SIL Electronic Working Papers 1997-006, December 1997

A PC-PATR Implementation of GB Syntax

Cheryl A. Black

Contents:

- Abstract
- 1 Introduction
- 2 Phrase Structure and Subcategorization
  - 2.1 GB: X-Bar Theory
    - 2.1.1 Subcategorization
    - 2.1.2 X-Bar Phrase Structure Rules
    - 2.1.3 Sentences and Clauses in X-Bar Theory
  - 2.2 PC-PATR Implementation
    - 2.2.1 Constrained Phrase Structure Rules
    - 2.2.2 Linked List within [Subcat]
- 3 Question Formation
  - 3.1 GB: Movement with Traces
  - 3.2 PC-PATR Implementation
    - 3.2.1 Null Elements and the [Moved] Features
    - 3.2.2 [Down Subcat] and [Up Subcat] Features
- 4 Constructions Involving Movement to Subject Position
  - 4.1 GB: Semantics in the Syntax
  - 4.2 PC-PATR Implementation
    - 4.2.1 NP-A-trace and the [Moved: A] Feature
    - 4.2.2 Features, Templates and Disjunction
- 5 Case and Agreement
  - 5.1 GB: Case Filter and Specifier-Head Agreement
  - 5.2 PC-PATR: [Agr] Features and Constraints on NPs
- 6 Binding of Reflexives and Equi Constructions
  - 6.1 GB: Binding Theory
  - 6.2 PC-PATR: Agreement Features Passed Up in [Bind]
- 7 Conclusion
- Appendix A: Two Examples
  - 1. Who might Jorge seem to think Amy likes?
  - 2. Amy tries to like herself.
- Appendix B: Example Sentences the Grammar Accounts For
  - 1. Coordination, subcategorization and particle movement
  - 2. Questions
  - 3. Auxiliaries, case, agreement, and copular constructions
  - 4. Passives and unaccusatives
  - 5. Equi and Raising constructions
  - 6. Reflexives
  - 7. Adjuncts versus complements
- Endnotes
- References

Editor's note: This paper was originally presented at SIL's General CARLA Conference, 14-15 November 1996, Waxhaw, NC. CARLA, for Computer-Assisted Related Language Adaptation, is the application of machine translation techniques between languages that are so closely related to each other that a literal translation can produce a useful first draft.

Abstract

This paper documents the current status of a computational implementation of English syntax according to the Government and Binding Theory. The program used is PC-PATR (see McConnel 1995), based on the PATR formalism. Both the relevant GB theory and the details of the implementation are covered in the text. The appendices include sample trees and features and a list of example sentences.

1 Introduction

Some current syntactic theories, such as Lexical Functional Grammar (LFG) (Bresnan 1982), Generalized Phrase Structure Grammar (GPSG) (Gazdar, Klein, Pullum, and Sag 1985) and Head-Driven Phrase Structure Grammar (HPSG) (Pollard and Sag 1987, 1994), were developed with computer implementation in mind. They all employ feature unification as an essential ingredient in their theory. Government and Binding Theory (GB), on the other hand, was developed totally independently of any computational concerns. LFG, GPSG, and HPSG have all been modeled to at least some extent via the PATR formalism (Shieber 1986) upon which PC-PATR is founded. To my knowledge, no such implementation has ever been done for GB. This paper documents the current status of an implementation of most of GB syntax using the PC-PATR program (McConnel 1995).

The purpose of this experimental implementation is two-fold. Initially, it can be used as a teaching tool for both GB theory and for PC-PATR. The goal is ultimately to develop a full grammar for Zapotec (and later other languages) to be used as a syntactic parser and disambiguator for adaptation.

The major challenge that presented itself was how to implement a derivational or multistratal model within a single stratum system. Since the surface string is what is available to parse, S-structure is modeled by the grammar. GB's traces allow the underlying D-structure to be seen within the S-structure, and constraints between the two levels are captured via feature passing. In each sub-theory within GB, what part of the theory could be modeled computationally and how to do so within the current capabilities of PC-PATR had to be determined.

