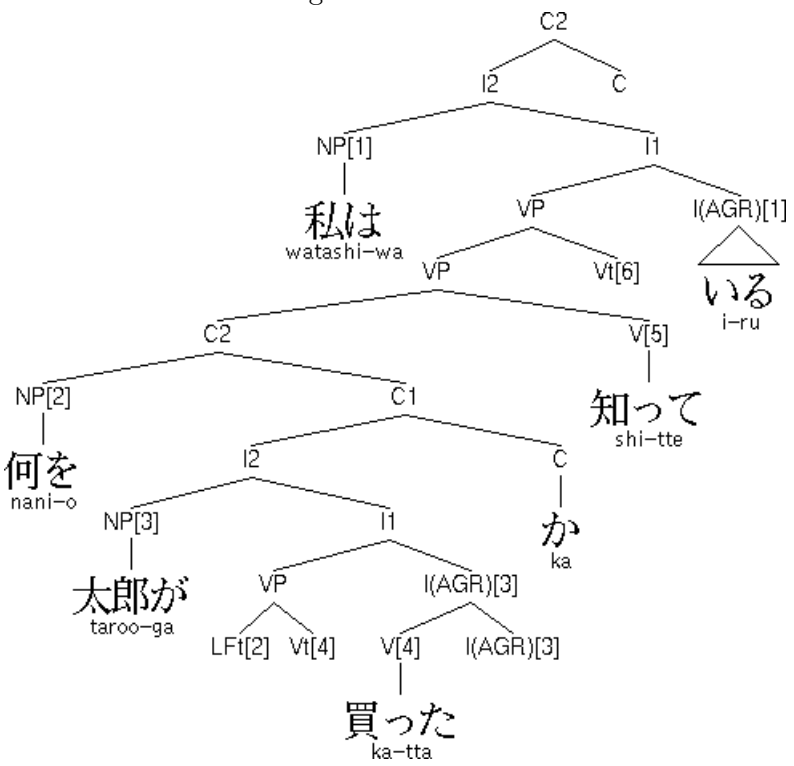


2.5.2 *Wh*-Comp Requirement

Following (Lasnik & Saito, 1984), although Japanese is a *wh*-in-situ language, we assume that *wh*-phrases must covertly raise to the specifier of CP at LF to satisfy the *wh*-Comp requirement. That is, if Comp is Q, realized as *-no* or *-ka* in Japanese and covert in English, the corresponding specifier position must be filled by a *wh*-phrase. This requirement holds in overt syntax and LF for English, and at LF for Japanese. PAPPI implements LF movement by transforming the trees formed by initial phrase structure recovery. In (8), from (Lasnik & Saito, 1984), *nani* (what) is raised at LF to the specifier of the embedded CP, mirroring its English counterpart.

[3] GC(α) is defined as the smallest domain having α, a governor and an accessible SUBJECT for α.

(8) Watashi-wa Taroo-ga nani-o katta ka shitte iru
I-TOP Taroo-NOM what-ACC bought Q know
I know what John bought



3 The Kanda Extensions

This section describe the extensions and changes to the PAPPI system, as configured in section 2, to account for a variety of Japanese phenomena including anti-superiority, indirect passives and potentials, *o*/*ni*-causative and dative subject constructions, and the double-*o* constraint.

3.1 Morphology

Japanese verbs exhibit complex morphology not present in the English system. Not only do verbs inflect for tense, but negation as well as argument and Case changing morphemes such as passives, causatives and potentials can be added. For Japanese, it was necessary to add a morphological decomposition stage PARSEPF prior to phrase structure recovery. For regular verbs such as *kau* (buy) and *shinu* (die), only the stem is explicitly listed in the lexicon, as shown in (9).

(9)
```
lexicon(ka, v,[morph(kau,base(u)),    grid([agent],[theme]),eng(buy)]).
lexicon(shi,v,[morph(shinu,base(nu)), grid([],[theme]),noCasemark(+),eng(die)]).
```

PAPPI defines both general and verb class-specific stemming rules. For example, the rules in (10) define `X+ru`, `X+ta` and `X+nai` to rewrite as `X` followed by the morphemes NPAST (non-past), PAST and NEG,NPAST (NEG+NPAST), respectively.

(10)
```
contraction(vEnd,X+ru,  [X=pf([require(vStem2)]),npast]]).
contraction(vEnd,X+ta,  [X=pf([require([vStem1,vStem2,vStem4])]),past]).
contraction(vEnd,X+nai, [X=pf([require([vNStem,vStem1,vStem2])]),neg,npast]).
```

Stemming rules may also be particular to verb classes. For example, the rules in (11) are specific to -*nu* verbs, thus allowing *shinu* (die+NPAST) and *shinda* (die+PAST) to be formed, plus two regular stems for passive and negation, e.g. *shinareta* (die+PASS+PAST), and *shinanai* (die+NEG+NPAST), that combine with the regular endings from (10).

(11)
```
contraction(vEnd,X+nu,    [X=word(base(nu)),npast]). % shinu: shi+NPAST
contraction(vEnd,X+nda,   [X=word(base(nu)),past]).  % shinda: shi+PAST
contraction(vStem2,X+nare,[X=word(base(nu)),pass]).  % shinare(ru): shi+PASS
contraction(vNStem,X+na,  [X=word(base(nu))]).       % shinanai: shi+na+NEG
```

It is important to note that morphemes generated by lexical decomposition have the option of projecting syntactic structure. For example, NEG as in *yonde inai* (be not reading) in (7) projects NegP. Similarly, PASS will project as an auxiliary verb, with the same syntactic properties as English passive *be*, for details see section 3.3. Morphemes such as NPAST/PAST will show up as tense features in inflection.

3.2 Anti-Superiority

A word order restriction exists for multiple *wh*-questions in Japanese. (12b) is the scrambled version of (12a).

(12) (a) ∗ Taroo-ga naze nani-o katta no
Taroo-NOM why what-ACC bought Q
Why did Taroo buy what?

(b) Taroo-ga nani-o naze katta no
Taroo-NOM what-ACC why bought Q

Given the basic constraints outlined in section 2, PAPPI treats both examples as grammatical and thus cannot distinguish between them. According to (Watanabe, 1992, p.266), there is an Anti-Superiority condition at work:

(13) The *wh*-phrase that is moved first cannot c-command the other *wh*-phrase at S-structure which takes the same scope.

Condition (13) can be implemented as follows:

1. (`multiWh`) Identify specifier of CP positions containing multiple *wh*-phrases.

