Asymmetry, Zero Morphology and Tractability

Anna Maria Di Sciullo
Department of Linguistics
Université du Québec à Montréal
C.P. 8888, Succursale Centre Ville
Montréal, Qc H3C 3P8
Di_sciullo.anne-marie@uqam.ca

Sandiway Fong
NEC Research Institute, Inc.
4 Independence Way
Princeton, NJ 08540
sandiway@research.nj.nec.com

Abstract

We present a theory of grammar based on asymmetrical relations, the Strict Asymmetry Theory, and we provide evidence form Indo-European and non Indo-European languages to show that argument structure restrictions on morphological composition follow in a principled way from the theory. We describe and implement a bottom-up parser for morphological selection in the Strict Asymmetry framework. Core lexical properties including argument structure and derivational affix selection are encoded in a uniform mechanism. We consider the computational implications of three different implementations. In particular, we examine the effect on bottom-up parsing from varying the Specifier-Head-Complement order. We provide computational motivation for the logical separation of overt and covert affixation.

1 Introduction

We assume a principled-based theory of morphology, (Di Sciullo, 1995), where morphological expressions include Specifier-Head-Complement structures in the course of derivation, and where the Spec, the Head, as well as the Compl can be overt or covert. We show that optimal recovery of morpho-syntactic structure requires the separation of overt and non-overt processing. We also assume that asymmetry inverting operations affect Spec-Head-Compl relations in the derivation of syntactic expressions. Evidence from computational tractability suggests that this is also the case in the derivation of morphological expressions, though in a specific way.

The organization of this paper is as follows. The first section presents basic properties of Strict Asymmetry Theory, (Di Sciullo, 1999), and show that argument structure restrictions on morphological composition follow in a principled way. The second section presents different PAPPI, (Fong, 1991), implementations of the theory, and provides evidence to the effect that the positions of the Specifier and the Complement have an effect on tractability, as well as the separation of overt and covert processing. The last section considers consequences for the properties of the interfaces between grammar and the performance systems.

2 Strict Asymmetry Theory

Strict Asymmetry Theory (SAT) is a theory of the structural relations of the grammar. The central axiom of SAT, here in (1), limits the formal properties of these relations.

(1) **Strict Asymmetry Hypothesis:** Grammatical relations are asymmetrical.

According to (1), any structural relation, be it primitive or derived, is asymmetrical. We take asymmetry to be a unidirectional relation r, given the following definition: r is asymmetrical $=_{df}$ $(\forall x)((\forall y)(rxy \supset \neg ryx)$. Thus the primitive structural relations of 'precedence' and 'dominance'

are asymmetrical, as is the case for the other structural relations of the grammar, including 'complement of', 'specifier of' and 'adjunct of' relations, as well as asymmetrical c-command, (Chomsky, 1995), (1998), (1999).

SAT precludes structural relations to be symmetrical. Thus, there is no bare sisterhood relation or symmetrical c-command relation that plays a role in the grammar. Consequently, expressions that have been analyzed in terms of bare sisterhood, such as root compounds, (Scalise, 1994), are re-analyzed in terms of asymmetrical relations.[1] SAT correctly predicts that strict ordering holds between the members of root compounds. This does not follow from a bare sisterhood analysis, as 'sister of' is a symmetrical relation, it does not impose a strict ordering of elements.

In SAT, the configuration in (2), where the Specifier, the Head and the Complement may be null (lacking phonetic features but not semantic features) is the elementary representation for morphological elements, roots and affixes, as proposed in (Di Sciullo, 1995).

(2)

X''

Specifier X'

X Complement
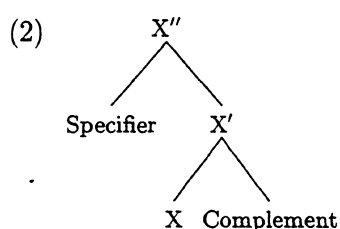
Motivation for (2) in morphology comes from the fact that a morphological expressions includes a categorial head, as argued for in (Williams, 1981) and (Di Sciullo and Williams, 1987). By definition, a head is in asymmetrical relation with a dependent category. Thus in (2) the head X is asymmetrically related to the Complement as well to the Specifier.

Motivation for (2) in morphology also comes from the Linear Correspondence Axiom (LCA), (Kayne, 1994).[2] The LCA in turn motivates the Universal Base Hypothesis, according to which the order of the base constituents is universally as in (2), where the Specifier precedes the Head and the Complement follows. If the LCA extends under the word level, as suggested in(Kayne, 1994), (Keyser and Roeper, 1997), the Universal Base Hypothesis does also. Thus the linear order of the base constituents in morphological structures is as in (2).