The paper is divided into five basic sections: Phrase Structure and Subcategorization, Question Formation, Constructions Involving Movement to Subject Position, Case and Agreement, and Binding of Reflexives and Equi Constructions. Within each section, the relevant GB theory is discussed first, then how it was implemented in PC-PATR is covered.[1] Since the English grammar is the most complete, it is the focus of the sections on the PC-PATR implementation, except where the work on another language provides additional insights. Two appendices show the trees and feature structures for two complex examples and a list of the sentences that the grammar accounts for so far.[2] These sentences include many ungrammatical examples that the grammar must rule out to be parsing correctly.

2 Phrase Structure and Subcategorization

Both GB and PC-PATR are phrase structure grammars. GB uses a particular form of phrase structure, called X-Bar Theory, that can be modeled in PC-PATR.

2.1 GB: X-Bar Theory

X-Bar Theory assumes that all phrases (including sentences and clauses) have the same structure, made up of heads and their projections. The general rules for this phrase structure are covered in section 2.1.2, after the importance of subcategorization is discussed.

2.1.1 Subcategorization

A word, such as a noun, verb, adjective or preposition is a lexical category. In structural terms, they are called **heads**. A particular head is choosy about what can combine with it to form a phrase.[3]

VP examples:[4]

(1) died / *died the corpse / *died to Sue about politics
(2) relied on Max / *relied / *relied from Max / *relied to Max
(3) dismembered the corpse / *dismembered
(4) talked (to Sue) (about politics) / *talked that the economy is poor
(5) read (the book) (to John) / read that the economy is poor

A **complement** is a phrase that a head takes or selects or subcategorizes for. Which complements a particular verb takes is an arbitrary property of that verb. Adjectives, nouns, and prepositions also subcategorize for their complements.

AP examples:

(6) red / *red that Sylvia would win
(7) afraid (of snakes) / *afraid to this issue
(8) ambivalent ((to Joe) about her feelings)
(9) certain (that Sylvia would win)

NP examples:

(10) individual
(11) book (about photography) / *book to Fred
(12) dislike of Fred
(13) ambivalence (to John) about my feelings
(14) rumor that all is well

PP examples:

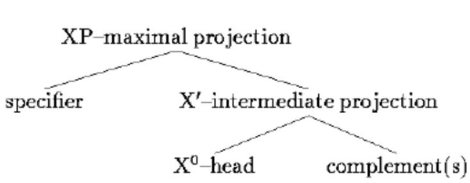
(15) about [the talk $]_{NP}$
(16) before [we leave $]_{S}$
(17) from [over the hill $]_{PP}$

Heads and complements are not the only parts of phrases. For example, NPs can be preceded by words (or sometimes phrases) like: *the, no, some, every, John's, my mother's*. APs can be preceded by degree words such as: *very, extremely, rather, quite*. PPs can be preceded by *just, right*. These items, called **specifiers**, differ from complements in their position in the phrase and in the fact that they are not subcategorized for by a particular head.

2.1.2 X-Bar Phrase Structure Rules

GB seeks to capture the similarities between the different categories of lexical phrases by assigning the same structure to them.

(18) Basic X-Bar Structure



The structure in (18) is generated by the two general rules in (19).[5]

(19) Phrase Structure Rules:

XP ---> Specifier X'

X' ---> X^0 Complements (=YP*)

Subcategorization is built into X-Bar Theory by the claim that a head (=X^0) subcategorizes for all and only its sisters. The complements are always phrases. In addition, heads and their maximal projections share some features, allowing heads to subcategorize for the heads of their sisters (e.g. *rely*). Heads and their specifiers may also share some features.