   Multiple *wh*-phrases at the same location will show up as adjunction structures. The first movement to specifier of CP is by substitution, and by adjunction thereafter.

2. (`findTraceDomain`) Identify the trace of the first phrase to arrive.

3. (`dominateTraceOf`) Check to see it does not dominate the trace of the 2nd *wh*-phrase.

The code is given in (14).

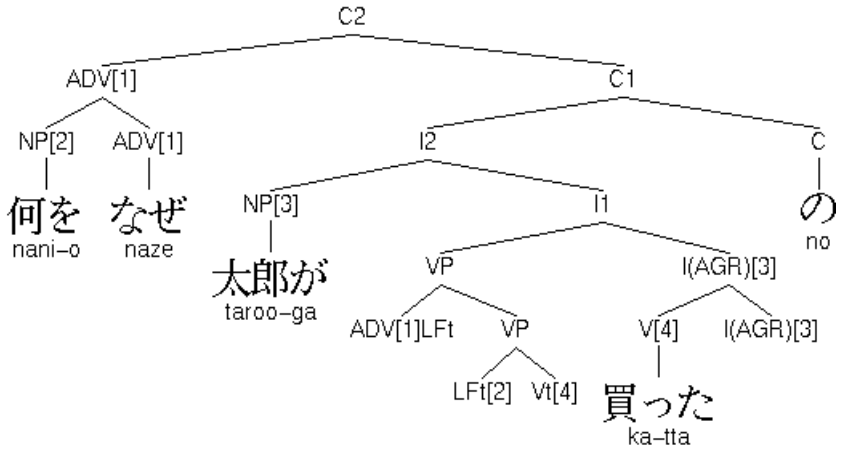
(14)
```
antiSuperiority in_all_configurations X where
        multiWh(X,Wh1,Wh2,IP) then \+[4] traceCC(Wh1,Wh2,IP).

multiWh(CP,Wh1,Wh2,IP) :-
        cat(CP,c2), Spec specifier_of CF,
```

[4] \+ is the negation operator in Prolog.

```
        adjoined(Spec,Wh2,Wh1), Wh1 has_feature moved(lf),
        IP complement_of CP.

traceCC(Wh1,Wh2,IP) :- findTraceDomain(Wh1,Dom), dominateTraceOf(Dom,Wh2).

findTraceDomain(Wh1,Dom) (not given) Dom is the c-command domain for Wh1-trace

dominatesTraceOf(X,Y) :- traceOf(X,Y).
dominatesTraceOf(X,Y) :- X has_constituent Z, dominatesTraceOf(Z,Y).
```

With the definition in (14), PAPPI correctly reports that (12a) is filtered out by Anti-Superiority.

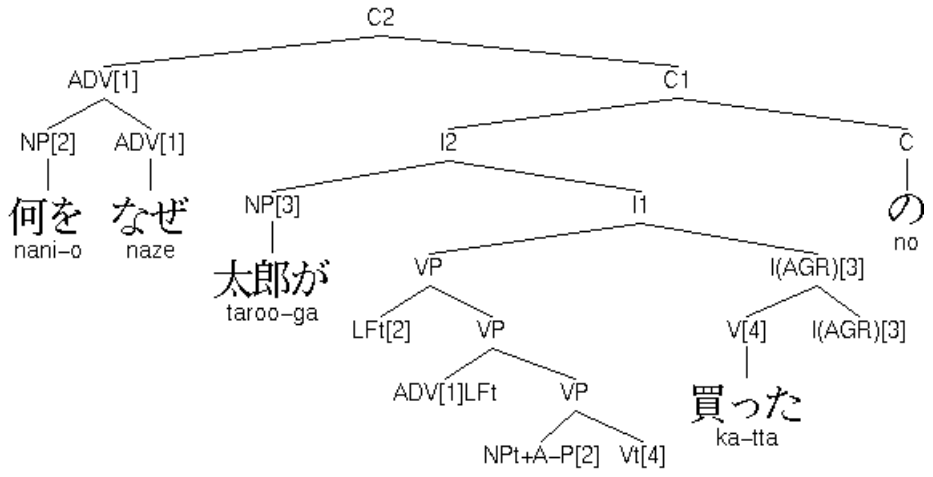
(15) Parsing: *Taroo–ga naze nani–o katta no
Enter Anti–Superiority: (1)



Parse blocked by Anti–Superiority
No parses found

The trace of *naze* (why), ADV[1]LF*t*, c-commands the trace of *nani* LF*t*[2] in (15). Compare with (16), where LF*t*[2] is higher than ADV[1]LF*t* by virtue of scrambling.

(16) Parsing: Taroo–ga nani–o naze katta no
LF (1):



One parse found

3.3 Direct and Indirect Passives

(17) (a) Hanako-ga (Taroo-ni) hihansareta
Hanako-NOM (Taroo-DAT) criticize+PASS+PAST
Hanako was criticized (by Taroo)

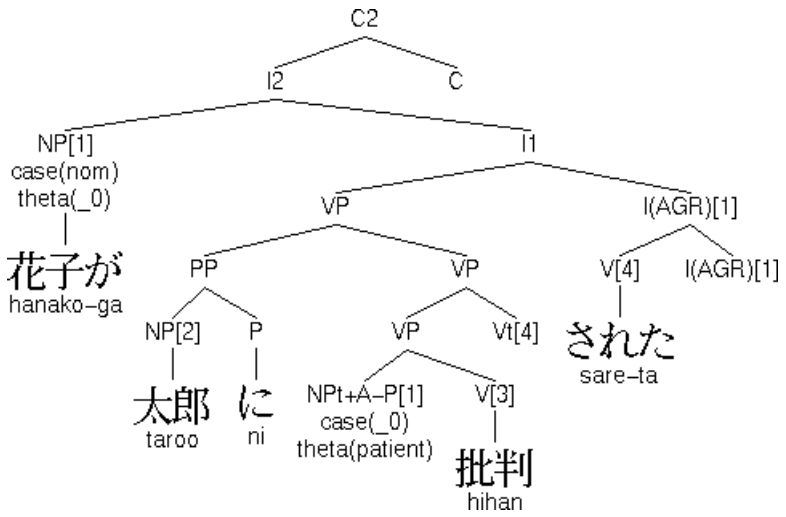
(b) Hanako-ga kodomo-ni shinareta
Hanako-NOM child-DAT die+PASS+PAST
As for Hanako, her child has died

Following section 3.1, lexical decomposition of a verb involving a passive morpheme, e.g. *hihansareta* (criticize+PASS+PAST), results in the creation of both a main verb and an auxiliary verb that behaves like the English passive verb *be*. The lexical entry for the passive is given in (18).