Motivation for (2) in morphology also comes from the properties of argument structure. The Spec-X-Compl asymmetrical structure has been shown to best express the relations of arguments to predicates, (Hale and Keyser, 1993), (Di Sciullo, 1999), (Chomsky, 1998), (1999). As roots and affixes are predicates, the properties of their form are best expressed in terms of (2).

In SAT, the operations and the conditions of the grammar conspire to generate asymmetrical relations in the derivation of linguistic expressions.[3] The Argument Linking condition proposed in (Di Sciullo, 1995), here in (3), is a core condition in derivational morphology, as it restricts the composition of affixes and roots.

(3) **Argument Linking**: Every non-A must be A-LINKed. Every A may be A-LINKed.

The conditions of the grammar ensure that the operations, at each step apply locally. They also ensure that the interpretation at the PF (Phonetic Form) and LF (Logical Form) interfaces is

---

[1]The empirical predictions of SAT with respect to both morphological and syntactic expressions are beyond the scope of theories not based on strict asymmetry, such as Distributed Morphology, (Hale and Marantz, 1993) and A-morphous Morphology, (Anderson, 1992).

[2]The LCA relates the linear-order of terminal elements to the total asymmetrical c-command relations that hold between the immediate non-terminals.

[3]SAT includes two basic operations, a structure building operation and a linking operation. The first derives complex categories on the basis of more elementary ones; the second relates categories in derivations and representations.

done under locality. We define the local domain of an asymmetrical relation r as the smallest domain where r applies. The local domain for morphological structure building operation is the complement domain of an affixal head. The local domain for morphological linking is the X′ projection of an affixal head and its complement domain.

SAT thus consists of elementary asymmetrical relations and a limited set of operations and conditions that preserve the basic asymmetrical nature of linguistic expressions. We consider some predictions of the theory for the properties of morphological expressions in what follows.
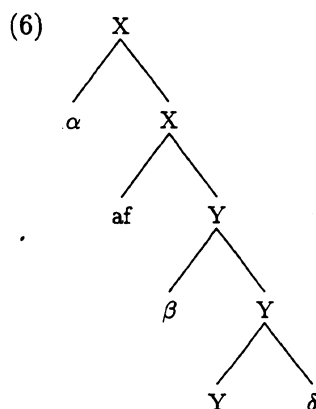
2.1 Predictions

We focus on the SAT analysis of zero and non-zero morphology in Indo-European languages, mainly English and in Romance (Italian and French) as well as in non Indo-European languages, mainly Mandarin Chinese. We provide evidence to show that SAT makes the right predictions with respect to overt and covert composition of affixes and roots. There is evidence to the effect that SAT makes more accurate predications than other theories of morphology with respect to the combinatorial restrictions imposed by affixes on roots. It also provides a rational as for why this is so.[4]

2.1.1 Indo-European

English has both zero N→V and non-zero derivation, as the examples in (4) illustrate. There is also reason to assume that a covert locational prefix is present in the structure of English verbs such as (5a), on a par with the overt spell-out of the prefix in Romance, as in (5b) and (5c).

(4) (a) to bottle the wine
     (b) to computerize the accounting department

(5) (a) (to) bottle
     (b) *im*bottigliare   (Italian)
     (c) *em*bouteiller   (French)

Strong empirical support for SAT in Indo-European morphology is the presence of systematic configurational restrictions on the composition of overt affixes and roots. We have shown in (Di Sciullo, 1995) that the Argument Linking relation restricts the composition of derivational affixes and roots. Given a configurational representation of argument structure, it is possible to limit the over-generation due to c-selection for derivational affixes. In our view, an affix projects and selects an asymmetrical structure. Affixes differ with respect to the specific argument structure requirement they impose on the configuration they combine with. This is summarized in (6) for a subset of English affixes, where α, β, δ are argument (A) positions; α, β are external argument positions and δ is an internal argument position.

(6)

```
        X
       /\
      /  \
     α    X
         /\
        /  \
       af   Y
           /\
          /  \
         β    Y
             /\
            /  \
           Y    δ
```

---

[4]To this extent its explanatory capacity extends way behond morphological theories including c-selection for derivational affixes: *-er*: [V], *-ize*: [A], *-able*:[V_] , (Grimshaw, 1990), (Lieber, 1992) and (Anderson, 1992).