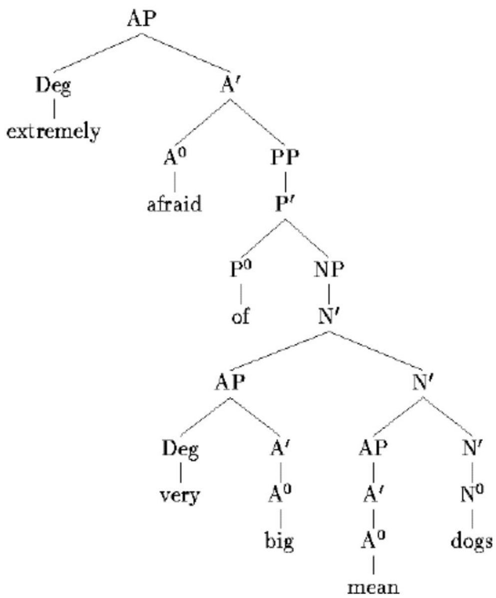
The only other phrase structure rules allowed are an unordered, recursive adjunction rule (20) (used for adjective phrases and relative clauses modifying nouns, and adverbial phrases) and a general conjunction rule (21):

(20) Z ---> Y Z

(21) Z ---> Z Conj Z

A sample X-Bar tree is given in (22), where *afraid* is the head, *extremely* is the specifier, and *of very big mean dogs* is the complement in the AP. The APs *very big* and *mean* modifying the noun *dogs* illustrate the recursive adjunction rule.

(22)



2.1.3 Sentences and Clauses in X-Bar Theory

The traditional rules in (23) do not fit the X-Bar schemata in (19).

(23) i. $\bar{S}$ ---> COMP S

ii. S ---> NP VP

Based upon data like (24)-(26) Chomsky (1986) argues that the complementizer is the head of the clause.

(24) a. Everyone insisted that the store would close on Thursdays.

b. *Everyone insisted for the store to close on Thursdays.

c. *Everyone insisted whether the store would close on Thursdays.

(25) a. They managed for their children to be happy.

b. *They managed that their children would be happy.

c. *They managed whether their children would be happy.

(26) a. Sue wondered whether the smoke would clear before daylight.

b. *Sue wondered that the smoke would clear before daylight.

c. *Sue wondered for the smoke to clear before daylight.

Each of the main verbs not only takes a clausal complement, but chooses which complementizer the clause must have. This is reminiscent of a verb like *rely* which subcategorizes for a PP complement which must have *on* as the preposition. Only heads can subcategorize, and only complements or the head of a complement may be subcategorized for. Therefore, the complementizer must be the head of the complement clause, which is a complementizer phrase or CP in X-Bar terms.

To reanalyze the structure of the sentence in X-Bar terms, its head, complement, and specifier must be determined. Neither of the constituents on the right side of the rule (23ii) can be the head of a phrase because they are phrases themselves, not lexical items or words. Evidence of subcategorization is again present in data like (27)-(29).

(27) a. Everyone insisted that the store would close on Thursdays.

b. Everyone insisted that the store was closed last Thursday.

c. *Everyone insisted that the store to close on Thursdays.

(28) a. They managed for their children to be happy.

b. *They managed for their children were happy.

c. *They managed for their children would be happy.

(29) a. Sue wondered whether the smoke would clear before daylight.

b. Sue wondered whether the smoke cleared before daylight.

c. *Sue wondered whether the smoke to clear before daylight.

When the complementizer is either *that* or *whether*, the sentence that follows is a regular finite sentence. In contrast, when the complementizer is *for* (28), *to* must be present followed by the bare form of the verb. The complementizers *that* and *whether* subcategorize for a finite complement, whereas *for* requires a nonfinite complement.

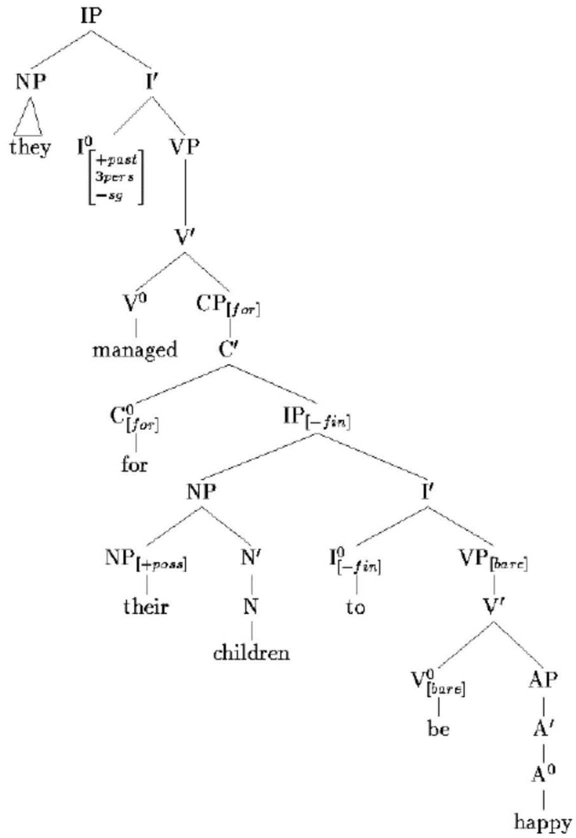
The marker for nonfinite clauses in English is *to*. Since the complementizer *for* subcategorizes for a nonfinite complement, *to* must be the head. Further evidence that *to* is a head comes from the fact that *to* subcategorizes for the bare form of the verb following it.

