Parsing in the Minimalist Program:

On SOV Languages and Relativization

Sandiway Fong[1]
Departments of Linguistics and Computer Science
University of Arizona
Tucson AZ USA
sandiway@email.arizona.edu

Abstract

This paper examines computational issues in the processing of SOV languages in the *probe-goal* framework, a theory in the Minimalist Program (MP). A generative theory that seeks to minimize search, such as the probe-goal model, provides a strong linguistic basis for the investigation of efficient parsing architecture. For parsing, two main design challenges present themselves: (1) how to limit search while incrementally recovering structure from input without the benefit of a pre-determined lexical array, and (2) how to come up with a system that not only correctly resolves parsing ambiguities in accordance with empirical data but does so with mechanisms that are architecturally justified. We take as our starting point an existing probe-goal parser with features that allows it to compute syntactic representation without recourse to derivation history search. We extend this parser to handle pre-nominal relative clauses of the sort found in SOV languages. In doing so we tie together and provide a unified computational account of facts on possessor (and non-possessor) relativization and processing preferences in Turkish, Japanese and Korean.

---

[1] The author gratefully acknowledges discussion and data from Nobuko Hasegawa,Yuki Hirose, Cağlar Iskender, So Young Kang, Shigeru Miyagawa and Kyung Sook Shin. Parts of this paper have been presented at the *Conference on Interfaces*, July 30th–August 1st 2004, Pescara, Italy, and the MIT IAP *Computational Linguistics Fest*, January 14th 2005.

Introduction

Recent proposals in the framework of the Minimalist Program (MP), e.g. (Chomsky 1998,1999), have highlighted the role of efficient, locally deterministic computation for the assembly of phrase structure. From a generative standpoint, phrase structure assembly proceeds in bottom-up fashion using the primitive combinatory operations of MERGE and MOVE selecting from a domain of pre-determined lexical items known as a *lexical array* (LA). The set of LA items available and their lexical properties and features limit the combinatory options and hence possible phrase structure. Further limits on phrase structure result from the interaction of heads known as *probes* and *goals*. In Chomsky's formulation of the Case-agreement system, probes, e.g. functional heads such as T and v*, target and *agree* with goals, e.g. referential and expletive nominals, within their c-command domain and value their *uninterpretable* Case features. Within this system, Case-agreement can be long-distance and does not necessarily trigger movement, e.g. in the case of *there*-expletive constructions and Icelandic Quirky Case.

In the case of parsing systems that aim to implement MP models of the kind outlined above, we can identify two major design challenges that should be met.

First, the proposed parser architecture should support similar design goals to the original (generative) model in the sense that computation should be driven by lexical properties and features, and be locally deterministic where possible. However, the fact that the generative model is not a directly viable model of parsing (in a sense to be made clear below) means that the efficient recovery of structure is not guaranteed. More specifically, if we assume that a parser should process input from left to right and incrementally build phrase structure, the two operations, MERGE and MOVE, that lie at the heart of the (bottom-up) generative model cannot be employed directly. Moreover, for a parser there exists no pre-determined LA. It must attempt to efficiently reconstruct the participating lexical and functional elements (possibly covert) solely on the basis of overt input and its knowledge of grammar. Finally, parser architecture needs to provide support for efficient computation of probe-goal agreement relations. In an ideal model, a probe would identify its goal (or goals) without invoking search, i.e. without sifting through the derivation history represented by constructed phrase structure.

The second design challenge concerns temporary ambiguities encountered in the course of the recovery of structure. Temporary ambiguities will manifest themselves as computational choice points. Numerous architectural options are available to the parser designer. However, in the ideal case, a parser should always resolve temporary ambiguities in favor of the (locally) least expensive computational option.[2]

In this paper, we focus on the computational issues involved in meeting the second design challenge. We will present a parsing model and provide cross-linguistic empirical support for its proper operation. We take as our starting point a left-to-right, incremental parser in the MP framework (Fong 2005). This implemented parser is designed to recover phrase structure in accordance with Chomsky's probe-goal model for the Case-agreement system as described in (Chomsky 1998). The parser includes architectural features that allow search to be minimized in the computation of probe-goal relations; thus facilitating efficient computation in the sense of the

[2] Note that this does not necessarily imply that the parser need select the globally least expensive option. More to the point, we are not advocating a return to a parsing model based on some metric from a modern formulation of the *derivational theory of complexity* hypothesis (Miller and Chomsky, 1963).

first design challenge identified above. We propagate this design efficiency into the realm of parsing preferences by appealing to computational cost reduction and simplicity.