(a) $((\beta))$: *-er, -en*
(b) $((\delta))$: *-ize, -ify*
(c) $((\beta,\delta))$: *-able, -ee, -ive*

The examples in (7) illustrate the facts. In English, the nominal affix *-er* may compose with an unergative verb (7a), that is a verb with an external argument and no internal argument, or a transitive verb (7d), but not with an ergative verb (7b), that is a verb with an internal argument and no external argument, or an obligatory ditransitive verb (7c). The verbal affix *-ize*, may compose with an ergative (8a) or an unergative adjective (8b), but not with a transitive adjective, (8c). The adjectival affix *-able* may only compose with a transitive verb (9d).

(7) (a) Mary is a swimmer
   (b)* John is a departer
   (c)* Lucy is a putter
   (d) Luc is an eater

(8) (a) They formalized the ideas
   (b) They certified the passport
   (c)* They sameized the results

(9) (a)* This is a sitable chair
   (b)* This vase is fallable
   (c)* This dress is putable
   (d) This book is readable

These facts indicate that morphological composition combine asymmetrical structures and not categories. The fact that asymmetrical relations override c-selection in derivational morphology is expected in our theory, as asymmetrical relations are basic in grammar.

Strong empirical support for SAT also come from prefixation. We argued in (Di Sciullo, 1997) that internal prefixes, directional and locational, contribute to the argument structure and the aksionsart of verbal projections they are a part of. Internal prefixes (P) head the lower projection of predicate argument structure as they are the morphological spell-out of the resulting change of state or change of location relations. Internal prefixes are asymmetrically c-commanded by external prefixes, such as the iterative and the inverse prefixes. In favor of this analysis is that i) they necessarily precede directional or locational prefixes, and ii) they have scope over the whole event denoted by the verbal projection.

The data in (10) and (11) show that languages differ with respect to the morphological spell-out of internal prefixes.

(10) (a) abottonare, incipriare, imburrare (Italian)
    (b) boutonner, poudrer, beurrer (French)
    (c) to button, to powder, to butter

(11) (a) imbottigliare, inscatolare, incassare (Italian)
    (b) embouteiller, emboîter, encaissser (French)
    (c) to bottle, to box, to cash

While Italian and French differ with respect to the overtness of the internal directional affix (10a) vs. (10b), they do not with respect to the overtness of the locational prefix (11a) and (11b). On the other hand English shows no overt prefixation, (10c) and (11c). There are nevertheless reasons to assume the projection of a covert prefix structure in the English case on a par with

Romance. In favor of this analysis is that i) in some cases a locational prefix can be overt, as in *to encase* and *to enlarge*, and ii) that the English verbs are change of location verbs, as it is the case for the Romance verbs, as such they include a prepositional structure where the complement of the preposition is the resulting Location.

Strong empirical evidence for SAT also comes from the fact that languages where conversion is an option, as it is the case for English N->V, but not for Romance where a verbal suffix is required, the conversion holds under structural asymmetry, relating a functional affix, be it an infinitival complementizer or a tense suffix, to the verb.

(12) (a) butter / to butter / He butters the bread / He buttered the bread

   (b) beurre / beurrer / Il beurre le pain / Il a beurré le pain

   (c) burro / imburrare / (He/She) imburra il pane / (He/She) Ha imburrato il pane

This last piece of evidence shows that asymmetrical relations hold within morphological expressions, as English zero derived verbs and Romance non zero derived verbs are the expressions of inchoative as well as causative-inchoative relations. It also shows that asymmetrical relations holds for the set of expressions derived by the grammar, as different features, say N and V are supported by different asymmetrical relations.

2.1.2 Non-Indo-European

East Asian languages have both zero N→V and non-zero derivation, as the examples in (13)–(14) and (15) from Mandarin Chinese illustrate. There is also reason to assume that a covert locational prefix is present in the structure of Chinese verbs such as *xiànjin* '(to) cash', on a par with the overt spell-out of the prefix in Romance, as in (5b) and (5c) above.

The examples in (13) from (Packard, 2000, :35) illustrate the fact that words such as *bīng* and *di* can be used as a noun or as a verb.