Chomsky (1986) posits that the tense and agreement features fill the same head position in finite sentences that *to* fills in nonfinite sentences. The category is therefore called Inflection, or Infl or I for short. This means a sentence is an IP.

$I_{[+fin]}$ is never filled by a lexical word at D-structure in English.[6] It always takes a VP as its complement, just as nonfinite *to* does. The subject NP is assumed to fill the specifier position in the IP.

The tree for (28a) is exemplified in (30).

(30)



In GB, both D-structure and S-structure must conform to the X-Bar phrase structure rules. The subcategorization requirements are specified to hold at D-structure.

2.2 PC-PATR Implementation

The PC-PATR implementation models the general rules of X-Bar Theory with constrained phrase structure rules. Verbal subcategorization requirements are implemented using a linked list within the subcat feature, as shown in section 2.2.2.

2.2.1 Constrained Phrase Structure Rules

PC-PATR provides the capability to write general phrase structure rules which can be constrained via unification constraints. Nothing within PC-PATR limits the rules to X-Bar Theory; the person writing the rules must discipline himself to stay within the theory they are modeling.

The phrase structure rules in the English model of GB in PC-PATR are all specific instances of the X-Bar rules for a particular category. Each rule is constrained to assure that the rule is only used for the correct lexical items and to pass certain features between mother and daughter nodes by requiring that they unify. For example, the rule for the subtree including the nonfinite marker *to* is given in (31).

(31)
```
rule{for nonfinite to}
        I' = I VP
          <I' subcat>     = <VP subcat>
          <I' moved>      = <VP moved>
          <I' bind>       = <VP bind>
          <I' type>       = <I type>
          <I' type fin>   = -
          <I' type q>     = -
          <VP agr vform>  = bare
          <VP agr person> = irrelevant
```

The rule says that the intermediate projection I' splits into the head I and its complement VP. The first four unification constraints require feature sharing between the mother I' and a particular one of the daughters. The `[subcat]`, `[moved]`, and `[bind]` features (to be explained later) on the VP are passed up to the I' and must unify with any value for those features present on I' due to another rule. The `[type]` features on I' must unify with those on I. From the next two constraints, we see that `[type]` must include both `[fin:-]` and `[q:-]`. The last two constraints restrict the `[vform]` and `[person]` features on the VP appropriately for a nonfinite clause.

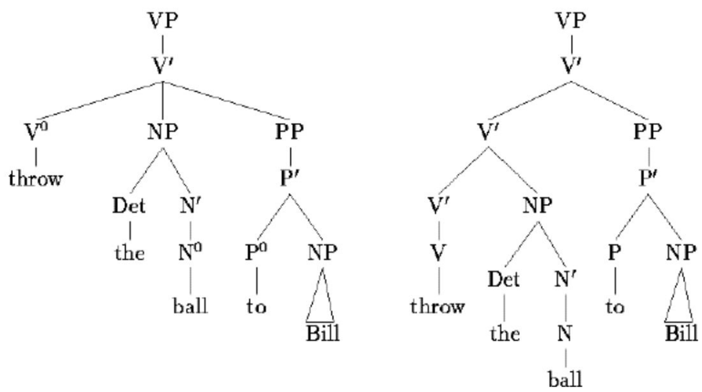
The only PC-PATR phrase structure rules that do not directly fit the X-Bar rules are the rules which introduce subcategorized complements for verbs. Rather than having all the complements be sisters of V^0 under V', the complements are each sisters of separate instances of V', in a stacked structure generated by rules such as:[7]