We extend the probe-goal architecture, paying special attention to SOV language data in the area of possessor (and non-possessor) relativization. More specifically, we show how a parser that seeks to minimize search when faced with temporary ambiguity can account for and tie together independent facts on relativization with respect to *bare* (i.e. non-Case-marked) noun phrases (BNPs) in Turkish and object scrambling in Japanese and Korean. We propose that the same (possibly universal) mechanism that resolves subject-object ambiguity in the case of Turkish BNPs is also at work in the case of (Case-marked) object scrambling in Japanese and Korean.

The remainder of the paper is organized as follows. First, we will briefly review and highlight relevant design features of the probe-goal parser described in (Fong, 2005). Next, we will describe how the basic system can be adapted to accommodate the head-final nature of SOV languages such as Turkish and Japanese. We will then extend the model to include a bottom-up component necessary to accommodate pre-nominal relative clauses in these languages. Finally, we will describe a (non-language-particular) mechanism of relativization motivated by the desire to avoid search and document the empirical support for the proposed model.

Probe-Goal Parser Design

In this section, we will provide an overview of a parser that implements Chomsky's probe-goal model. We will discuss the layout of the lexicon, lay out the basic computational procedure, and highlight architectural features introduced for minimizing search in probe-goal agreement.[3]

The Lexicon

We begin with the lexicon, which lies at the heart of the generative model. Bottom-up computation via MERGE and MOVE is driven by lexical properties such as selection and the need to eliminate uninterpretable features within narrow syntax. We assume the parser operates with, and is propelled by, the same set of properties and features as the generative theory; i.e. the parser does not come with its own set of parsing-specific uninterpretable features.

An illustrative sample of lexical items and their properties is given in Figure 1 below.[4] The property of selection, denoted by select(X) where X is a category, forms the basis for a top-down selection-driven model. For example, sentence parsing begin with the complementizer *c* at the top. *c* selects for tense T, which in turn selects for *v** and a specifier position (shown as spec(select(N))). *v** selects for V plus a sentential subject in specifier position. Finally, V selects for an object nominal N in the simple transitive case. An example of a tree recovered by the parser for the basic transitive sentence *John saw Mary* is given in Figure 2. Note that the

---

[3] For the full details and step-by-step worked examples, see (Fong 2005).

[4] Variants of the categories are not shown here. For example, *v* comes in several flavors containing different subsets of the properties and features of transitive *v**. Both unaccusative and unergative *v* do not have uninterpretable ϕ-features or the EPP option, and lack the ability to value accusative Case (shown as acc). T comes either with a full set of uninterpretable ϕ-features and the ability to value nominative Case (shown as nom), or in a defective version $T_{\bar{\phi}}$ lacking several uninterpretable ϕ-features and Case-valuing ability.