(13) (a) Lushang jiézine bīng   (b) Qing   ba   nàping   piju   bīngshang
         road-top congeal-ASP ice   Please BA   that:CL   beer   ice-attain
         'Ice has formed on the road'   'Please ice that beer'

   (c) You wàng xià di   (d) Wo lian yi di   dou méi hé
         oil toward down drip   I   even on drop all not drink
         'The oil dripped down'   'I didn't even drink a drop'

According to (Chan and Tai, 1994, :3), Mandarin has four types of zero-derived denominal verbs. The denominal verbs are characterized by the following criteria. The source noun and the verb form have identical form, except in some cases from difference in tone and aspiration or vowel quality. The denominal verb must be monosyllabic, the souce noun may not be, it may be suffixed or serve as a head of a nominal compound. Some examples of homophonous pairs are:

(14) a. $bian^1$ / 'whip'   b. $bian^1$ / 'to whip'   Instrument verbs
     c. $chuan^4$ /'string'   d. $chuan^4$ / 'to string'   Goal verbs
     e. $jiao^4$ / 'cellar'   f. $jiao^4$ / 'to store in a cellar' Location verbs
     g. $dian^3$ / 'dot'   h. $dian^3$ / 'to dot'   Locatum verbs

The examples in (15) illustrate the fact that the nominalizing suffix *–zhě* combines with verbs, nouns and adjectives to derive agentive nominals.

(15) (a) dúzhě / read – agt. suf. / 'reader'

   (b) zuézhě / study – agt. suf. / 'scholar'

   (c) bǐzhě / pen – agt. suf. / 'author'

Given SAT we predict that *–zhě* cannot combine with ergative verbs such as *dào* 'to arrive' and *shuaidao* 'to fall', while it can combine with unergative verbs such as *laodòng* 'to work'. This prediction is borne out.

(16) *dàozhě 'arriver', *shuaidaozhě 'faller' vs. laodòngzhě 'worker'

Chinese also has a verbalizing suffix *–huà*, that combines with nouns and adjectives to yield causative verbs, as (17) illustrates. We correctly predict that *–huà* cannot combine with transitive nouns such as kinship nouns, and it cannot combine with unergative adjectives such as *jiao'ào* 'proud'. These predictions are also borne out, see (18).

(17) (a) diànhuà / electricity–v.suf. / '(to) electrify'
     (b) jixiehuà / machine–v.suf. / '(to) mechanize'
     (c) kāihuà / open–v.suf. / '(to) civilize'

(18) (a) *mèimèihuà / sister–v.suf. / '(to) sisterize'
     (b) *jiao'àohuà / proud–.v.suf. / '(to) proudize'

3   Three Parsing Models

Linguistic theory often provides enough freedom for a variety of different computational models to be tested. Whilst remaining faithful to the theory, from a computational perspective, we are also interested in improving the efficiency of parsing models. The contribution of particular elements of a theory to computational complexity can be determined through experimentation. In particular, we will discuss the effects of variation in Specifier-Head-Complement order, and the contribution of empty heads and prefixes to the complexity of morphology within the framework of bottom-up shift-reduce parsing.

3.1   Background

In this section we describe, in turn, the encoding of X′ structure, the lexical representation of morphemes and heads, and the implementation of the Linking constraint.

3.1.1   X′ Structure

The context-free X′ grammar shown in (19) encodes two-level X′ phrase structure:

(19) `rule XP -> [X1,YP] st max(XP), bar(X1), proj(X1,XP), max(YP).`
     `rule X1 -> [ZP,X]  st bar(X1), head(X), proj(X,X1), max(ZP).`
     `rule xp -> [].`

X, X1 and XP are logical variables ranging over category labels at the head, bar and maximal projection levels, respectively. Heads are grounded in the lexicon. Maximal projections YP and ZP are recursively realized as either X′ structures or as xp introduced by the empty category rule shown above. The relevant categories and the projection relation are defined below:

(20) `head(n). head(v). head(a). head(p).`
     `bar(n1). bar(v1). bar(a1). bar(p1).`
     `max(np). max(vp). max(ap). max(pp). max(xp).`

     `proj(n,n1).  proj(v,v1).  proj(a,a1).  proj(p,p1).`
     `proj(n1,np). proj(v1,vp). proj(a1,ap). proj(p1,pp).`

A LR(1)-based parser was adopted for analysis. LR(1)-parsing, (Knuth, 1965) is a well-known and highly-efficient method of shift-reduce (bottom-up) parsing that processes morphemes in a left-to-right manner using a single symbol of lookahead for local disambiguation.[5] The adopted algorithm relaxes the strict LR(1) requirement of zero conflicts by allowing shift/reduce and reduce/reduce conflicts to exist in the table. Conflict resolution is handled by backtracking in the underlying Prolog system (Fong, 1991).