(18)
```
lex(pass,v,[morph(rareru,[]),aux,blockTheta,passive,
    subcat(vp$[morph(_,base(_)),grid([_],[_|_])],[noCasemark(+)])]).
```

(18) states that PASS is an auxiliary verb (`aux`) which subcategorizes (`subcat(vp$_,_)`) for a transitive (`grid([_],[_|_])`) main verb stem (`morph(_,base(_))`). Following classical analysis, it also removes the Case-marking ability of the base verb (`noCasemark(+)`), and prevents the external role from being expressed in subject position (`blockTheta`), thus forcing NP-movement of the object into subject position to obtain Case. (19) illustrates the process for (17a).

(19)



However, indirect passives must be handled differently. Thematically, the affected subject is not an intrinsic argument of the main verb. For example, in (17b), *shinu* (die) is an unaccusative verb taking a single argument *kodomo* (child). Here, we employ an ECM analysis. As shown in (20), indirect PASS supplies a subject experiencer role and takes a sentence as its complement.

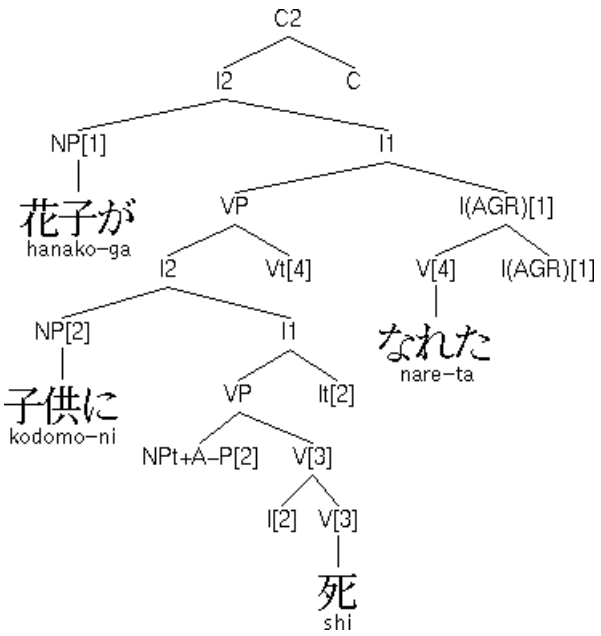
(20)
```
lex(pass,v,[morph(rareru,[]),grid([experiencer],[reducedProp]),
    ecm(oblig),doCase(dat)]).
```

It also assigns dative Case (`doCase(dat)`) via Exceptional Case Marking (`ecm(oblig)`) to the embedded sentential subject. The resulting parse for (17b) is shown in (21).[5]

[5]Two points to note:

1. In (21), the object of *shinu* (die) undergoes NP-movement to the embedded subject position for Case reasons. Its definition as an unaccusative was given previously in (9).

(21)



An outstanding problem remains. For (17a), PAPPI produces two additional indirect passive analyses, shown in (22).

(22) (a) $[_{IP}$ Hanako$_i$-ga $[_{VP}$ $[_{IP}$ Taroo-ni $[_{VP}$ pro$_j$ hihan]] PASS+PAST]]

(b) $*[_{IP}$ Hanako$_i$-ga $[_{VP}$ $[_{IP}$ Taroo-ni $[_{VP}$ pro$_i$ hihan]] PASS+PAST]]

In particular, (22b) is not blocked in the current implementation. Condition B from section 2.5.1 allows the embedded object *pro* to be A-bound by *Hanako*.[6]

3.4 *ni*- and *o*-Causatives

(23) (a) Hanako-ga Taroo-o utawaseta
Hanako-NOM Taroo-ACC sing+CAUS+PAST
Hanako made Taroo sing

(b) Hanako-ga Taroo-ni utawaseta
Hanako-NOM Taroo-DAT sing+CAUS+PAST
Hanako let Taroo sing

(c) Hanako-ga kodomo-o shinaseta
Hanako-NOM child-ACC die+CAUS+PAST
Hanako made the child die

(d) * Hanako-ga kodomo-ni shinaseta
Hanako-NOM child-DAT die+CAUS+PAST
Hanako let the child die

2. Also in (21), nonfinite INFL lowers to V in the embedded clause. For finite INFL, the verb raises instead. We tentatively assume, following (Pollock, 1989), that Japanese verbal inflection is strong, as is the case in Romance.

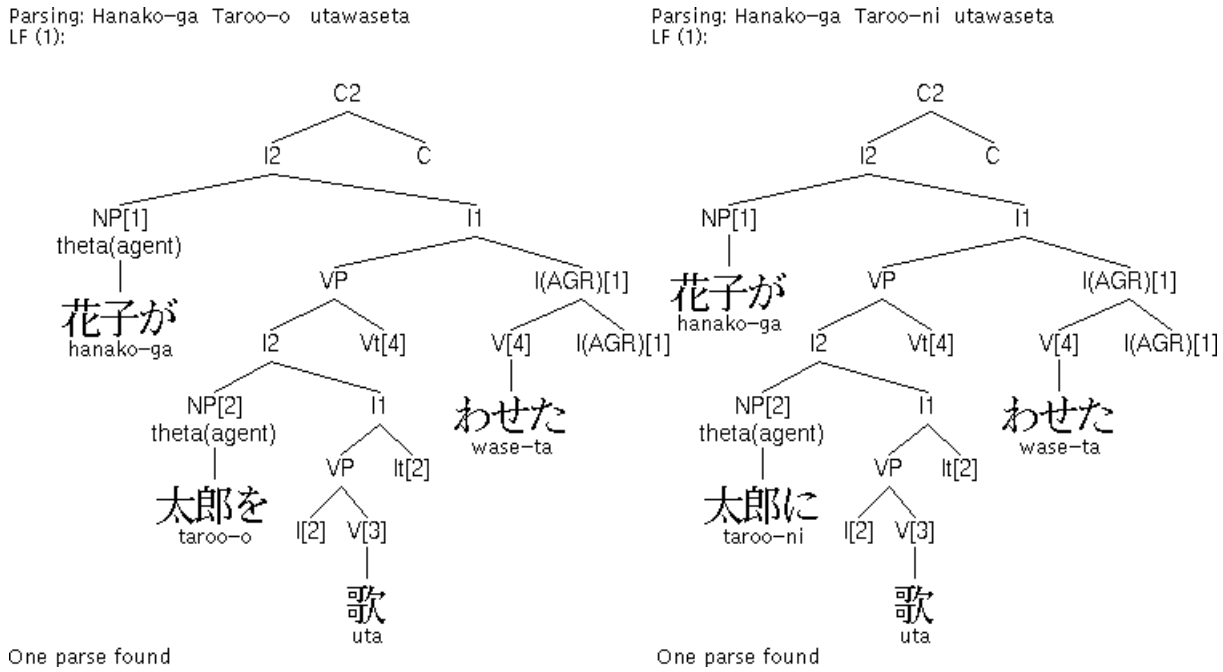
[6]In parallel with English ECM examples like *John believes Mary to like him*, the Binding domain is determined to be the embedded clause.