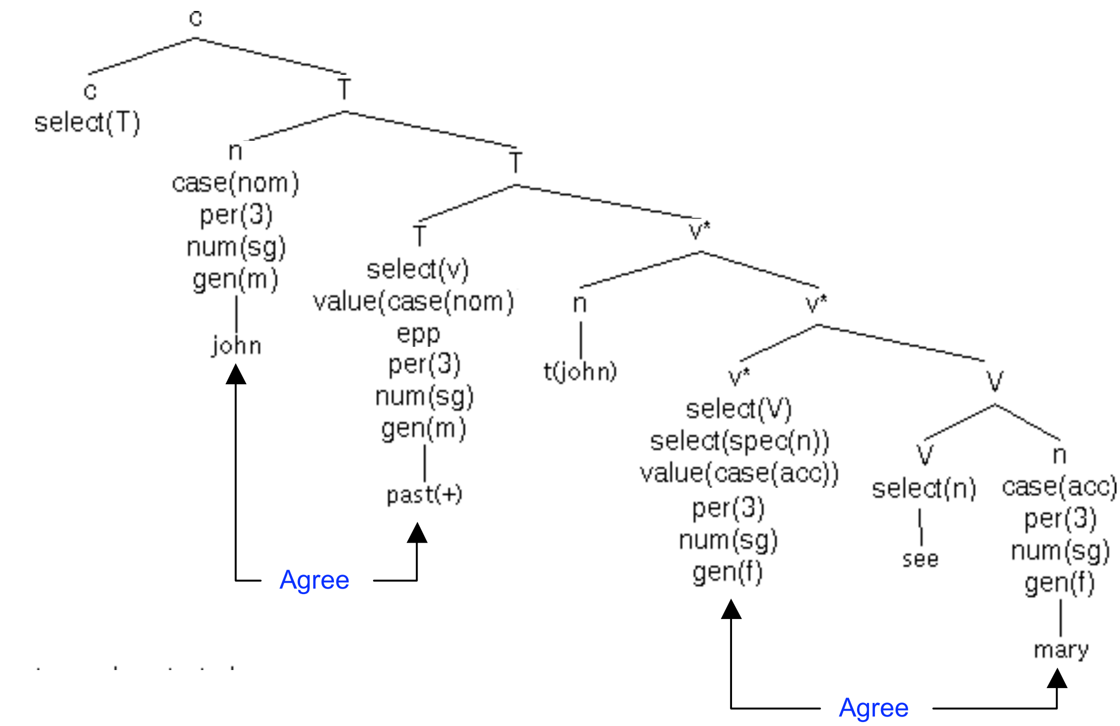
sequence of selection steps (outlined above) extends the derivation from left to right in a similar fashion to (Phillips 1995). Note also that a strict interpretation of bottom-up MERGE and MOVE would result in a right-to-left parse.

Figure 1: A Sample Lexicon

Lexical Item (LI)	**Properties**	**Uninterpretable Features**		**Interpretable Features**
		ϕ-features	**Other**	
v* (transitive)	select(V) spec(select(N)) value(case(acc))	per(_) num(_) gen(_)	(EPP)	
V (transitive) (unaccusative)	select(N) select($T_{\bar{\phi}}$)			
T	select(v) value(case(nom))	per(_) num(_) gen(_)	epp	
c	select(T)			
N (referential)			case(_)	per(P) num(N) gen(G)

Feature matching is an important component of probe-goal agreement and the parsing process. It is also lexically driven. For example, the uninterpretable ϕ-features of the probe *v** (representing person, number and gender) must be matched, valued and cancelled by the parallel interpretable ϕ-features from a (nominal) goal in *v**'s c-command domain. (Uninterpretable features can be viewed as features with *unvalued* slots, depicted here using '_'.) At the same time, the uninterpretable Case feature belonging to the relevant nominal will be valued and cancelled provided the probe has the property of valuing Case. A (valid) parse tree is one that obeys the selectional properties of the lexical items involved, covers the entire input, and no uninterpretable feature remains uncancelled.

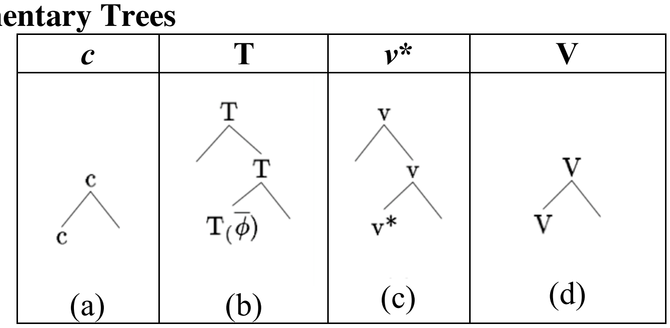
Figure 2: Basic Phrase Structure



Computation with Elementary Trees

Since MERGE and MOVE cannot form the basis for a left-to-right parser model, (Fong 2005) adopts a system driven by elementary tree (ET) composition with respect to a range of heads in the extended verbal projection (*v**, V, *c* and T). ETs are underspecified phrases with structural options determined by lexical properties. They contain *open positions* to be filled by input and movement during the course of parsing. Examples of ETs implied by the lexicon of Figure 1 are given in Figure 3.

Figure 3: Basic Elementary Trees



The *skeletal*, i.e. non-rooted, equivalent of the tree shown in Figure 2 can be formed by the sequential composition of ETs (a) through (d) in Figure 3.[5] As ET composition proceeds from left to right, open positions at the terminal nodes may be filled sequentially as the input is presented. A summary of the implemented ET parsing procedure is given in Figure 4(b). The system is *incremental* in the sense that a partially-specified parse is available throughout all stages of processing.