3.1.2 The Lexicon

The lexicon uses a default feature system for lexical entries. The following declaration expresses the lexical defaults for nouns, verbs and adjectives:

(21) `default_features([n,v,a], [specR(f(a(-))),selR(f(a(-)))]).`

In (21), `specR` imposes restrictions on specifier positions, and `selR` on complement positions. `f(a(-))` indicates that the referenced position should have feature `a(-)`. By convention, $A/\overline{A}$-positions are encoded using `a(±)`. Hence by default, specifiers and complements are $\overline{A}$-positions (unless otherwise indicated).

Consider the nouns in (22). By default, *computer* has two $\overline{A}$-positions. *Form* and *father*, on the other hand, have one and two A-positions, respectively.

(22) `lex(computer, n, []).`
    `lex(form,     n, [selR(f(a(+)))]).`
    `lex(father,   n, [specR(f(a(+))),selR(f(a(+)))]).`

As discussed in (6), affixes impose constraints on their complement domain. For example, the nominal affix *-er*, e.g. as in *employer*, in (23) indicates that the specifier of its complement must be an A-position. Similarly, *-ee*, e.g. as in *employee*, restricts both the specifier and complement *within* its complement to be A-positions.

(23) `lex(er,   n, [link(spec),  selR(spec(f(a(+))))]).`
    `lex(ee,   n, [link(compl), selR([spec(f(a(+))),compl(f(a(+)))])]).`

Finally, the abstract causative morpheme, `caus` in (24), differs from inchoative `inc` in that it has an A-specifier as well as restricting the specifier of its complement to be an $\overline{A}$-position.

(24) `lex(caus, v, [specR(f(a(+))), selR(spec(f(a(-)))), caus]).`
    `lex(inc,  v, [link(compl),    selR(spec(f(a(-)))), inc]).`

The *-er*, *-ee* and `inc` morphemes have an additional feature `link(spec/compl)`, which specifies the target of $\overline{A}$-Specifier Linking in a complement domain. We discuss the role of this feature in the next section.

3.1.3 Linking

Following (3), affix A-bar-positions must be linked to an A-position in their complement domain (provided one exists). This is implemented by the universally-quantified (over tree structure) condition `linkR` shown in (25):

(25) `linkR in_all_configurations CF where`
       `linkConfig(CF,XP,Type,Dom)`
    `then findApos(YP,Type,Dom), coindex(XP,YP).`

---

[5]The small size of the grammar permits the adoption of full LR(1)-style parsing. The machines described have about 100 states.

```
linkConfig(CF,XP,Type,Dom) :-
     maxProj(CF),          CF has_feature link(Type),⁶
     XP specifier_of CF,  \+ XP has_feature a(+),
     Dom complement_of CF.
```

This definition looks for configurations meeting the requirements of `linkConfig`: i.e. CF must be a maximal projection with an A̅-specifier XP and a complement Dom. For all satisfying configurations, `findApos` (definition not shown) will extract a phrase YP occupying an A-position of the appropriate kind indicated by Type. Here, Type refers to a lexical feature `link` and will be either `compl`, as in the case of *-er* in (23), or `spec`, as in the case of *-ee* and `inc` in (23) and (24), respectively. The resulting linking relation between XP and YP is indicated via coindexation.

All parses assigned an X′ structure will be filtered using this rule. Any parse found with an unlinked A̅-specifier will be rejected.

3.2 Specifier-Head-Complement Asymmetry

In this section, we consider the computational consequences of varying the specifier-head linear order for the LR shift-reduce parsing framework, i.e. (26a) vs. (26b) below:[7]

(26)

(a) Specifier on the left   (b) Specifier on the right
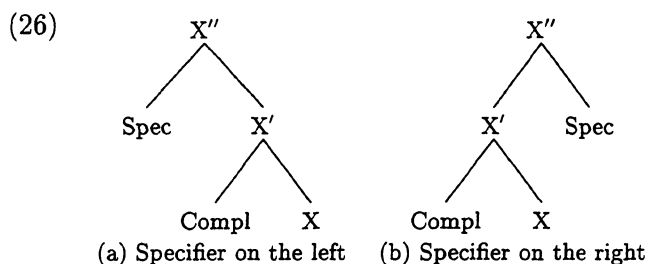
It turns out there is a considerable difference both in terms of the number of LR actions performed and the stack depth required to process an example like *formalize*, analyzed as *form-al-i(z)-e* in (27) below:

(27)