The Japanese periphrastic causative *sase*(*ru*) can be handled in a similar fashion to that for the indirect passive *rare*(*ru*) from section 3.3. CAUS, in (24), cf. PASS in (20), is defined to be an ECM verb.

(24)
```
lex(caus,v,[morph(saseru,[]),grid([agent],[reducedProp]), % make/let
            ecm(oblig),doCase(oneof(I,[acc,dat])),
            selR(goal(niCausConstraint(X),X))]).
```

Two modifications are necessary in order to handle the variation in embedded Case, as exemplified in (23a) and (23b), and to block *ni*-causative in cases where the embedded subject is non-agentive, as in (23d). The feature `doCase(oneof(I,[acc,dat]))` allows CAUS to exceptionally Case mark the embedded subject as either accusative or dative. The parses for (23a) and (23b) are given in (25).

(25)

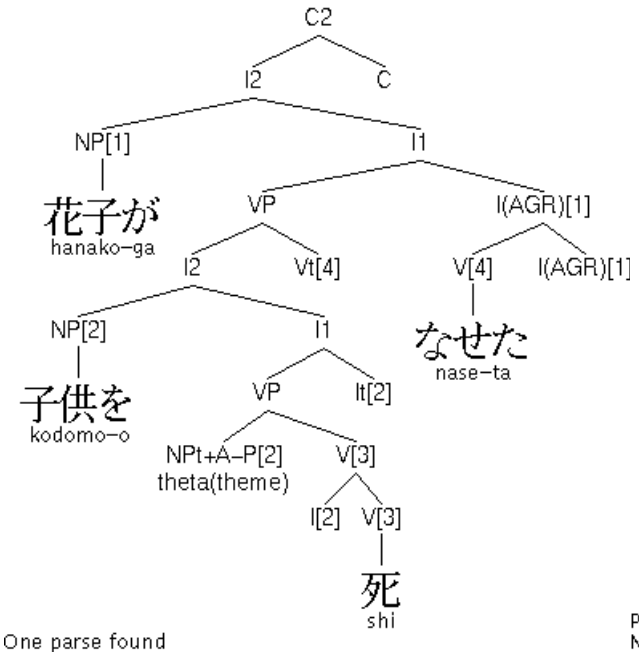


The selectional restriction (`selR(_)`) feature in (24) causes the goal `niCausConstraint(X)`, defined in (26), to be applied to the sentential complement `X`.

(26)
```
niCausConstraint(IP) :-
            Subj specifier_of IP,
            addFeature(noHeadChain,Subj) if niMarked(Subj).
```

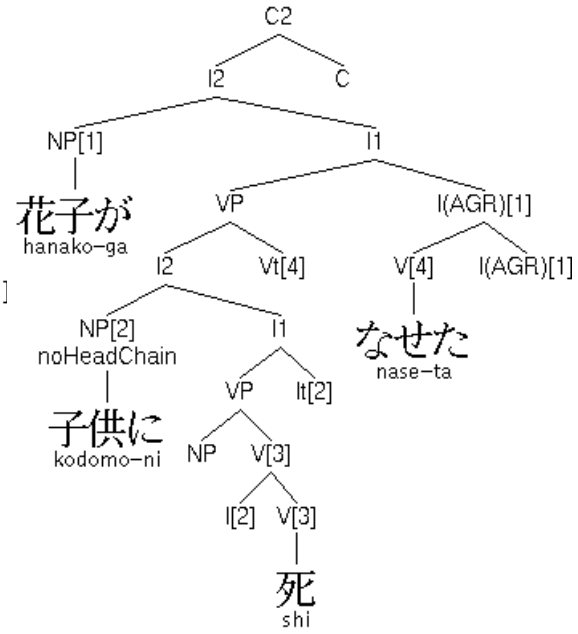
Because θ-roles are assigned to D-structure positions, the parser is unable to check for agentivity merely by inspecting the embedded subject. The following work-around is used instead: `niCausConstraint` picks out the embedded subject (`Subj`) and checks to see if it is *ni*-marked. If so, it adds a feature `noHeadChain` which explicitly prevents chain formation from using it as the head of a (non-trivial) chain. Since agent θ-roles are directly assigned to sentential subjects, no movement being necessary, this effectively blocks verbs like *shinu* (die), which assign an object theme θ-role, from forming the unaccusative chain it needs to be properly licensed. (27) shows both the completed parse for (23c) and the `noHeadChain`-augmented structure for (23d) that is blocked by the θ-criterion.

(27) Parsing: Hanako-ga kodomo-o shinaseta
LF (1):



One parse found

Parsing: Hanako-ga kodomo-ni shinaseta
Exit Trace Theory: (2)



Parse blocked by Theta Criterion
No parses found

Finally, the causative implementation also has important implications for PAPPI's scrambling mechanism. (28b) and (28c) below illustrate two instances of object scrambling in a causative construction.

(28) (a) Hanako-ga Taroo-ni hon-o yomaseta
Hanako-NOM Taroo-DAT book-ACC read+CAUS+PAST
Hanako let Taroo read the book

(b) Hanako-ga hon-o Taroo-ni yomaseta
Hanako-NOM book-ACC Taroo-DAT read+CAUS+PAST

