22

A Linear Approach to Multiple Clause Embedding

Kei Yoshimoto

22.1 Introduction

It is generally accepted among psycholinguists that real-time human sentence processing proceeds incrementally from left to right (see for example Mazuka & Itoh 1995). Recently proposals have been made in the domain of syntax to reduce phenomena which have hitherto been accounted for in terms of linguistic performance to linear structures given at the level of competence (for example Babyonyshev & Gibson 1999, Joshi 1990, Rambow & Joshi 1994, and Lewis & Nakayama 2001). Keeping in line with this tendency in research, this paper tries to reestablish the much discussed relationship between the two aspects of language, competence and performance: the issue of processing difficulty dependent on sorts of multiple clause embedding is addressed by incorporating into HPSG a mechanism reflecting *left-to-right* processing and *memory costs* calculated at each processing step.

The organization of this paper is as follows. After delineating processing difficulty caused by multiply embedded clauses in Section 22.2, a short introduction to the psycholinguistic theory we rely on, the Syntactic Prediction Locality Theory, is provided in Section 22.3. Section 22.4 proposes an extension of the linearization-based version of HPSG to equip it with an architecture which evaluates sentence complexity. Then Section 22.5 illustrates how the mechanism copes with the difference in processing complexity between differently embedded rela-

The Proceedings of the 9th International Conference on HPSG.
Jong-bok Kim and Stephen Wechsler.

tive clauses in Japanese. Section 22.6 proposes an application of our approach to a yet unknown relationship between memory load and prosody. The last section summarizes the discussion and mentions a possible use of the proposed theory as a uniform framework to process diverse understudied linguistic phenomena.

22.2 Types of Relative Clause Embedding and Processing Difficulty

It is well established that understanding of multiply embedded clauses is affected by how they are embedded. Sentence (1a)—an example from a right-branching language, English—in which embedded clauses each appear to the right of their heads is much easier to understand than (1b) in which center embedding or mixture of right-branching and left-branching doubly occurs (Chomsky & Miller 1963):

(1) a. Mary saw the friend [who recommended the real estate agent [who found the great apartment]].

b. *The rat [the cat [the dog chased] ate] died.

In Japanese, a typical left-branching and head-final language, a sentence with left-branching relative clauses, as in (2a), causes no difficulty, while a center-embedded sentence (2b) is harder to understand.[1]

(2) a. [$_{\text{S}}$ [$_{\text{S}}$ Rinjin ga kodomo ni kure-ta] ringo wo
neighbor SBJ child OBJ2 give-PAST apple OBJ

kajit-ta] nezumi wo neko ga oikake-ta.
gnaw-PAST rat OBJ cat SBJ chase-PAST

'The cat chased the rat which gnawed the apple the neighbor gave to the child.'

b. ??Kōchō ga [$_{\text{S}}$ sensei ga [$_{\text{S}}$ nezumi ga kajit-ta]
principal SBJ teacher SBJ rat SBJ gnaw-PAST

ringo wo kure-ta] kodomo wo shikat-ta.
apple OBJ give-PAST child OBJ scold-PAST

'The principal scolded the child to whom the teacher gave an apple a rat gnawed.'

Phrase Structure Grammar (PSG) and constraint-based grammars with representations reflecting PSG assign recursive structures to both types of embedding, disregarding the difference between them. For this reason, and also because of graded distinctions in comprehensibility, the

[1] See Mazuka & Itoh (1995) for the result of an experiment which shows an increase in reading time when subjects were given a sentence with this kind of syntactic structure.

prevailing view has been that the types of embedding must be captured in terms of performance rather than competence.

In recent years, accounts have been proposed on issues such as multiple clause embedding and word order based on left-to-right processing of sentences. Gibson and Babyonyshev, advocating the Syntactic Prediction Locality Theory, attempt to rate the on-line processing complexity of a variety of nested constructions in English and Japanese (see e.g. Gibson 1998 and Babyonyshev & Gibson 1999). (Bottom-up) Embedded Pushdown Automaton by Joshi (1990) and Rambow & Joshi (1994) copes with Dutch and German word orders from the point of view of limitations within a left-to-right processing model.[2] Lewis & Nakayama (2001) sets up a hypothesis that interference based on syntactic and positional similarity crucially affects human sentence processing, specifically that of center embedding. Furthermore, Kempson et al. (2001) establishes a basis of a logico-semantic approach to various syntactic difficulties by incrementally building up semantic representations as sentences are processed from left to right.

The proposed study copes with the processing difficulty involving multiple embedding, which exceeds the limitations of standard HPSG, by adopting Gibson and Babyonyshev's rating of left-to-right processing complexity.

22.3 Syntactic Prediction Locality Theory

Gibson and Babyonyshev (specifically, Gibson 1998) try to quantify the sentence complexity involving multiple embedding by the memory load of syntactically predicted categories. Their theory called 'the Syntactic Prediction Locality Theory (SPLT)' is based on two notions of processing cost. *Memory cost* is calculated in terms of how many syntactic categories are required to complete the input constituent as a grammatical sentence. *Integration cost* involves computational resources that are necessary to integrate the new input string to the currently existing syntactic structure. The resources are proportional to the distance between the two constituents.

In this paper, memory cost is adopted as the only criterion to measure sentence complexity, following Babyonyshev & Gibson (1999). This is because, first, memory cost alone covers all the phenomena discussed in this paper. Furthermore, this is 'effectively potential integration cost' (Gibson 1998) and the proposed constraints can be extended later to account for integration cost too.

[2] Hawkins (1994) puts forward a similar, but less formal theory on word-order universals from the point of view of processing efficiency.

The SPLT forms the basis on which the proposal of this paper develops, since it can account for the complexity of an abundant variety of sentences in a manner verifiable by psycholinguistic experiments. Another advantage of the theory is that it is neutral in terms of syntactic assumptions, and accordingly easily applicable to HPSG. Thus the proposed framework is essentially a syntactic one. In fact, it has been pointed out that the processing difficulty is a complex issue involving lexical, semantic, pragmatic, and discoursal conditions. I assume that constraints from the other components of grammar are imposed additionally on the syntax.