(a) *Formalize* with Left Specifier   (b) *Formalize* with Right Specifier

---
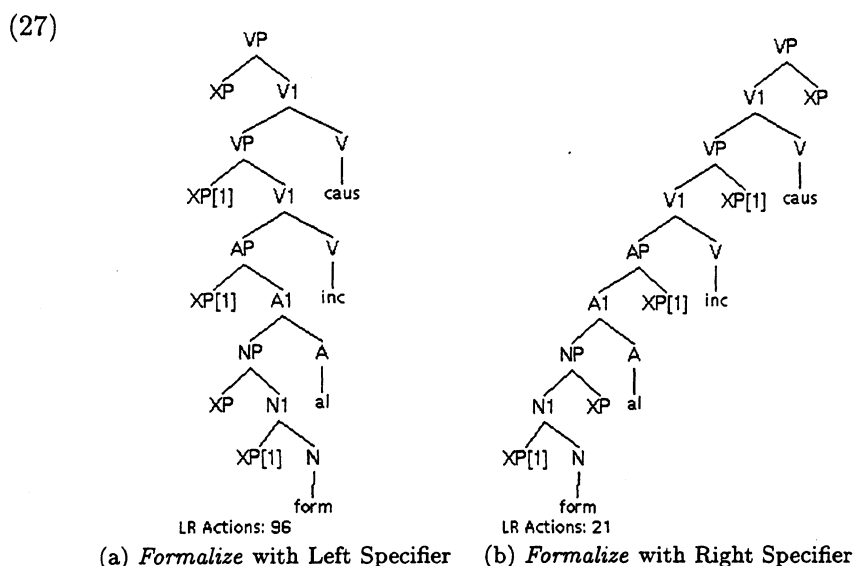
⁶Features of the head such as `link` are also available at the maximal projection level.

⁷Two other specifier-head-complement configurations can be obtained. However, we fix the head as being to the right of the complement for reasons of linearization. Note this also obviates the need for head-to-head movement.

The simple explanation is that the LR machine has to be able to predict an arbitrary number of empty argument positions before it can shift or "read" the first item, namely *form*, in (27a).[8] Contrast this with (27b), where specifiers are generated on the right only. Here, the LR machine only needs to generate a single empty argument position before a shift can take place. Hence only 21 actions and a stack depth of 2 is required in this case, compared to 96 and a stack depth of 5 in (27b). The following table compares left and right specifiers for a variety of examples:

(28)

| Word	Items	LR Actions	
		Left Specifier	Right Specifier
form	1	8	6
read-able	2	19	11
simpl(e)-i(f)-y	3	47	16
form-al-i(z)-e	4	96	21
form-al-i(z)-(e)-able	5	172	26

Finally, note that the right specifier model is *nearly* optimal in terms of the minimum number of actions required to analyze each word.[9] In general, the minimum number is given by the formula $4i + 2$, i being the number of items in the analysis of the word.[10]

3.3 Empty Heads and Prefixes

In this section, we describe the implementation and discuss the computational consequences of introducing empty heads and prefixes into the system.

3.3.1 Empty Heads

English has both zero and non-zero N → V conversion, see (4) above. *Computerize* is analyzed as the noun *computer* followed by the suffix *-ize*, which is, in turn, further decomposed into *-i(z)-e*, where the two constituents are merely instantiations of the (abstract) inchoative and causative morphemes, shown earlier in (24). As (29) indicates, we can analyze *bottle* along the same lines:

(29)

(a) *Bottle*   (b) *Computerize*
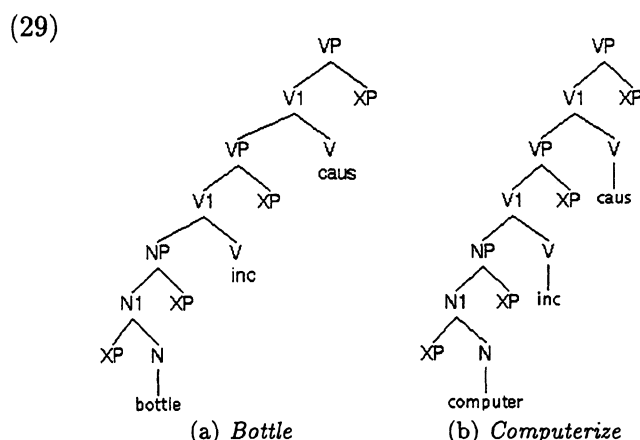
That is, the only difference with *bottle* as a verb is that the inchoative and causative morphemes are zero affixes. This is implemented by the following empty verb rule, defined to take on features from either abstract `caus` or `inc`:

---

[8]This is a form of infinite looping. This is controlled in the implementation by a stack limit on the number of consecutive empty categories.