(32)
```
a. V' = V'_2 NP
b. V' = V'_2 PP
c. V' = V
```

This structure allows the subcategorization requirements to be represented in the form of a linked list (introduced by Shieber 1986:29), as discussed in the next section. It also has the benefit that fewer phrase structure rules are needed than if each different ordering of complements required a separate rule. For example, (32a) is called twice for a verb taking two NPs; (32a) then (32b) is used for a verb taking an NP and a PP; and two instances of (32b) are needed for a verb subcategorizing for two PPs. Subcategorized complements are still distinguished from adjuncts in the PC-PATR implementation, because the adjuncts are adjoined to VP and must not be subcategorized for.

To illustrate the difference, the GB tree for the VP *throw the ball to Bill* is given in (33a) and the PC-PATR stacked structure is shown in (33b).

(33) a. GB tree versus b. PC-PATR tree



2.2.2 Linked List within [Subcat]

The subcategorization information for a particular verb is encoded in the `\f` line in the lexicon with an abbreviation for a template. This template is then expanded in the grammar file into a feature structure. The part of the feature structure relating to subcategorization is under the feature `[subcat]`, which is broken down into the subfeatures `[subj]` and `[comp]` for 'subject' and 'complements', respectively.[8] The information within `[subj]` and `[comp]` has the form of a linked list. This allows the complements to be matched one at a time with the proper category in the phrase structure rule as the list is stepped through.

For example, the subcategorization information for a ditransitive verb like *throw* would be encoded as shown in (34).

(34)
```
[ subcat: [ subj: [ first: [ cat : NP ]
                    rest : none ]
            comp: [ first: [ cat : NP ]
                    rest : [ first: [ cat : PP ]
                             rest : none ]   ] ] ]
```

The three phrase structure rules from (32) are repeated in (35) with the unification constraints relevant to the subcat features added to each one.

(35) a.
```
V' = V'_2 NP
    <V'_2 subcat comp first> = <NP>
    <V' subcat comp>         = <V'_2 subcat comp rest>
    <V' subcat subj>         = <V'_2 subcat subj>
```

b.
```
V' = V'_2 PP
    <V'_2 subcat comp first> = <PP>
    <V' subcat comp>         = <V'_2 subcat comp rest>
    <V' subcat subj>         = <V'_2 subcat subj>
```

c.
```
V' = V
    <V' subcat>              = <V subcat>
```

Given the subcat feature structure in (34) and the constrained rules in (35), the PC-PATR program generates the tree (33b) from the bottom up. The verb *throw* is marked in the lexicon as ditransitive, so the feature structure in (34) is assigned to it. The verb is placed into the tree by rule (35c). The unification constraint on this rule passes the subcat information to the mother node V'. The next two words *the ball* would be separately parsed into an NP. The sequence `V' NP` matches the right side of rule (35a). The first unification constraint requires that the features in `<V'_2 subcat comp first>`, which are `[cat:NP]`, unify with the features on the NP, as they do. The second unification constraint then serves to step the list down one, passing only the remaining part of the complement subcat information up to the mother V'. The third unification constraint percolates the subject subcat information. Thus, the subcat feature structure for the new V' is:

(36)
```
[ subcat: [ subj: [ first: [ cat : NP ]
                    rest: none ]
            comp: [ first: [ cat : PP ]
                    rest: none ]    ] ]
```

The process just discussed for the NP complement is repeated for the PP complement with rule (35b). After stepping down again through the complement subcat information, the feature structure passed to the new mother V' allows no more complements, as shown in (37).

(37)
```
[ subcat: [ subj: [ first: [ cat : NP ]
                    rest: none ]
            comp: none       ] ]
```

A similar process restricts the subject subcat information at the appropriate point in the tree.

At present, only verbal subcategorization is implemented.