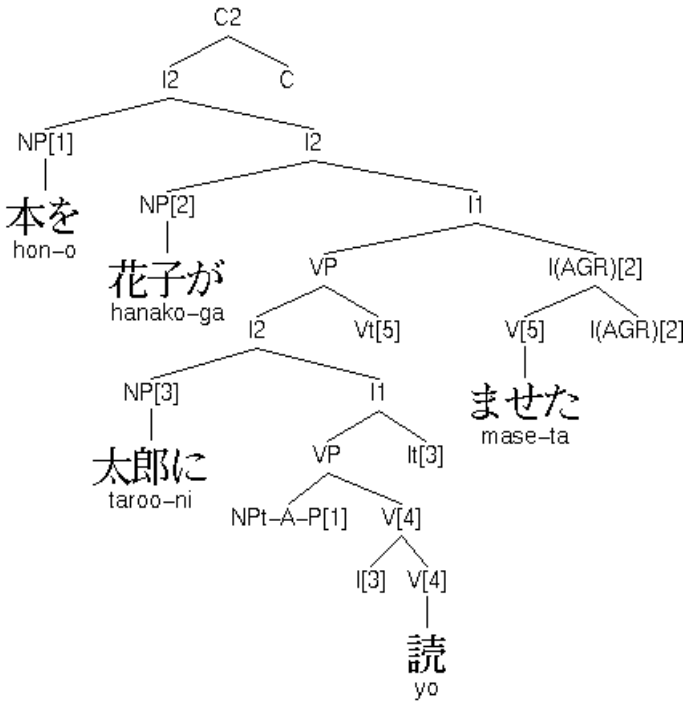
(c) hon-o Hanako-ga Taroo-ni yomaseta
book-ACC Hanako-NOM Taroo-DAT read+CAUS+PAST

Since CAUS subcategorizes for an IP, (28b) and (28c) count as cases of medium and long-distance scrambling, respectively. In both cases, the object *hon-o* (book-ACC) is scrambled above the embedded subject *Taroo*. By default, PAPPI assumes, unless explicitly mentioned, that scrambled positions are A-positions. Hence, we need to define (code not shown) the configuration in (29) as being an instance of long-distance scrambling. That is, the landing site for NP_i should be marked as an $\overline{\text{A}}$-position to satisfy Binding theory.

(29) $[_{\text{IP}}\ \text{NP}_i\ [_{\text{IP}}\ [_{\text{VP}}\ [_{\text{IP}}\ \ldots t_i \ldots\ ]\ \text{V}]]]$

The parse for (28c) is given in (30).

(30) Parsing: [9b] hon–o Hanako–ga Taroo–ni yomaseta
LF (1):



One parse found

Note that the trace of *hon-o* (book-ACC), namely NP*t*-A-P[1], is marked as a variable and thus satisfies Condition C of the Binding theory.

3.5 Double-*o* Constraint

Japanese does not permit two *o*-marked NPs in the same clause. For example, the *o*-causative construction shown in (31a) is ungrammatical.

(31) (a) ∗ Hanako-ga Taroo-o hon-o yomaseta
Hanako-NOM Taroo-ACC book-ACC read+CAUS+PAST
Hanako made Taroo read the book

(b) ∗ hon-o Hanako-ga Taroo-o yomaseta
book-ACC Hanako-NOM Taroo-o read+CAUS+PAST

However, a naïve implementation that simply scans the surface string for two consecutive *o*-marked NPs is insufficient for several reasons. First, one of the NPs may be scrambled, as in (31b). Scanning the entire string is not a viable solution either. As (32) illustrates, two (or more) *o*-marked NPs are acceptable if they are separated by a clausal boundary.

(32) Taroo-ga tegami-o kaita kodomo-o hihanshita
Taroo-NOM letter-ACC wrote child-ACC criticized
Taroo criticized the child who wrote the letter

Finally, the double-*o* constraint also operates in the presence of empty categories. For example, (33) is ungrammatical because the covert object of *yomaseta* (read+CAUS+PAST) is also marked with accusative Case.

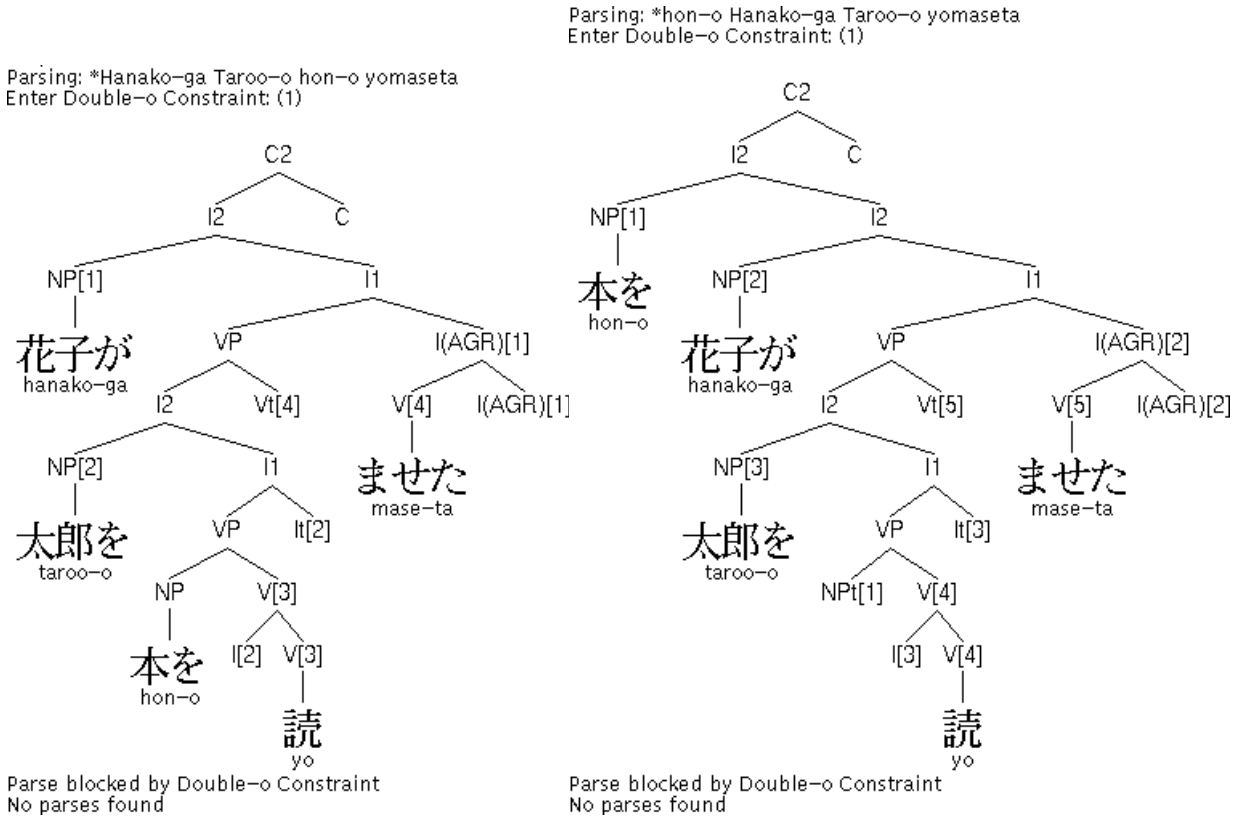
(33)∗ John-ga Mary-o yomaseta
John-NOM Mary-ACC read+CAUS+PAST
John made Mary read (something)

The constraint implemented by (34) is activated by an instance of a double-*o* configuration (`doubleOConfig`), in which an *o*-marked NP occupies either a subject or IP adjunct position. Then, `oMarkedIn` simply searches the tree for another *o*-marked NP, stopping only at clause boundaries.