[9]More precisely, the right specifier model "garden-paths" for exactly one action at a particular shift/reduce conflict point for every affix. A more detailed analysis is beyond the scope of this paper.

[10]To produce a two-bar-level structure, 3 reduce actions (for X′, X″ and the xp specifier) and 1 shift for the head are required. Hence, the total is 4 actions per phrase. The final accept and lowest complement xp reduction accounts for the fixed +2 part of the formula.

(30) `rule v with Fs -> [] st emptyVFs(Fs).`
     `emptyVFs(Fs) :- causative(Fs) ; inchoative(Fs).`

Unlike the case of empty specifiers or complements, empty heads permit infinite recursion. To limit this for inc and caus, we appeal to semantic considerations: that is, inc and caus occur at most once per singular event. This is enforced by a unique feature constraint, declared as follows:

(31) `top(max). % Dummy non-terminal`
     `unique_feature_domain max.`
     `    inc unique_feature_in max.  caus unique_feature_in max.`

The topmost (dummy) node returned by the LR machine, named max, is defined to be the domain for features inc and caus, which are declared as unique features, and therefore may be inserted at most once in max, thus guaranteeing parsing termination.[11]

3.3.2 Empty Prefixes

As mentioned earlier, see (5), there is reason to assume a covert directional prefix is present in the structure for *bottle* as a verb. In this case *bottle* represents the final location of an object that is the target of the event. The directional prefix, call it *en-*, will identify or supply an A-position that is necessary for the proper linking of the $\overline{\text{A}}$-specifier of inc. Similarly, *en-* is also required in the case of the verb *computerize*. The role of *en-* as a prefix is made explicit in the following parses:

(32)

(a) *Bottle*   (b) *Computerize*
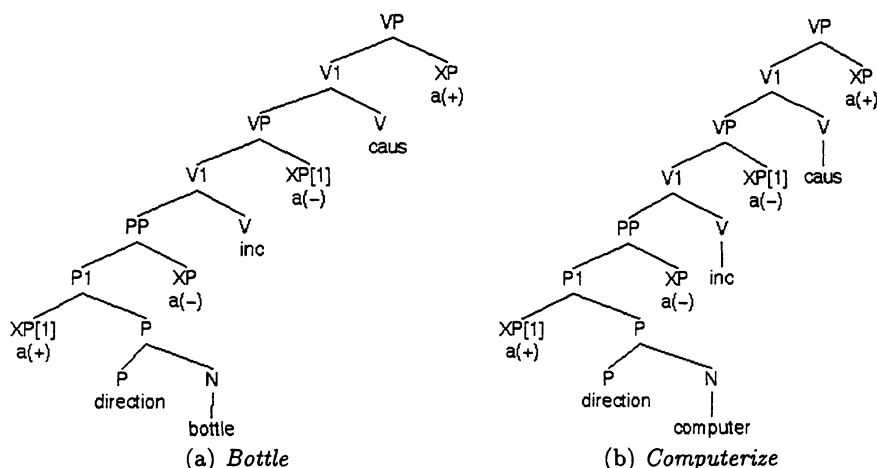
In (32a), *bottle* and *en-* are head adjoined together. This is implemented by adopting the following rules:

(33) `adjunction rule p -> [p,X] st head(X).`
     `rule p with Fs -> [] st en(Fs). % See lexical entry in (34)`

The empty preposition defined in (33) references the lexical entry for *en-* in (34):

(34) `lex(en, p, [selR(f(a(+))),direction,`
     `            hdAdj,adjR([f(selR(f(a(-)))),f(specR(f(a(-))))]),`
     `    direction unique_feature_in max.`

---

[11]PAPPI implements this efficiently by checking for duplicates immediately at abstract affix insertion time.

This basically states that *en-* is a directional preposition that selects for a complement A-position. Additionally, it must participate in a head adjunction structure (hdAdj) along with the restriction (adjR) that the adjoined element contains only A̅-positions. Finally, the feature direction is declared as unique for semantic and termination reasons, as in the case of inc and caus described earlier.