In order for the tree in Figure 3 to be a valid parse, the *Agree operation* must also connect the probe T with the sentential subject *John*, valuing its Case feature, and similarly, *v** with the object *Mary*. The ϕ-features of probes T and v* are valued from those of the goals *John* and *Mary*, respectively.

Careful consideration must be given to the problem of identifying goals efficiently, given that input items are inserted into open positions in tree structure as soon as they are encountered during parsing. In the framework of this model, goal identification seems to be a *prima facie* case for requiring tree search. As a computationally more attractive option, (Fong 2005) makes use of two short-term memory devices, shown in Figure 4(a), with linguistically well-motivated properties: a Move Box that encodes movement in accordance with theta theory, and a Probe Box that approximate the notion of Phase boundaries.[6] The Agree operation then operates solely

---

[5] ET composition is a basic component of *Tree-Adjoining Grammars* (TAG) (Joshi & Schabes 1997) and theories of morphology such as (Di Sciullo 2002).

[6] The Move Box can be viewed as a (principled) version of Wood's (1970) *ad hoc* HOLD register. It is filled by input material at the head of a chain and emptied at the underlying theta position. Limiting the Move Box to nesting will preserve the no-search model but prevent cases where movement chains may cross. The single Probe Box restriction means that probes cannot "see" past another probe; thereby implementing the Phase Impenetrability Condition (PIC).

in short-term memory space, thereby eliminating *lookback,* or search of the derivation history formed so far.

Figure 4: Short-Term Memory Items and the Parse Procedure

(a) Short-Term Memory Items:

Box	**Description**
Move	contains nominals (possibly nested) *filled when an open position is filled from the input, obeys θ-theory*
Probe	contains a single probe *p* only *filled when an open position is filled by a head that is also a probe*

(b) Parse Procedure:

a. Given a category *c*, pick a ET headed by *c*.
b. From the Move Box *m or* input (put a copy in the Move Box):
   i. Fill in the specifier
   ii. Run Agree(*p*,*m*) if *p* and *m* are non-empty
   iii. Fill in the head *h*. Copy *h* to the Probe Box *p* if *h* is a probe
   iv. Fill in the complement. Repeat from (a) with c' such that *c* select(c')

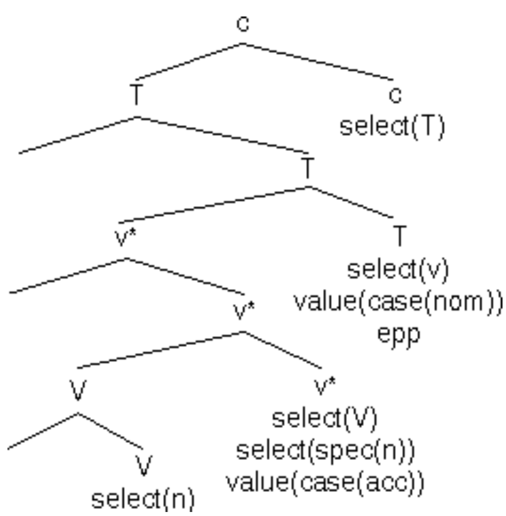
We emphasize the basic system is *on-line* in the sense that once an input element has fulfilled its function and has been placed in structure, the only way to reference it without recourse to search is when a copy exists in one of the Boxes. With respect to the relativization data to be presented below, we will allow the parser to also specifically target the specifier position of T.

SOV Languages and Probe-Goal Parsing

The top-down, selection-driven model outlined above also applies directly to SOV languages like Turkish, Japanese and Korean. Simple canonical SOV structure can be directly accommodated by an equivalent set of ETs to those shown earlier in Figure 3. (We will return to discuss the case of object scrambling later.) The only adjustment is to respect the head-final word order.

An abstract SOV counterpart to the basic transitive structure from Figure 2 is given in Figure 5. The basic parsing mechanism is unchanged, i.e. parses are grown top-down through ET composition, respecting categorial selection. Open positions shown on the left edge of the tree are filled in left to right in an identical fashion to that described for the English example earlier.