(34)
```
doubleO in_all_configurations X where
        doubleOConfig(X,Dom) then \+ oMarkedIn(Dom).

doubleOConfig(IP,VP) :- % [IP NP-o [VP .. ]] (o-marked subject)
        cat(IP,i2),
        Subj specifier_of IP, oMarked(Subj),
        VP complement_of IP.

doubleOConfig(IP,IP2) :- % [IP NP-o [IP .. ]] (scrambled direct object)
        cat(IP,i2),
        adjoined(IP,NP,IP2), oMarked(NP). % [IP NP IP2]

oMarkedIn(X) :- simpleNP(X), oMarked(X).
oMarkedIn(X) :- \+ cat(X,c2), X has_constituent Y, oMarkedIn(Y).

oMarked(X) :-  cat(X,np), X has_feature case(Case), Case == acc.
```

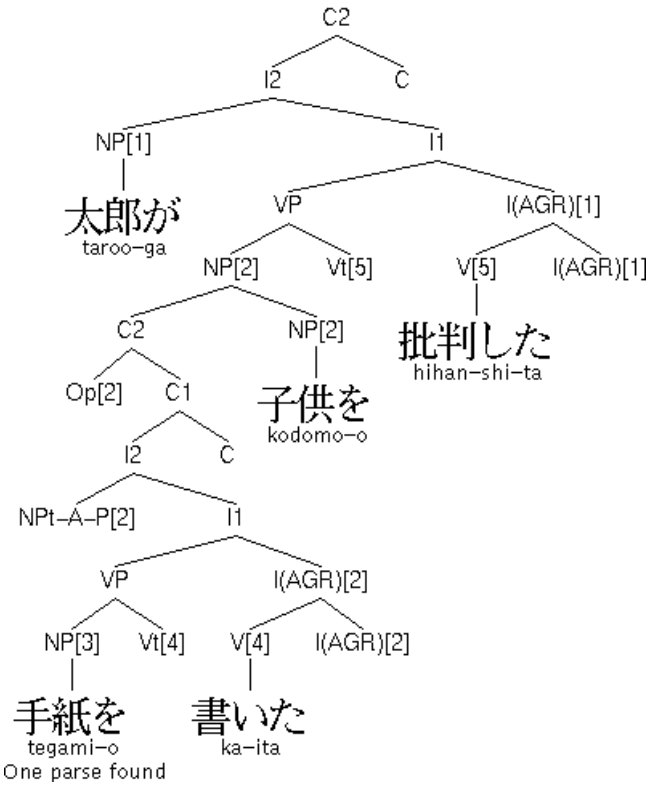
As (35) illustrates, the double-*o* constraint correctly blocks both (31a) and (31b).

(35)



Furthermore, the implemented condition correctly permits (32).

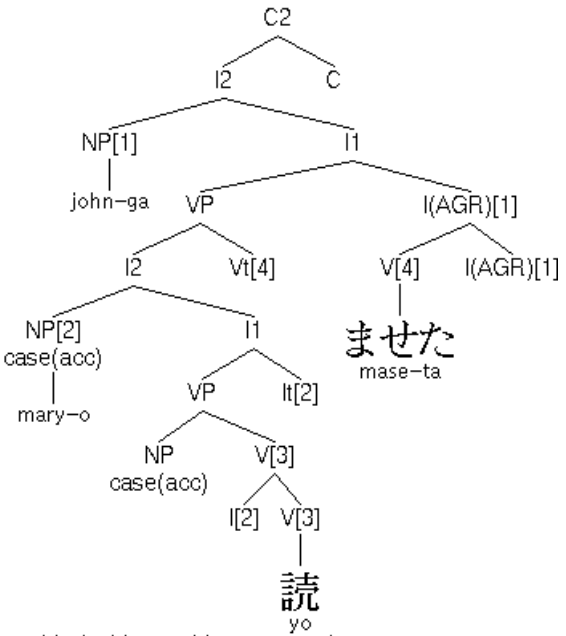
(36) Parsing: Taroo-ga tegami-o kaita kodomo-o hihanshita
LF (1):



One parse found

Finally, (33) is ruled out as accusative Case has been assigned to both *Mary* and *pro*.

(37) Parsing: *John-ga Mary-o yomaseta
Enter Double-o Constraint: (1)



Parse blocked by Double-o Constraint
No parses found

3.6 Dative Subject Construction

The Dative Subject Construction (DSC) may occur with a variety of predicates. For example, as (38) illustrates, it may occur with psych-predicates, stative verbs and verbs modified by the

potential suffix.

(38) (a) Taroo-ni hebi-ga kowai
Taroo-DAT snake-NOM fearful+NPAST
Taroo is fearful of snakes

(b) Taroo-ni eigo-ga wakaru
Taroo-DAT English-NOM understand+NPAST
Taroo understands English

(c) Taroo-ni eigo-ga hanaseru
Taroo-DAT English-NOM speak+POT+NPAST
Taroo can speak English

In the examples in (38), the experiencer subject is marked with dative Case, and the theme argument receives nominative Case. As (39a) and (39b) illustrate, a nominative subject is also possible. However, in this configuration the theme argument must either be nominative or accusative-marked. Note that the object cannot be marked with accusative Case in the presence of a dative subject, as shown in (39c).

(39) (a) Taroo-ga eigo-o wakaru
Taroo-NOM English-ACC understand+NPAST
Taroo understands English

(b) Taroo-ga eigo-ga wakaru
Taroo-NOM English-NOM understand+NPAST
Taroo understands English

(c) * Taroo-ni eigo-o wakaru
Taroo-DAT English-ACC understand+NPAST
Taroo understands English

The DSC poses a double challenge for the PAPPI system. First, the simple model of configurational Case assignment from section 2.3 can no longer be maintained. Apparently, subjects may receive dative Case, and objects nominative Case.[7] Second, for computational efficiency the scrambling mechanism introduced in section 2.2 relies on the inspection of Case particles to determine whether an argument has been displaced. With the introduction of the DSC, subjects and objects can no longer be reliably distinguished until the parser has encountered the predicate. We address the problem of Case assignment here, postponing a discussion of parser efficiency to the final section.