The proposed study has been constrained by the present state of psycholinguistic research. Although Lewis & Nakayama (2001) is important in that it explains data not accountable by Gibson and Babyonyshev, it is not adopted in this paper. This is because, first, given that data discussed by Lewis and Nakayama and those dealt with by Gibson and Babyonyshev cannot be accounted for by each other's theory, it is extremely difficult at this stage of research to decide which side is right.[3] Second, with many syntactic details remaining unclear, it is too early to apply the HPSG formalization to Lewis and Nakayama's hypothesis. If future studies may reveal that this line of research has essential importance, the advanced theory will be revised by giving additional constraints to the DOM list proposed in the next section.

22.4 HPSG Formalization

The grammar I propose is an extension of the linearization-based version of HPSG (Reape 1994 and Kathol 2000) in which the DOM feature is used as a record of memory costs to represent the processing complexity involving the prediction and satisfaction of syntactic categories. An additional feature S(YNTACTIC-)P(REDICTION-)L(OCALITY)-INF(ORMATION) within *spl*(-)*u*(*nit*), a type constituting the DOM list corresponding to Kathol's (2000) *dom-obj*, stands for this information. The feature's value is a feature structure specified for attributes LOC(AL)-VAL(UE), STACK, PREV(IOUS)-STACK, and BASE-STACK, all with values of type *list(mem-cost)*. STACK is the place where the information on the *splu*'s memory cost is stored. This is obtained based on the values of LOC-VAL and BASE-STACK. As a value of LOC-VAL a memory cost is first brought into existence and is then propagated to STACK. The BASE-STACK feature in turn is built up

[3] The only study I know which compares the two theories based on common data is Kruijff & Vasishth (2001). They discuss that both can account for subsets of their data.

from relevant principles and the value of PREV-STACK representing the STACK information of the immediately preceding *splu*.

(3)
$$
\textit{sign} \longrightarrow
\left[\begin{array}{ll}
\text{PHON} & \textit{phon} \\
\text{SYNSEM} & \textit{synsem} \\
\text{DOM} & \textit{list}\left(\left[\begin{array}{ll}
\multicolumn{2}{l}{\textit{splu}} \\
\text{PHON} & \textit{phon} \\
\text{SYNSEM} & \textit{synsem} \\
\text{SPL-INF} & \left[\begin{array}{ll}
\text{LOC-VAL} & \textit{list(mem-cost)} \\
\text{STACK} & \textit{list(mem-cost)} \\
\text{PREV-STACK} & \textit{list(mem-cost)} \\
\text{BASE-STACK} & \textit{list(mem-cost)}
\end{array}\right]
\end{array}\right]\right)
\end{array}\right]
$$

The type *mem-cost* has features PHON, HEAD, and a new feature DISTANCE with a value of type *number* representing the number of intervening sytactic categories processed until a predicted category is met.

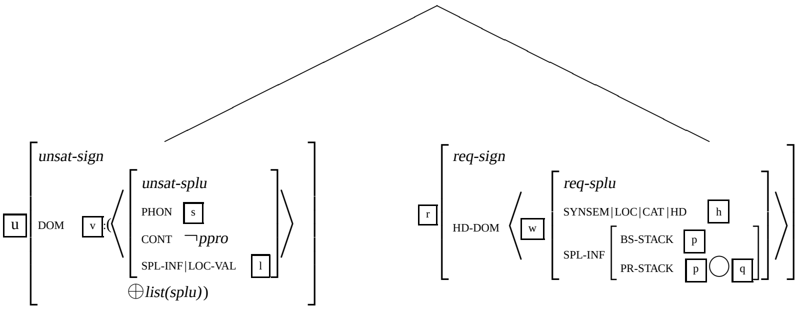
(4) a.
$$
\textit{mem-cost} \longrightarrow
\left[\begin{array}{ll}
\text{PHON} & \textit{phon} \\
\text{HEAD} & \textit{head} \\
\text{DISTANCE} & \textit{num}
\end{array}\right]
$$

b. $\textit{head}(\textit{phon}, \textit{num})$

(4a) is abbreviated as (4b) hereafter.

A *mem-cost* is introduced as a member in the vaule of LOC-VAL, percolated to the value of STACK, and later eliminated from the latter value when the predicted syntactic category is processed. The introduction and elimination is specified by the Memory Cost Principle (MCP) common to Japanese and English. In the following, $\oplus$ and $\bigcirc$ stand for list appending and the 'shuffle' relation.

(5) Memory Cost Principle



Conditions:

(i) $\boxed{\text{l}}$ is the smallest list of *mem-cost*s such that

$$\boxed{\text{l}} = (\boxed{\text{m}}\text{:}\mathit{list(mem\text{-}cost)}) \bigcirc \left\langle \boxed{\text{n}} \begin{bmatrix} \mathit{mem\text{-}cost} & \\ \text{PHON} & \boxed{\text{s}} \\ \text{HD} & \boxed{\text{h}} \\ \text{DIST} & 0 \end{bmatrix} \right\rangle$$

(ii) $\boxed{\text{q}}$ is the smallest list of *mem-costs* such that

$$\boxed{\text{q}} \ni \begin{bmatrix} \mathit{mem\text{-}cost} & \\ \text{PHON} & \boxed{\text{s}} \\ \text{HD} & \boxed{\text{h}} \end{bmatrix}$$

(iii) $\forall \boxed{\text{x}} \in \boxed{\text{v}} (\boxed{\text{x}} \prec \boxed{\text{w}})$

(iv) $(\boxed{\text{u}} = \mathit{head\text{-}dtr} \wedge \boxed{\text{r}} = \mathit{comp\text{-}dtr}) \vee (\boxed{\text{u}} = \mathit{comp\text{-}dtr} \wedge \boxed{\text{r}} = \mathit{head\text{-}dtr}) \vee (\boxed{\text{u}} = \mathit{adj\text{-}dtr} \wedge \boxed{\text{r}} = \mathit{head\text{-}dtr})$

(v) $\boxed{\text{n}} \notin \boxed{\text{m}}$

(vi) $\neg(\boxed{\text{w}} = \mathit{matrix\text{-}pred\text{-}splu})$

Condition (i) says that $\boxed{\text{l}}$, the LOC-VAL value of the *splu* into which the memory cost $\boxed{\text{n}}$ predicting the corresponding head is introduced, must be the smallest list including this memory cost. This is because it may contain other list elements derived as a result of multiply applying the MCP or the principle for a relativized nominal formation defined in (14). Condition (ii) helps eliminate from BASE-STACK of the head's *splu* the prediction for the head, i.e. the memory cost originating from $\boxed{\text{n}}$. As with Condition (i), this *memory-cost* is not the only element to be popped off from BASE-STACK, since the same *splu* may udergo the MCP or (14) repeatedly.