3 Question Formation

Not all surface sentences are the same as their underlying form: Yes/No questions and content questions are good examples. GB proposes a movement account which is restricted by certain principles and leaves behind a coindexed trace. The S-structure after movement is modeled in PC-PATR using null elements for traces and feature passing to match the subcategorization requirements appropriately.

3.1 GB: Movement with Traces

The relationship between a declarative sentence, such as (38a) and a Yes/No question, such as (38b), is taken to be a clear case of **head movement** in GB. (38c-f) show that more is going on in Yes/No questions than simply putting a verb in front of the subject.

(38) a. Sally has declined the job.

b. Has Sally declined the job?

c. *Has Sally might decline the job?

d. *Might have Sally declined the job?

e. Might Sally have declined the job?
f. *Declined Sally the job?

Sentences like (39) show that auxiliaries come in a certain order and they subcategorize for the form of the verb that comes after them.

(39) Jill could have been playing the piano.

In Yes/No questions, this order and form restriction still holds. Then, the highest auxiliary moves in front of the subject. Two other phenomena show that the highest auxiliary is important also: VP deletion and the placement of negation. In (40) negation follows the first auxiliary, and in VP deletion constructions, at least one auxiliary must remain behind. A coherent account of these two facts yields that the highest auxiliary must move to I^0.

(40) Jill couldn't have been playing the piano but Bill could (have (been)).

Yes/No questions also move the highest auxiliary, but movement to I^0 does not change the word order. Something further is needed for questions.

GB places two strong restrictions on movement. First, the only possible types of movement are structure preserving or adjunction. Structure preserving movement means that the moved element must correctly fit into the tree structure which existed at D-structure. Heads can only move to head positions and phrases can only move to specifier positions, since a complement that was not subcategorized for cannot be added. Adjunction is allowed so the auxiliary could possibly be adjoined to the IP, but the adjunction rule is recursive, which incorrectly predicts that more than one auxiliary may be fronted (38d). The second restriction is the principle of 'No Loss of Information,' meaning movement is not allowed to a position which is already lexically filled. Thus, movement of the highest auxiliary to $I^0_{[+fin]}$ is possible since the position is not filled by a word at D-structure. But where does it go from there to form a question?

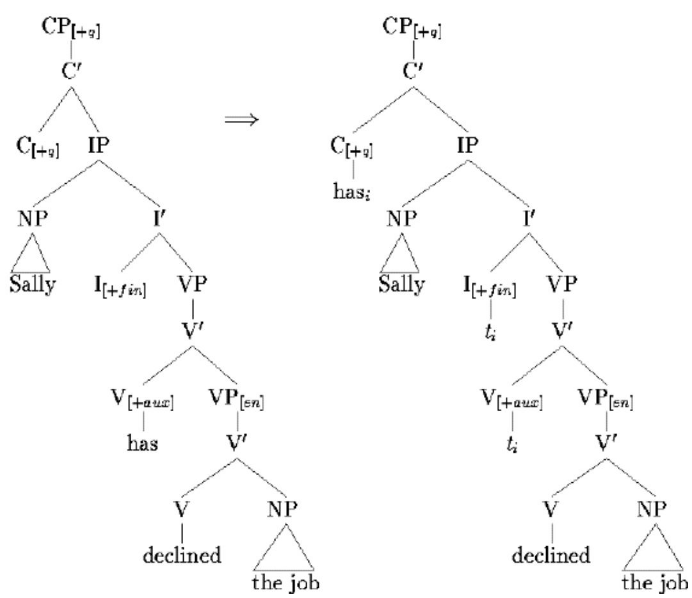
The solution to this puzzle comes from embedded Yes/No questions such as (41).

(41) I wonder whether Sally has declined the job.

Embedded clauses are CPs, so main clauses are also assumed to be CPs. Clauses containing a question have the feature [+q].

The D-structure for (38b), shown in (42a), shows that both the $I_{[+fin]}$ and the $C_{[+q]}$ heads are available for movement. (42b) shows the S-structure, with *has* filling the $C_{[+q]}$ position and coindexed traces in both the $I_{[+fin]}$ and $V_{[+aux]}$ positions to show where it moved from.