For the stative verb *wakaru* (understand), we assume it consists of a bound form *waka* followed by an abstract verb STATIVE and tense, i.e. *waka*+V[STATIVE]+NPAST. The lexical entries for *waka* and STATIVE are given in (40) and (41), respectively.

(40)
```
lexicon(waka,v,[morph(wakaru,base(ru(1))),
                grid([],[theme]),stative,eng(understand)]).
```

(41)
```
lex(stative,v,[aux,morph(stative,[]),grid([experiencer],subcat),
          subcat(vp$[stative],[doCase(oneof(I,[nom,accNom]))]),
          adjR([addFeature(subjCase(oneof(I,[dat,nom])))])]).
```

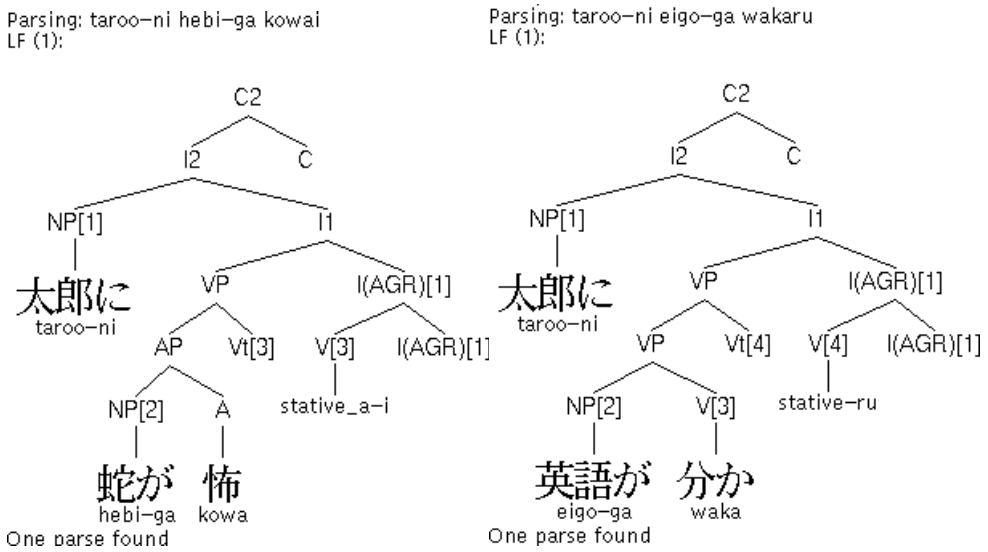
The experiencer subject role will be supplied by STATIVE, with the bound form *waka* supplying a theme argument only. STATIVE is defined as an auxiliary (`aux`) verb that subcategorizes (`subcat(vp$[stative],_)`) for any bound form with feature `stative`.[8] Moreover, the

[7] According to (Ura, 1999), the dative experiencer (but not the nominative theme) exhibits signs of subjecthood with respect to binding, control and subject-honorification.

[8] In the case of the adjective *kowai* (fearful), we assume that STATIVE has a counterpart STATIVE_A that subcategorizes for adjectival phrases.

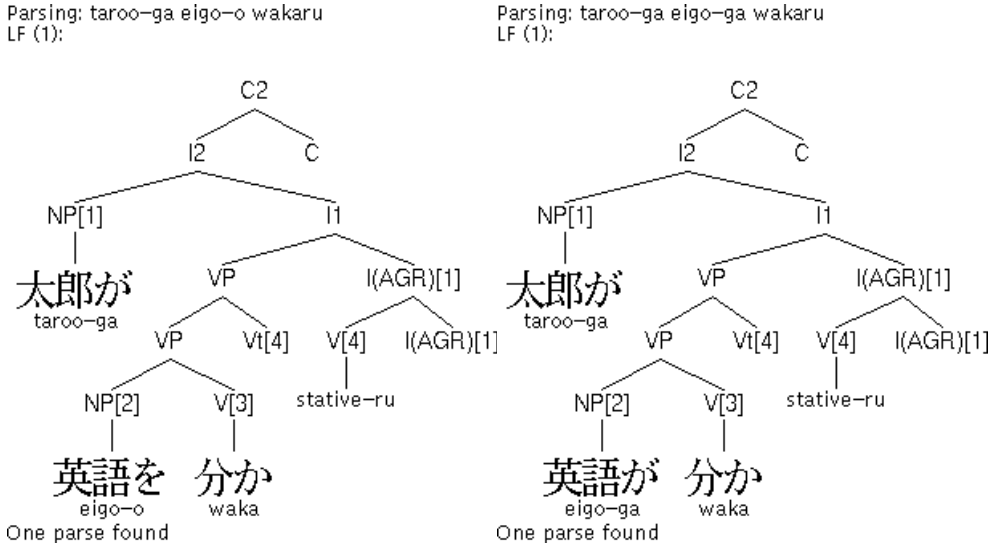
PAPPI `oneof(I,List)` construct is used to enforce Case parallelism for direct object and subject Cases via the features `doCase(_)` and `subjCase(_)`, respectively.[9] STATIVE assigns the feature `doCase(oneof(I,[nom,accNom]))` to the bound form it subcategories for. Additionally, STATIVE bears tense features and adjoins to inflection via head movement. By adjoining to inflection, the feature `subjCase(oneof(I,[dat,nom]))` will be transferred for assignment to the specifier of IP. The resulting parses for DSC sentences (38a) and (38b) are shown in (42).

(42)



(43) illustrates the Case assignment possibilities for the theme argument when the subject is assigned nominative Case in (39a) and (39b).

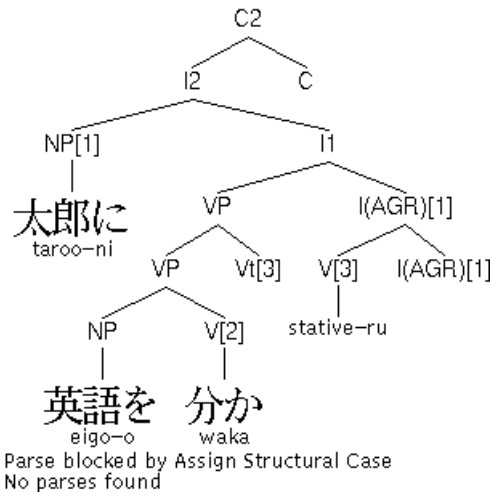
(43)



Furthermore, (39c) is blocked during Case assignment:

[9]The `oneof(I,List)` construct works as follows: `I` represents the ith member of `List`. Hence, `I`=1 when `dat` in `subjCase(one(I,[dat,nom]))` is selected. Since `I` is shared, this forces `nom` to be selected in `doCase(oneof(I,[nom,accNom]))`.