Condition (iii) limits the application of this principle to cases in which the *unsat(isfied)-sign* precedes the *req(uired)-sign*. Condition (iv) prescribes what can be the *unsat-sign* and *req-sign*: the principle can apply to a complement and its head when the complement either precedes or follows the head. But an adjunct and its head are subject to the principle only when the former occurs before the latter, since a head does not necessarily call for a following adjunct. Condition (v) is needed to prohibit more than one complement from introducing a *mem-cost* to be popped out by one and the same head. Thus in a right-branching structure as in (6), the first complement C_1 may cause to exist a memory cost predicting for a head H, but the second complement C_2 is forbidden to repeatedly make the same prediction.

(6) $[C_1 \ \ [C_2 \ \ H]]$

The last condition (vi) constrains a new memory cost not to be introduced at an *splu* which is a constituent of the matrix sentence: the principle does not apply to the constituent of the matrix sentence, since the prediction of the head of the matrix sentence, i.e. the matrix verb, is assumed to be costless (Gibson 1998 and Babyonyshev & Gibson 1999).

In (5), the new *mem-cost* $\boxed{\text{n}}$ appears within the leftmost element of the *unsat-sign*'s DOM list. Owing to this specification, the *mem-cost* is introduced at the first constituent of the *unsat-sign*. Its DISTANCE value, at first set to 0, is increased by one by the function *increase-by-1* defined as in (7) each time a new input is processed, and finally the *mem-cost* is popped out from the list when the head of the *req-sign* is met.[4] The H(EA)D-DOM feature is used to percolate the DOM feature of the head constituent to the whole *req-sign*.

(7) $\textit{increase-by-1}\,(\langle hd(\boxed{1}, \boxed{n_1}), hd(\boxed{2}, \boxed{n_2}), \ldots, hd(\boxed{\text{i}}, \boxed{n_\text{i}})\rangle)$
$\overset{def}{=} \langle hd(\boxed{1}, \boxed{n_1} + 1), hd(\boxed{2}, \boxed{n_2} + 1), \ldots, hd(\boxed{\text{i}}, \boxed{n_\text{i}} + 1)\rangle$

By (5), the introduction of a new *mem-cost* is limited to a constituent which is not a personal pronoun. This is because, both in English and Japanese, an embedded clause with a personal pronoun case phrase is easier to process than a clause with a full NP (Babyonyshev & Gibson 1999):[5]

(8) The pictures which the photographer **I** met yesterday took were damaged by the child.

As in (5) and the other following specifications, the value of STACK in an *splu* is obtained on the basis of that of PREV(IOUS)-STACK representing the STACK value of the immediately preceding *splu*. The relationship between the two feature values is established by the following rule common to Japanese and English specifying the interdependency between two adjacent *splu*s.

(9) STACK Adjacency Rule
For any pair of adjoining elements of the DOM value list $\boxed{\text{i}}$ and $\boxed{\text{j}}$ such that $\boxed{\text{i}} \prec \boxed{\text{j}}$,

$$\boxed{\text{i}} = \begin{bmatrix} \textit{splu} & \\ \text{SPL-INF|STACK} & \boxed{\text{s}} \end{bmatrix} \wedge \boxed{\text{j}} = \begin{bmatrix} \textit{splu} & \\ \text{SPL-INF|PREV-STACK} & \boxed{\text{s}} \end{bmatrix}.$$

For Japanese, the SPL-related information of complements and adjuncts is formed together with (5) by the lexical information of the

[4] Throughout this paper, accounts are often given as if the processing were performed procedurally from left to right. But of course, they are just metaphors to enhance intelligibility.

[5] Babyonyshev and Gibson's statement that the difference in processing complexity derives from the newness/oldness distinction in the introduced discourse referents is incorrect.

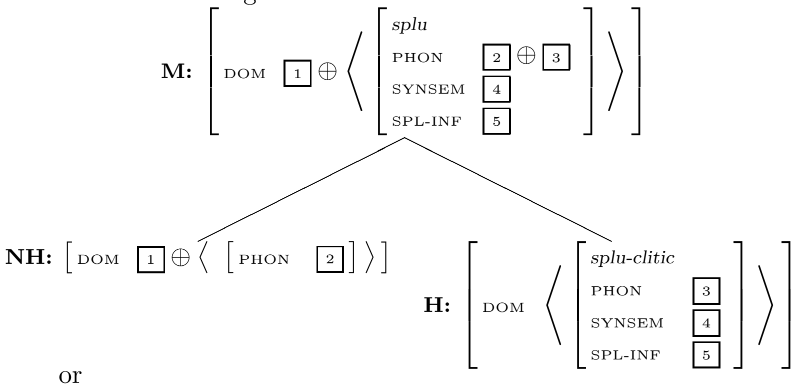
postposition:

(10)
$$\textit{nonhead-postposition} \longrightarrow \begin{bmatrix} \text{DOM} \left\langle \begin{bmatrix} \textit{splu-clitic} & \\ \text{SPL-INF} & \begin{bmatrix} \text{LOC-VAL} & \boxed{1} \\ \text{STACK} & \textit{increase-by-1}(\boxed{2}) \oplus \boxed{1} \\ \text{BASE-STACK} & \boxed{2} \end{bmatrix} \end{bmatrix} \right\rangle \end{bmatrix}$$

If the value of BASE-STACK is not specified explicitly by the principles defined in (5) or (14), it unifies with that of PREV-STACK by default. An *splu* is constructed in Japanese by the Order Domain Principle below. The principle is divided into two parts, *compaction* which is applied to cases in which the head is a function word with a clitic status and *liberation* which applies otherwise. The distinction has been made so that it can work in parallel with a principle to form an accentual phrase (AP), a tonal domain fundamental in Japanese phonology, since in Japanese APs are also basic units of scrambling (See Yoshimoto 2000. See also Gunji 1999 and Chung & Kim 2002.).