Figure 5: Canonical SOV Tree



The only significant modification for the parser lies in the timing of probe-goal Case agreement, given that the relevant probes, T and *v**, are now head-final. Assuming that Case agreement involves the same bilateral exchange of ϕ-features and Case described earlier, nesting of the Move Box will be necessary for the canonical case, and the fill order of the complement and head in Figure 4(b) must be swapped to accommodate the surface order of the elements participating in the Agree operation.[7,8]

Prenominal Relative Clauses and Bottom-Up Parsing

The parser architecture presented so far is a wholly expectation-driven, top-down model. In this section, we show that a bottom-up component to the parser must also be introduced.

Consider the necessary steps a parser must take after processing the input prefix: S O V. Three possible continuations of the parse are given in (1). In (1a), there is no further input: the sentence is a simplex one, and structure of the form given previously in Figure 5 suffices. In (1b) and (1c) however, the prefix is recursively embedded as a complement to a V from a higher clause or adjoined as a relative clause to a higher head N, respectively. Additional structure must be introduced to cover the unexpected input.

(1) a. S O V (*simplex sentence*)
b. [S O V] V (*complement clause*)
c. [S O V] N (*prenominal relative clause*)

In (1b–c), the additional surface cue encountered (N or V) forces the parser out of its expectation-driven model in order to wrap a higher layer of sentence structure around the simplex tree. The extra layer is inserted by a bottom-up component that is activated when the

---

7 Since Japanese does not exhibit overt agreement, proposals exist in the literature that suggest neither T nor *v** induces ϕ-feature agreement, e.g. (Kuroda 1988) and (Fukui and Takano 1998).
8 The linear order of Case-agreement elements is S–O–*v**–T. Hence, O must be stacked on top of S. Agree(*v**,O) takes place first. Afterwards, O is popped off the stack. Finally, Agree(T,S) can be performed.

predictive top-down component fails to incorporate the extra input. For example, consider what must happen in the case of (1c) assuming an operator account of relative clauses. On encountering N, the parser must merge an empty operator Op into the specifier of *c* of the simplex clause, as shown in (2a) below.[9] Next it must generate the template for the higher clause incorporating N as shown in (2b). The relative clause in (2a) must be adjoined to the head N in (2b).[10]

(2) a. [[[Op [[$_T$ S [$_v$ t(S) [$_V$ O V] *v**] T] *c*]]
 b. [[N [_ [_ V]*v*]T]*c*]

Note that no *reanalysis* or *reparse* is required since there is no change in predicate-argument structure for the prefix when it is embedded in a higher clause.[11] In fact, this two step top-down/bottom-up model, in which (possibly unnecessary) structure is only generated in the presence of overt cues, is supported by psycholinguistic evidence. For example, it has been shown that relative clauses in Japanese are initially processed as main clauses with dropped arguments, (Yamashita 1994). In the remainder of the paper, we will restrict our attention to sentences of type (1c), the relative clause case.

---

[9] In (2a), either S or O must be a gap bound by Op. t(S) represents a copy of S left behind in specifier of v* after movement.
[10] In (2b), the underscore character ('_') is used to represent open positions to be filled after N.
[11] Cases of clause boundary ambiguity such as S–O–V–N–…–V, in which the object O may be construed as belonging either to an embedded clause headed by the first V or to the higher clause headed by the second V (depending on various factors including prosody) are not addressed in this paper. See (Hirose 2003) for relevant discussion with respect to Japanese data.

Bare Noun Phrases and Turkish Relative Clauses

Turkish is a SOV language exhibiting rich morphology in the case of relative clause constructions. Relativization is signaled in Turkish via morphological case and an obligatory overt relativizer. The basic paradigm is given in (3a) and (3b) for object and subject relative clauses, respectively.

(3) a. [S-GEN *e* V-OREL-AGR] H (*object relative clause*)
b. [*e* O-ACC V-SREL] H (*subject relative clause*)

In (3), *e* represents the relative gap, and H is the head of the relative clause construction. OREL represents the overt object relativizer –dUk, and SREL the overt subject relativizer –An. Note that in the case of object relatives, the (normally nominative) subject is exceptionally marked with genitive case, as shown in (1a).