(44) Parsing: taroo–ni eigo–o wakaru
Enter Assign Structural Case: (1)



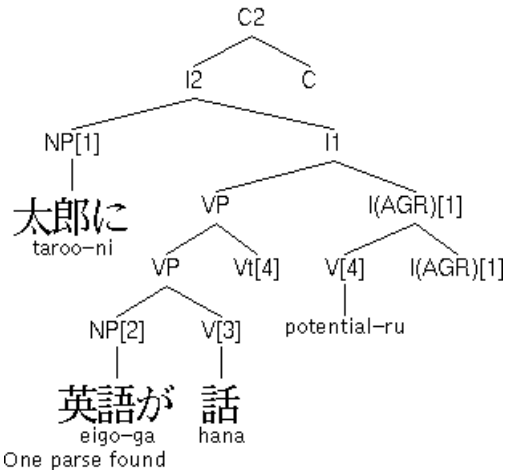
Parse blocked by Assign Structural Case
No parses found

Finally, in (38c) we assume *hanaseru* (able to speak) is composed of a main verb stem *hana* (speak) followed by an abstract morpheme POTENTIAL and tense. The lexical entry for POTENTIAL shown in (45) is identical to that for STATIVE in (41), except that POTENTIAL does not subcategorize for stative stems (`subcat(vp$[not(stative)],_)`).

(45)
```
lex(potential,v,[aux,morph(potential,[]),grid([experiencer],subcat),
          subcat(vp$[not(stative)],[doCase(oneof(I,[nom,accNom]))]),
          adjR([addFeature(subjCase(oneof(I,[dat,nom])))])]).
```

The corresponding parse for (38c) is given in (46).

(46) Parsing: taroo–ni eigo–ga hanaseru
LF (1):



One parse found

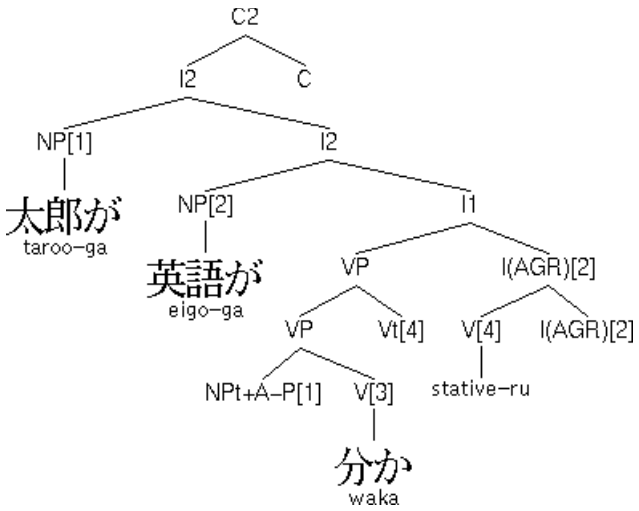
4 Conclusions

To summarize, the Kanda Extensions described in section 3 fall into two categories. Some extensions, e.g. indirect passives and causatives, exploit or re-use existing mechanisms or principles, e.g. exceptional Case marking, and only require the positing of new lexical entries. Others, such as the case of anti-superiority and the double-*o* constraint, involve the definition of new structural constraints. Given the highly modular nature of grammar, we have seen that there is

also considerable interaction between the various extensions, for instance, between the double-*o* constraint, scrambling and causatives. Nevertheless, the modifications to the system have been relatively small. These extensions represent a monotonic improvement in coverage without either a wholesale or radical reformulation of phrase structure or existing principles.

On the other hand, the DSC effectively prevents the phrase structure recovery module from operating efficiently. As mentioned earlier, since both subjects and objects can be *ga*-marked, PAPPI is unable to use Case particles to determine which elements have been scrambled. Due to this structural ambiguity, phrase structure recovery will come up with two parses for a sentence like (39b). See (43) (2nd parse) and (47) (below).

(47)



(39b) can be limited to a single parse by imposing an animacy requirement on experiencer role assignment, as shown in (48). Since *eigo* (English) is inanimate, (47) can be eliminated.

(48) `csrRestriction(experiencer,goal(animate(X),X)).`

However, aside from issues of incompleteness, the broader computational problem is this: given the strict left-to-right, bottom-up strategy employed by phrase structure recovery, the parser is forced to engage in decision-making on building structure and phrasal attachment before the main predicate has been encountered. For a head-final language like Japanese, the overhead from clause-internal garden-pathing can be especially severe.[10] A possible solution might be to logically postpone such decisions. For instance, an underspecified (flat) representation could be constructed first, with hierarchical positioning to follow once Case and θ-properties of the verbal head are known. A strategy of lazy incremental parsing, or computational procrasination, should go a long way towards eliminating unnecessary computational choice points without impacting parser competence.

References

CHOMSKY N. (1981). *Lectures on Government and Binding.* Foris Publications.

COLMERAUER A., KANOUI H., PASERO R. & ROUSSEL P. (1973). *Une Système de Communication Homme-Machine en Français.* Report, artificial intelligence group, Université d'Aix-Marseille II.

FONG S. (1991). *Computational Properties of Principle-Based Grammatical Theories.* PhD thesis, Artificial Intelligence Laboratory, MIT.

[10] Currently, the parser can spend a unusually high proportion of its time inside the phrase structure recovery module. For some examples, especially involving DSCs, this accounts for well over 99% of the compute time.

Fong S. (1994). Towards a proper linguistic and computational treatment of scrambling: An analysis of Japanese. In *COLING '94*, Kyoto, Japan.

Knuth D. E. (1965). On the translation of languages from left to right. *Information and Control*, **8**(6), 607–639.

Lasnik H. & Saito M. (1984). On the nature of proper government. *Linguistic Inquiry*, **15**(2).

Lasnik H. & Uriagereka J. (1988). *A Course in GB Syntax: Lectures on Binding and Empty Categories*. MIT Press.

Pollock J.-Y. (1989). Verb movement, universal grammar, and the structure of ip. *Linguistic Inquiry*, **20**(3).

Saito M. (1985). *Some Asymmetries in Japanese and Their Theoretical Implications*. PhD thesis, MIT.

Ura H. (1999). Checking theory and dative subject constructions in Japanese. *Journal of East Asian Linguistics*, **8**, 223–254.

Watanabe A. (1992). Subjacency and s-structure movement of Wh-in-situ. *Journal of East Asian Linguistics*, **1**, 255–291.