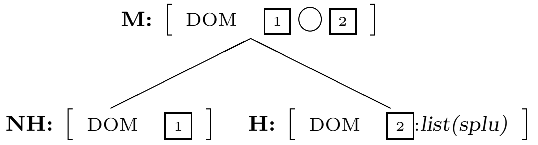
(11) Order Domain Principle
The mother's DOM feature is obtained either

(i) by compaction
If the head is a grammatical word:


M: [DOM 1 ⊕ ⟨[splu, PHON 2 ⊕ 3, SYNSEM 4, SPL-INF 5]⟩]
NH: [DOM 1 ⊕ ⟨[PHON 2]⟩]
H: [DOM ⟨[splu-clitic, PHON 3, SYNSEM 4, SPL-INF 5]⟩]

or

(ii) by liberation
If the head consists of (a) lexical word(s) possibly followed by grammatical words:



(i) compacts the last element of the non-head daughter and the functional head into one *splu.* (ii) shuffles the DOM lists of both daughters: the relative order within each list is observed, but otherwise elements of both lists can be mixed up with the caveat that they obey the Japanese linear precedence rule.

For English, let us assume that every word carries SPL-reated information and thus is an *splu*, following Gibson (1998).

Figure 1 shows how the MCP is applied to process a simple example (12).

(12) Haha wa [sensei ga seito wo tazuneru to
mother TOP teacher SBJ pupil OBJ visit QUOT
it ta] to omot-te iru.
say PAST QUOT think-PROG
'Mother thinks that the teacher said that he would visit the pupil.'

Given that the matrix clause subject involves no memory cost predicting the main predicate, as mentioned above, only the analysis of the parenthesized part of the sentence is shown.

Observe that a *mem-cost* introduced into STACK in processing *sensei ga* ('teacher-SBJ'), $v(\boxed{2}, 0)$, has its DISTANCE value increased by one each as *seito wo* ('pupil-OBJ') and *tazuneru to* ('visit-QUOTATIVE') are read in, and is finally eliminated from the STACK value when its counterpart head (i.e., the *required-sign*) *it-ta* ('say-PAST') is processed. In a similar manner, a prediction for the innermost predicate, $v(\boxed{4}, 0)$, is introduced at *seito wo*, but immediately popped off when the predicate *tazuneru* is scanned.

A question might have come up to the reader by now: *Why on earth the* DOM *feature?* This feature was developed by Reape (1994) and Kathol (2000) to cope with word order, and as such it originally has nothing to do with the complexity problem discussed in this paper.

The answer is as follows. By separating idiosyncrasies in word order from other factors we can capture the commonalities and differences between Japanese and English, because the main distinction between the two languages in terms of this issue depends completely on word order (or linearity). And it is by extending the DOM feature already available that we can most easily cope with the problem intricately involved with word order without overlapping. Furthermore, as discussed in Section 22.6, metrical boost, a prosodic marking of a non-default branching, can be accounted for by resorting to memory costs. Given the close relationship of prosodic representation to the DOM feature (see Yoshi-

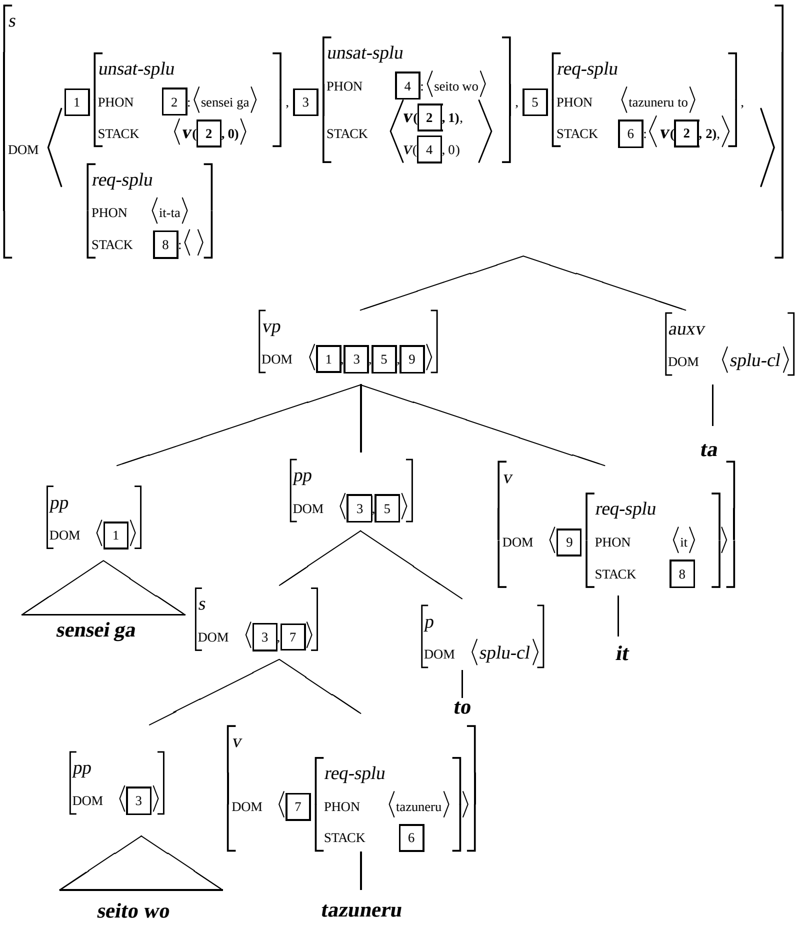


FIGURE 1 Analysis of a Part of Sentence (12)

moto 2000), it is reasonable to deal with the SPL-related information within this feature too.