3.3.3 A Two-Stage Model

The expansion of the LR(1) parsing engine to account for empty heads and prefixes actually resulted in a decrease in machine size from 117 to 71 states as shown in (35):

(35)

Machine	States	No conflicts	Single conflict	Multiple conflicts
Standard LR	117	371 63%	214 37%	0 0%
Plus empty heads/prefixes	71	145 41%	64 18%	146 41%

However, it is important to realize that a smaller machine does not necessarily mean a more efficient parser. The distribution of LR action conflicts provides a strong indication that the revised machine is computationally much less tractable. For instance, the revised machine has a substantial proportion of multiple conflicts (40%), i.e. computational choice points with three (or more) possible LR actions. This is borne out empirically as (36) attests:

(36)

Example	LR Actions	
	Single Stage	Two Stage
form	53	7
bottle	80	7
readable	75	12
simplify	91	18
computerize	139	18
formalize	147	23
formalizable	157	27

Here we have tested two competing parsing engines: (A) a single stage LR machine, containing both overt and non-overt heads and prefixes, and (B) a two stage engine consisting of a simplified (overt only) LR machine, followed by a second, separate stage, responsible for adding back in any empty heads or prefixes.[12] By comparing the results of the two stage engine against those shown earlier in (28), it should be clear that dividing up the computational burden up into two separate modules is a strategy that permits us to handle zero morphology in an efficient manner.

4 Consequences and Conclusions

This work has consequences for the properties of the grammar, the properties of the parser, and their interface. The results of the implementation bring evidence to the effect that the grammar and the parser are different systems. The grammar optimally derives asymmetrical relations in the Spec-Head-Compl format. The parser optimally recovers asymmetrical relations where the basic relations are inverted. This cannot be otherwise as the grammar and the parser are two different systems. The grammar is a model of linguistic knowledge and the parser is

---

[12]The second stage actually consists of two smaller sub-modules, one responsible for inc and caus which runs in constant time, and the other for internal prefixes such as *en-*, running in time proportional to the depth of the parse.

a model of linguistic use. The systems may interface because of the asymmetrical properties of structural descriptions generated by the grammar. The implementation of SAT provided strong computational motivation for both specifier-right, complement-left interface pairing of X′ structure and for the separation of overt and non-overt affix heads into distinct modules.

Acknowledgements

References

Anderson, S. 1992. *A-Morphous Morphology*. Cambridge University Press.

Chan, M. K. M. and J. H-Y. Tai. 1994. From nouns to verbs: Verbalization in Chinese dialects and East Asian languages. In J. Camacho and L. Choueiri, editors, *Sixth North American Conference on Chinese Linguistics. NACCL-6*, volume II. USC, pages 49–74.

Chomsky, N.A. 1995. *The Minimalist Program*. MIT Press.

Chomsky, N.A. 1998. Minimalist inquiries. Ms. MIT.

Chomsky, N.A. 1999. Derivation by phrase. Ms. MIT.

Di Sciullo, A.-M. 1995. X′ selection. In J. Roorick and L. Zaring, editors, *Phrase Structure and the Lexicon*. Kluwer.

Di Sciullo, A.-M. 1997. Prefixed verbs and adjunct identification. In A.-M. Di Sciullo, editor, *Projections and Interface Conditions*. Oxford University Press.

Di Sciullo, A.-M. 1999. *The Local Asymmetry Connection*, volume 35. MITWPL.

Di Sciullo, A.-M. and E. Williams. 1987. *On the Definition of Word*. MIT Press.

Fong, S. 1991. *Computational Properties of Principle-Based Grammatical Theories*. Ph.D. thesis, Artificial Intelligence Laboratory, MIT.

Grimshaw, J. 1990. *Argument Structure*. MIT Press.

Hale, K. and S. J. Keyser. 1993. On argument structure and the lexical expression of syntactic relations. In K. Hale and S. J. Keyser, editors, *The View from Building 20*. MIT Press.

Hale, K. and A. Marantz. 1993. Distributed morphology and the pieces of inflection. In K. Hale and S. J. Keyser, editors, *The View from Building 20*. MIT Press.

Kayne, R. 1994. *The Antisymmetry of Syntax*. MIT Press.

Keyser, S.J. and T. Roeper. 1997. Anti-symmetry and leftward movement in morphology. Ms. MIT and UMass.

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

Lieber, R. 1992. *Deconstructing Morphology*. University of Chicago Press.

Packard, J. L. 2000. *The Morphology of Chinese*. Cambridge University Press.

Scalise, S. 1994. *Generative Morphology*. Foris.

Williams, E.S. 1980. Predication. *Linguistic Inquiry*, 11(1):203–38.

Williams, E.S. 1981. Argument structure and morphology. *Linguistic Review*, 1:81–114.