In contrast to languages like Japanese, a case can be made that Turkish is a "parser-friendly" language given such rich surface cues for disambiguation. However, SREL does not always signal a subject relative clause, and if we utilize a bare NP (BNP), i.e. a NP unmarked for Case, there is room for ambiguity, as in the case of (4), which can be realized as either (5a) or (5b). In other words, the BNP could be in subject or object position.

(4) BNP V-SREL H

(5) a. [BNP *e* V-SREL] H (*object relative clause*)
b. [*e* BNP V-SREL] H (*subject relative clause*)

(6) Kitap oku-yan adam
Book read-SREL man
"*the man that read a book*"

However, Turkish exhibits a general preference for subject relativization, i.e. analyzing (4) as (5b), (Iskender, p.c.). The BNP appears in object position, and is obligatorily interpreted as an indefinite object NP, as in example (6).[12] This preference is reversed when possessive agreement is added to the BNP: the schema and an example is given in (7) and (8), respectively.

(7) BNP-AGR V-SEL H

(8) Hasta-sI oku-yan adam
patient-AGR3sg read-SREL man
"*the man whose patient read* (*something*)"

We can provide a simple computational account of this behavior. Let us assume that the relativizer SREL triggers (restrictive) relative clause: in particular, a restrictive relative clause should contain a gap. Suppose SREL is a probe that seeks a goal with the restriction that it must be a viable gap. Since both subject and object relative clauses exist, we know that the parser must be able to search for (and find) both types of gaps. However, computational efficiency will favor a subject gap analysis on the assumption that the probe SREL operates by searching tree structure. Put another way, suppose SREL has low-cost access to the specifier of T, as illustrated in Figure 6 by the link *find-e*. Or in terms of the MP framework, SREL must pay a computational penalty for access past the (strong) Phase boundary of *v** into the object position.

This accounts for the preference for the BNP to be interpreted as a non-specific object NP. If the BNP is in object position, the relative gap *e* occupies the favored specifier of T as in (6a). Turning to the case of BNP-AGR (with possessive agreement), the account is maintained if the

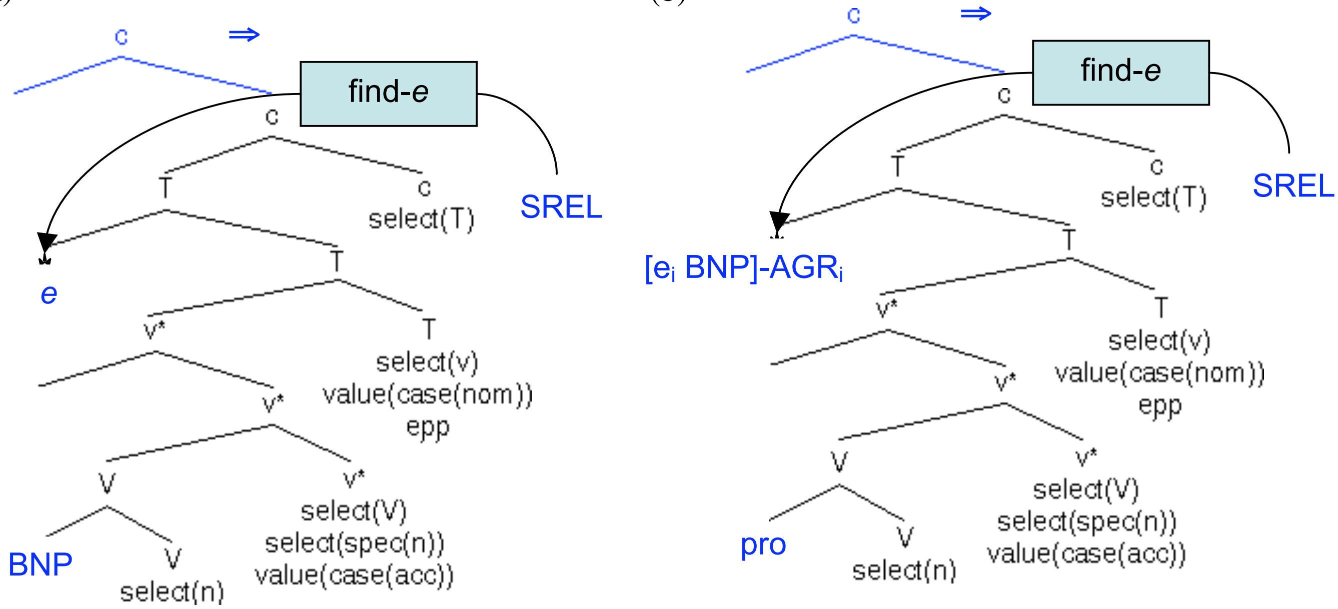
---

[12] There are cases of so-called *pseudo-agent incorporation* for concepts such as “bee sting” or “lightning strike”, where the BNP (“bee” and “lighting”) must be interpreted as a subject, see (Ozturk 2004). In these cases, the preference is for object relativization.