The proposal offered in this paper is to add constraints to simulate processing load, relying on sentence processing performed within the framework of the standard HPSG syntax. For instance, according to the analysis of example (12) illustrated in Figure 1, the nominal phrase *sensei ga* ('teacher-SBJ') is interpreted as the subject of the predicate *it-ta* ('say-PAST') — and simultaneously as that of *tazuneru* ('visit') — following one possible analysis, while it may also be related only to *tazuneru* and the outer predicate *it-ta* may have a zero pronominal subject referring to another entity. In this manner, (partial) ambiguity of a constituent being processed is not dealt with in this framework, assuming that it is disambiguated by the HPSG syntax. By contrast Dynamic Syntax (Kempson et al. 2001) makes possible a representation underspecified in terms of its syntactic status. Whereas this approach may draw some important generalizations about head-final languages including Japanese, it will not be further discussed in this paper.

22.5 Types of Relative Clause Embedding and Processing Difficulty

According to our analysis of Japanese relative clauses, the prediction of a counterpart head nominal is introduced into the STACK feature. This is supported by the results of Babyonyshev & Gibson's (1999) experiments which showed that the prediction affects comprehensibility. In the examples cited below, a construction (13b) with a sentential complement within a relative clause is much harder than a reverse embedding structure (13a). The difference can be accounted for by the longer distance in (13b) than in (13a) from the *pro* introduced by the doubly embedded relative clause predicate to the corresponding nominal head. During this procedure, it is assumed that the prediction for the nominal head is retained.

(13) a. Dōryō ga [kowai jōshi ga [[raikyaku ga ***pro***
coworker SBJ strict boss SBJ visitor SBJ (OBJ)
mushishi- **ta**] **hisho**] wo hihanshi- ta to]
ignore PAST secretary OBJ criticize PAST QUOT
it- ta.
say PAST
'The coworker said that the strict boss criticized the secretary whom the visitor ignored.'

b.

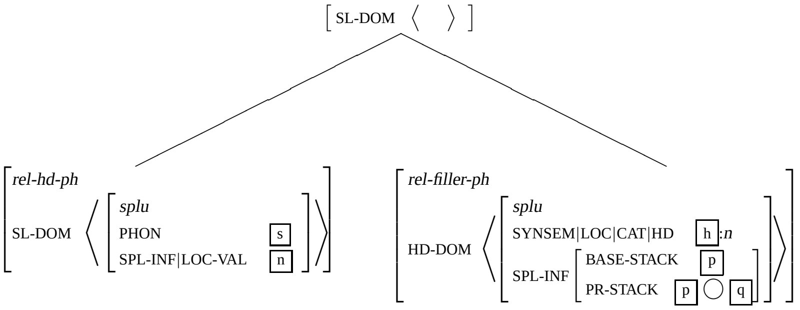
??Kōchō	ga	[[sensei	ga	[shōjo	ga	***pro***	**tsunet-ta**]
principal	SBJ	teacher	SBJ	girl	SBJ	(OBJ)	pinch-PAST

to	**it-**	**ta**]	**otonashii**	**shōnen**]	wo	seme-	ta.
QUOT	say	PAST	well-behaved	boy	OBJ	blame	PAST

'The principal blamed the well-behaved boy whom the teacher said that the girl pinched.'

A principle different from the MCP in (5), the Relative Clause Memory Cost Principle (RMCP) defined as (14), applies to a relative clause–nominal head construction. As shown below, the *mem-cost* or the prediction for a nominal head is introduced by means of the SL(ASHED)-DOM feature when the predicate which possesses a gapped case phrase is read in. The feature value is propagated from a gapped constituent to another each time the Nonlocal Feature Principle applies to pass the information on the gap (Pollard & Sag 1994), until the gap is discharged. The memory cost does not first come into existence at the leftmost constituent as defined in (5), since it would make a wrong prediction for the memory load. For example, in (13b), the prediction of the head nominal would be introduced when *sensei ga* ('teacher-SBJ') is processed, resulting in a memory load much heavier than in practice.

(14) Relative Clause Memory Cost Principle



Conditions:

(i) $\boxed{\text{n}}$ is the smallest list of *mem-cost*s such that

$$\boxed{\text{n}} \ni \begin{bmatrix} \textit{mem-cost} & \\ \text{PHON} & \boxed{\text{s}} \\ \text{HD} & \boxed{\text{h}} \\ \text{DIST} & 0 \end{bmatrix}$$

(ii) $\boxed{\text{q}}$ is the smallest list of *mem-cost*s such that

$$\boxed{\text{q}} \ni \begin{bmatrix} \textit{mem-cost} & \\ \text{PHON} & \boxed{\text{s}} \\ \text{HD} & \boxed{\text{h}} \end{bmatrix}$$

Figure 2 is a part of the result of processing example (13a).[6] The principle (14) applies when combining the relative clause *raikyaku ga mushishi-ta* ('the visitor ignored') and the nominal head *hisho* ('secretary'). Since *mushishi-ta* is the predicate which triggers the gapped case phrase and at which the prediction for the head nominal is introduced, its DOM value $\langle\boxed{8}\rangle$ unifies with that of the SLASHED-DOM feature of the relative clause. The memory cost in terms of the relative clause is represented by $n(\boxed{9}, 0)$.

Figure 3 partially summarizes how memory costs are obtained concerning sentence (13b). As shown in the figure, the head predicate of the gapped NP, *tsunet-ta* ('pinch-PAST'), is more deeply embedded than *mushishi-ta* in (13a); the memory cost predicting for a nominal head accordingly stays longer in the STACK value, resulting in $n(\boxed{6}, 2)$ in $\boxed{10}$ (at *otonashii* 'well-behaved') with the maximum DISTANCE value, which is much higher than that of (13a). The difference accounts for the processing difficulty observed for (13b) but not for (13a).

Note that both of the elements in the STACK list of the *splu* corresponding to *otonashii* ('well-behaved'), $n(\boxed{6}, 2)$ and $n(\boxed{11}, 0)$, disappear within the subsequent *splu* for *shōnen wo* ('boy-OBJ'). This is the result of the applications of both the RMCP, triggered when *shōjo ga tsunet-ta to it-ta* is combined with *otonashii shōnen*, and the MCP, which comes to work when the adjective *otonashii* is paired with its head *shōnen*.

22.6 Metrical Boost

In this section a possibility is pointed out to extend the approach which has hitherto been proposed to cope with relationships between the syntactic information on memory load and phonology. Kubozono (1987) has shown in his statistical phonetic experiments that, in a phrase with multiple modifiers, a modifyng nonhead (corresponding to an *unsat-sign* in this paper) has different pitch levels, depending on whether the phrase structure is right-branching or left-branching. He calls this phenomenon *metrical boost*. In the examples in Figures 4 and 5 each with two modifiers, the peak of the second accentual phrase (AP) *óoki-na* ('big'), which occurs in the right-branching structure in Figure 4, is significantly higher than that of *rémon no* ('lemon-GENITIVE'), an adjunct on the left-branching structure in Figure 5.

Choi et al. (1995) observe a similar difference in peaks of adverbial

[6] In both Figures 2 and 3, the past tense marker *ta*, standardly given an independent auxiliary verb status, is analyzed as if it were a verbal suffix. This is just for the simplification of the tree and causes no essential difference.

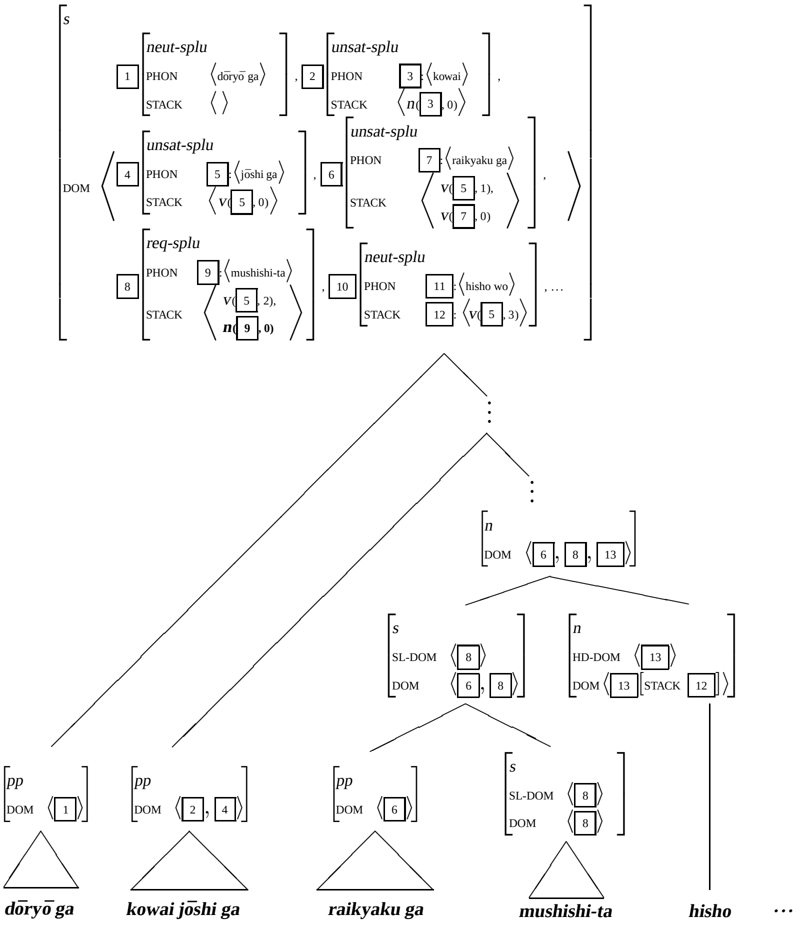


FIGURE 2 Analysis of Sentence (13a)

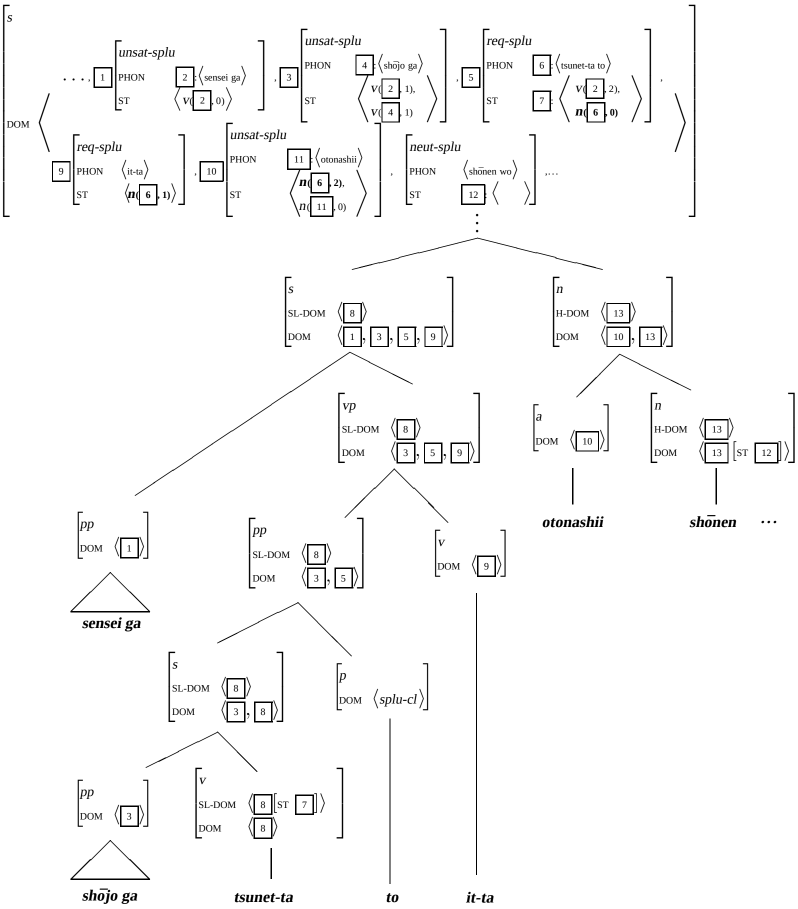


FIGURE 3 Analysis of Sentence (13b)

phrases in syntactically ambiguous sentences and demonstrate that the pitch difference is used to disambiguate the sentence. They also report that Korean, Mongolian, and Turkish employ prosodic means to resolve the ambiguity in sentences with structures parallel to their Japanese examples. Traditionally it has been assumed that the syntactic structure solely affects intonational phrasing, and accordingly has a strictly limited influence on prosody. In contradiction to this belief, the findings cited above show that the influence is much more direct and the formulation of an interface that transmits information on the syntactic hierarchy to phonology is called for.