possessive gap e_i can be more easily accessed from the specifier of T than from within the object position. This accounts for the switch in preference for BNP with possessive agreement to subject position, along with an empty pronominal object.

Figure 6: Turkish Relativization and BNPs

(a) (b)



Possessor Relativization in Japanese and Korean

We can find independent evidence for the specifier of T targeting from Japanese and Korean.

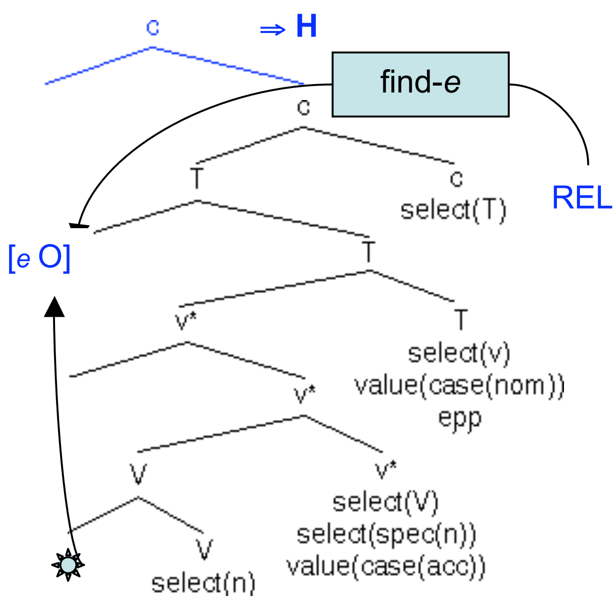
First, Japanese has a clear preference for subject relativization (Miyamoto and Nakamura, 2003): the situation is as in the Turkish BNP case shown in Figure 6(a), with the only differences being that the relativizer is always covert in Japanese and the object is morphologically marked with accusative Case.

Second, scrambling is preferred in the case of possessor of object relativization. That is, (9a) is easier to process than (9b), (Hirose, p.c.).[13]

(9) a. musume-o watashi-ga mita otoko
[*e* daughter]-ACC I-NOM e see-PAST man
b. ?watashi-ga musume-o mita otoko
I-NOM [*e* daughter]-ACC see-PAST man
"*the man whose daughter I saw*"

Assuming that Japanese short-distance scrambling is to specifier of T, as in (Miyagawa, *in press*), the object scrambling preference is also a consequence of placing *e* within reach of *find-e*, as illustrated in Figure 7.[14]

Figure 7: Japanese Possessor of Object Relativization



---

[13] Judgments given are strictly relative. For possessor relativization in Japanese, speakers typically prefer to use a resumptive pronoun instead of a gap as shown in (9) and (10).
[14] MO-phrases may provide a good test for the specifier of T hypothesis. Following (Hasegawa 2005), MO- (English: *also*) is a focus particle, and MO-marked phrases may occupy the specifier of T position. If *find-e* specifically targets the specifier of T position and not just the highest specifier, then relative clauses constructions such as the MO-phrase counterpart of (9a): *musume-o watashi-mo mita otoko*, should be harder to process than (9a).

Furthermore, in the case of possessor of subject relativization, the unscrambled, or canonical, order is preferred. That is, (10a) is easier to process than (10b). This is also predicted by the specifier of T hypothesis.