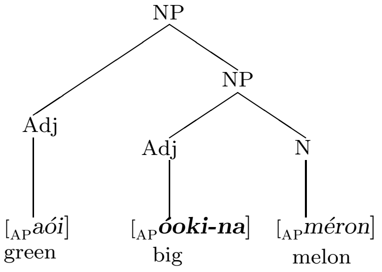
A hypothesis is put forward in this paper that metrical boost is an influence on prosodic information exerted by the information on the meory load; it signals marked, more memory-burdening branching (i.e., right-branching for Japanese), in other words deviation from unmarked, less memory-burdening branching (i.e., left-branching for Japanese). The following is a constraint for differentiating phonological information based on the STACK values of *splu*s:

(15)

$$\begin{bmatrix} \textit{unsat-splu} & \\ \text{SPL-INF} & \begin{bmatrix} \text{STACK} & (\boxed{1}:\textit{list(mem-cost)}) \oplus \left\langle \begin{bmatrix} \textit{mem-cost} & \\ \text{PHON} & \boxed{2} \\ \text{HEAD} & \boxed{3} \\ \text{DIST} & 1 \end{bmatrix} \right\rangle \\ \text{PREV-STACK} & \boxed{1} \oplus \left\langle \begin{bmatrix} \textit{mem-cost} & \\ \text{PHON} & \boxed{2} \\ \text{HEAD} & \boxed{3} \\ \text{DIST} & 0 \end{bmatrix} \right\rangle \end{bmatrix} \end{bmatrix}$$

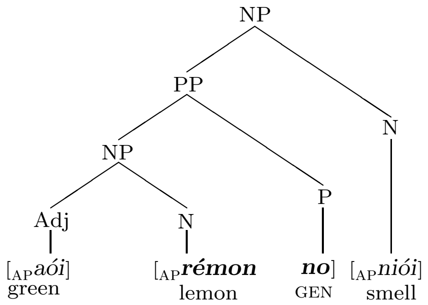
$$\longrightarrow \begin{bmatrix} \text{PHON}|\text{ACC-PROP}|\text{BOOST} & \textit{level-1} \end{bmatrix}$$

If the memory cost deriving from the immediately preceding *splu*



‘(lit.) a green, big melon’

FIGURE 4 Right-Branching Structure



‘the smell of a green lemon’

FIGURE 5 Left-Branching Structure

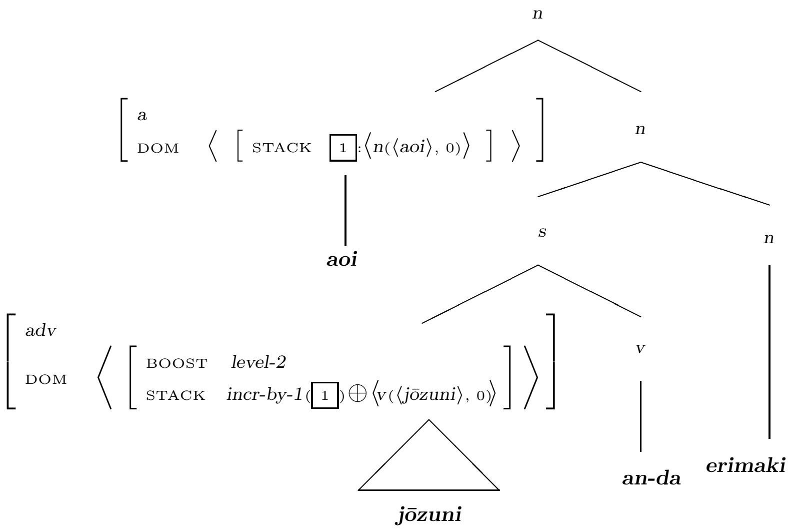

n
a
DOM ⟨ [STACK [1]:⟨n(⟨aoi⟩, 0)⟩] ⟩
n
aoi
s
n
adv
DOM ⟨ [BOOST level-2
STACK incr-by-1([1]) ⊕ ⟨v(⟨jōzuni⟩, 0)⟩] ⟩
v
an-da
erimaki
jōzuni

FIGURE 6 Memory Load in Phrase (16)

is not eliminated and remains within the current *splu*'s STACK, then its PHON|ACC(ENTUAL)-PROP(ERTY)|BOOST feature has a value *level-1*, which stands for a higher pitch than a default value *level-0*.

The hypothesis has the advantage of being able to account for metrical boosts within NPs with three modifying nonheads. Kubozono (1987) observed in his experiments that of the 4 possible syntactic structures with three modifiers, only the structure of the type below

(16)	$[_N$ aoi	$[_N [_S$ **jōzuni**	an-	da]	erimaki]]
	blue	skillfully	knit	PAST	muffler

'(lit.) the blue, skillfully knit muffler'

has a boosted pitch, which is even higher than other boosted phrases, on *jōzuni* ('skillfully') occurring at the left edge of the two embedded subtrees. According to my formalization, both the MCP and RMCP are applied to process example (16), giving the analysis in Figure 6 (irrelevant memory costs are left out).

Owing to this doubly embedded syntactic structure, *jōzuni* has a heavier memory load than others, i.e. two memory costs, each with DISTANCE values 1 and 0, predicting the nominal head and the verbal head. The following specification infers a BOOST value *level-2*, representing a higher pitch than *level-1*, from the value of STACK when this condition is met:

(17)

$$\begin{bmatrix} \textit{unsat-splu} & \\ \text{SPL-INF} & \begin{bmatrix} \text{STACK} & (\boxed{1}:\textit{list(mem-cost)}) \oplus \left\langle \begin{bmatrix}\textit{mem-cost} & \\ \text{PHON} & \boxed{2}\\ \text{HEAD} & \boxed{3} \\ \text{DIST} & 1\end{bmatrix}, \begin{bmatrix}\textit{mem-cost} & \\ \text{DIST} & 0\end{bmatrix} \right\rangle \\ \text{PREV-STACK} & \boxed{1} \oplus \left\langle \begin{bmatrix}\textit{mem-cost} & \\ \text{PHON} & \boxed{2}\\ \text{HEAD} & \boxed{3}\\ \text{DIST} & 0\end{bmatrix}\right\rangle \end{bmatrix}\end{bmatrix}$$

$$\longrightarrow \begin{bmatrix}\text{PHON|ACC-PROP|BOOST} & \textit{level-2}\end{bmatrix}$$

Kubozono (1987) found that the highest metrical boost observed for (16) does not occur within other structures with three modifiers.

(18) a.

[$_{\text{N}}$ [$_{\text{GP}}$ [$_{\text{N}}$ [$_{\text{GP}}$ Naoko	no]	ani]	no]	[$_{\text{N}}$ **aoi**	erimaki]]
NAME	GEN	brother	GEN	blue	muffler

‘Naoko’s big brother’s blue muffler’

b.

[$_{\text{N}}$ [$_{\text{GP}}$ Mariko	no]	[$_{\text{N}}$ **ōkina**	[$_{\text{N}}$ **aoi**	erimaki]]]
NAME	GEN	big	blue	muffler

‘Mariko’s big, blue muffler’

c.

[$_{\text{N}}$ [$_{\text{GP}}$ [$_{\text{N}}$ [$_{\text{GP}}$ Ayako	no]	[$_{\text{N}}$ [$_{\text{GP}}$ **men**	**no**]	erimaki]]	no]
NAME	GEN	cotton	GEN	muffler	GEN
iromoyō]					
design					

‘design of Ayako’s cotton muffler’

Kuboznono’s observation squares with the predictions by rules (15) and (17). *Aoi* in (18a), *ōkina* and *aoi* in (18b), and *men no* in (18c) are all given a BOOST value *level-1*, since *splu*s corresponding to them has a STACK value of the type in (15). Into these *unsat-splu*s no new memory cost is introduced: the prediction of the same head as the preceding constituent’s is prohibited by Condition (v) of the MCP in (5).

Thus the DOM feature can serve as an interface which transmits the information on the syntactic hierarchy to the phonological component. The proposal is also motivated by the relationship of this feature to prosodic information (Yoshimoto 2000).

22.7 Conclusions

It has been demonstrated that a linear syntax with additional information on the memory costs of anticipated heads can account for the issue of sentence complexity caused by multiple clause embedding. It has also been suggested that the DOM feature can be expanded to an interface where the linear aspect of syntax and prosodic information meet. The proposal, still being at a seminal stage, paves the way for an integrated linguistic model which sheds light on diverse linguistic issues based on processing efficiency in human language processing: they include word order discussed by Joshi (1990) and Rambow & Joshi (1994) and garden path sentences with which sentence's complexity is known to be involved with.

Acknowledgements

I would like to thank Masahiro Kobayashi, Yoshiki Mori, Hiroaki Nakamura, Takashi Ono, and Shūichi Yatabe for suggestions and comments.

References

Babyonyshev, M. and E. Gibson. 1999. The Complexity of Nested Structures in Japanese. *Language* 75:423-450.

Choi, Y.-S. and S. Satō and G.-J. Min. 1995. Tōgo–teki ni Aimaina Bun no Seisei ni okeru Inritsu-teki Yōso no Bunseki. *Proceedings of the Phonetic Society of Japan 1995 Annual Convention*, 26-31.

Chomsky, N. and G. Miller. 1963. Introduction to the Formal Analysis of Natural Languages. *Handbook of Mathematical Psychology*, R. Luce et al., eds. New York: John Wiley, 269-321.

Chung, C. and J.-B. Kim. 2002. Mismatch in Korean Copula Constructions and Linearization Effects. *Proceedings of PACLIC 16*, I.-H. Lee et al., eds. The Korean Society for Language and Information, 36-49.

Gibson, E. 1998. Linguistic Complexity: Locality of Syntactic Dependencies. *Cognition* 68:1-76.

Gunji, T. 1999. On Lexicalist Treatments of Japanese Causatives. *Studies in Contemporary Phrase Structure Grammar*, R. Levine and G. Green, eds. Cambridge: Cambridge University Press, 119-160.

Hawkins, J. 1994. *A Performance Theory of Word Order and Constituency.* Cambridge: Cambridge University Press.

Joshi, A. 1990. Processing Crossed and Nested Dependencies: An Automation Perspective on the Psycholinguistic Results. *Language and Cognitive Processes* 5:1-27.

Kathol, A. 2000. *Linear Syntax.* Oxford: Oxford University Press.

Kempson, R. et al. (2001) *Dynamic Syntax: The Flow of Language Understanding.* Oxford: Blackwell.

Kruijff, G.-J., and S. Vasishth. 2001. *Competence and Performance Modelling of Free Word Order.* 13th European Summer School in Logic, Language and Information, lecture notes. Helsinki, August 13-24.

Kubozono, H. 1987. *The Organization of Japanese Prosody.* University of Edinburgh doctoral thesis; Published 1993 Tokyo: Kurosio.

Lewis, R. and M. Nakayama. 2001. Syntactic and Positional Similarity Effects in the Processing of Japanese Embeddings. *Sentence Processing in East Asian Languages*, M. Nakayama, ed. Stanford: CSLI Publications, 85-111.

Mazuka, R. and K. Itoh. 1995. Can Japanese Speakers Be Led Down the Garden Path? *Japanese Sentence Processing*, R. Mazuka and N. Nagai, eds. Hillsdale, N.J.: Lawrence Erlbaum, 295-329.

Pollard, C. and I. Sag. 1994. *Head-Driven Phrase Structure Grammar.* Stanford: CSLI.

Rambow, O. and A. Joshi. 1994. A Processing Model for Free Word-Order Languages. *Perspectives on Sentence Processing*, C. Clifton et al., eds. Hillsdale, N.J.: Lawrence Erlbaum, 267-301.

Reape, M. 1994. Domain Union and Word Order Variation in German. *German in Head-Driven Phrase Structure Grammar*, J. Nerbonne et al., eds. Stanford: CSLI Publications, 151-197.

Yoshimoto, K. 2000. A Bistratal Approach to the Prosody-Syntax Interface in Japanese. *Grammatical Interfaces in HPSG*, R. Cann et al., eds. Stanford: CSLI Publications, 267-282.