(10) a. musume-ga watashi-o mita otoko
[*e* daughter]-NOM I-ACC see-PAST man
b. ?watashi-o musume-ga mita otoko
I-ACC [*e* daughter]-NOM see-PAST man
"*the man whose daughter saw me*"

The Japanese data also receive independent confirmation from Korean, (Shin & Kang, p.c.). The counterparts of (9) and (10) are given in (11) and (12), respectively

(11) a. ttal-ul nay-ka po-ass-ten namca
daughter-ACC I-NOM see-PAST-REL man
b. ?nay-ka ttal]-ul po-ass-ten namca
I-NOM daughter-ACC see-PAST-REL man
"*the man whose daughter I saw*"

(12) a. ttal-ka nay-ul po-ass-ten namca
daughter-NOM I-ACC see-PAST-REL man
b. ?nay-ul ttal-ka po-ass-ten namca
I-ACC daughter-NOM see-PAST-REL man
"*the man whose daughter saw me*"

Summary

The specifier of T-centric nature of *find-e* when it comes to SOV languages is supported by several sets of independent evidence, including the BNP object position preference and BNP-AGR subject position preference for Turkish relative clauses. In Japanese and Korean, there is support from a preference for the scrambled order when it comes to possessor of object relativization, with the reverse being true in the case of possessor of subject relativization. Finally, there is a general preference for subject relativization in Japanese.[15]

The empirical evidence for the parsing preferences has also architectural support in the sense that the canonical object position lies deeper in structure than T's specifier position, and therefore less accessible to the relativizer in complementizer position. More specifically, in terms of the probe-goal model, the canonical object position lies deep within *v**, a strong Phase boundary.

References

Chomsky, N. A. *Derivation by Phase*. MITWPL, 1999.

Chomsky, N. A. *Minimalist Inquiries: The Framework*, MITWPL, 1998.

Chomsky, N.A. *Derivation By Phase*. MITWPL 1999.

Di Scuillo, A.-M. The Asymmetry of Morphology. In *Many Morphologies*. Ed. Boucher, P. Cascadilla Press, 2002.

Fong, S. Computation with Probes and Goals: A Parsing Perspective. In Di Sciullo, A. M. and R. Delmonte (Eds.) *UG and External Systems.* John Benjamins. 2005.

Fukui, N. & Y. Takano. *Symmetry in syntax: Merge and demerge*. Journal of East Asian Linguistic*s* 7: 27-86. 1988.

Hasegawa, N. EPP Materialized First, AGREE Later: Subject Positions and Mo-Phrases in Japanese. Handout from the *MIT Workshop on Japanese Korean Linguistics*. January 2005.

Hirose, Y. *Recycling Prosodic Boundaries*. Journal of Psycholinguistic Research Vol 32 (2), 162-195. 2003.

Hsiao, F. & E. Gibson. *Processing Relative Clauses in Chinese*. Cognition (90) 3–27. 2003.

[15] It is tempting to generalize further, given that English also exhibits a subject relativization preference. That is, *find-e* has unparameterized behavior and is an example of a parser universal. However, Chinese is outwardly incompatible with the accessibility proposal, see (Hsiao & Gibson, 2003), also (Miyamoto & Nakamura, 2003). We speculate that this is perhaps due to the mixed word-order nature of Chinese.

Joshi, A. and Y. Schabes. Tree-Adjoining Grammars. In *Handbook of Formal Languages, vol 3*. Eds. Rosenberg, G. and A. Salomaa. pages 69–123. Springer-Verlag, 1997.

Kuroda, S.-Y. Whether we agree or not: A comparative syntax of English and Japanese. In *Papers from the Second International Workshop on Japanese Syntax*, 103-143. CSLI, 1988.

Miller, G. A. & N. Chomsky. Finitary models of language users. In D. R. Luce, R. R. Bush, and E. Galanter, editors, *Handbook of Mathematical Psychology*, Volume II. John Wiley, New York, 1963

Miyagawa, S. On the EPP. *Proceedings of the EPP/Phase Workshop,* N. Richards and M. McGinnis, Eds., MITWPL (in press).

Miyamoto, E. & M. Nakamura. Subject/Object Asymmetries in the Processing of Relative Clauses in Japanese. *Proceedings of WCCFL* 2003.

Ozturk, B. *Agent Incorporation*. ECO5 Syntax Workshop. University of Maryland, College Park. March 6–7 2004.

Phillips, C. *Order and Structure*. Ph.D. thesis. MIT, 1995.

Woods, W. A. *Transition Network Grammars for Natural Language Analysis*. Communications of the Association for Computing Machinery. pages 591–606. 13(10), 1970.

Yamashita, H. (1994). *Processing of Japanese and Korean*. Ph.D. thesis. Ohio State University.