(42) a. D-structure b. S-structure



The S-structure in (42b) meets the conditions on movement. Further, this movement provides a natural account for the fact that no inversion occurs in embedded Yes/No questions (41). In the embedded clause, *whether* fills the $C_{[+q]}$ position, so the principle of No Loss of Information blocks the movement. A separate condition is not necessary.

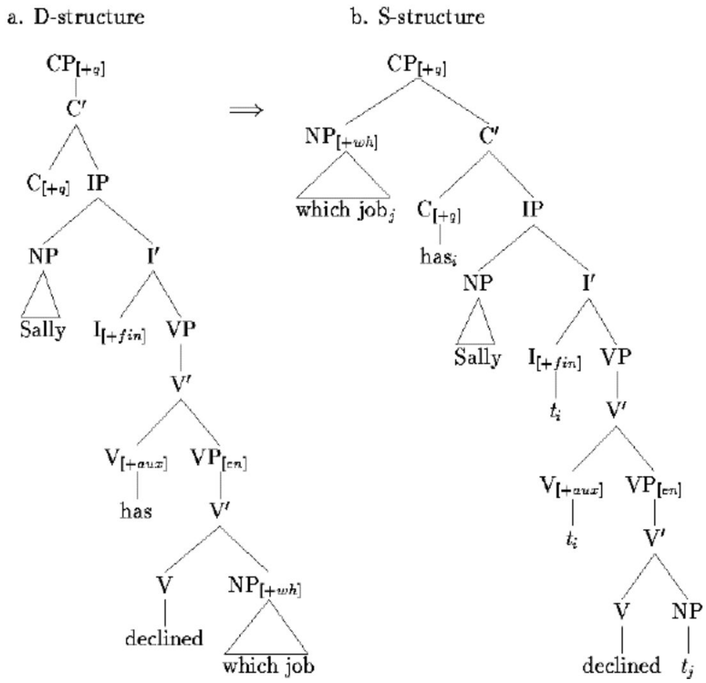
Content questions provide examples of $\bar{A}$-movement or movement to a non-argument position. Argument positions are complements or the subject, so a non-argument position is a specifier position other than the subject or an adjoined position.

The question in (43) is generated similarly to the Yes/No question above, with an additional movement added.

(43) Which job has Sally declined?

The D-structure in (44a) is almost identical to (42a), except that *which job* is in the complement position instead of *the job* and that NP is marked with the [+wh] feature. The S-structure is generated by moving the $NP_{[+wh]}$ to the specifier of $C_{[+q]}$, along with the movement of the auxiliary to $I_{[+fin]}$ and then to $C_{[+q]}$.

(44)



The coindexed traces left after movement form a chain between the D-structure position and the S-structure position of the lexical item. This chain allows D-structure information such as subcategorization to be retained at S-structure. It also accounts for proper assignment of semantic roles and case, as will be discussed in sections 4.1 and 5.1.

3.2 PC-PATR Implementation

The PC-PATR grammar parses the S-structure tree for each sentence. Since questions involve movement of the auxiliary and a *wh*-phrase, null elements are introduced for traces of the movements. The challenge of matching the subcategorization information with the moved elements is handled by passing features up or down through the tree.

3.2.1 Null Elements and the [Moved] Features

PC-PATR allows null elements, but they must be constrained carefully so that they only show up in the proper places. Therefore, various types of null elements are used to model the GB traces. V-trace is used for the trace of the auxiliary, NP-trace is used for the trace of a questioned NP, and PP-trace is the trace of a questioned PP. The phrase structure rules which define the null elements are given in (45a-c) and the rules which allow them to be placed into the tree are shown in (45d-f).[9]

```
(45) a. V-trace  =
     b. NP-trace =
     c. PP-trace =
     d. I'       = V-trace VP
     e. V'       = V'_2 NP-trace
     f. V'       = V'_2 PP-trace
```

In content questions, the trace in the D-structure position must be the correct category as required by the subcategorization frame of the verb, and this must also match the fronted element. The fact that an NP-trace must only be used for verbs which subcategorize for an NP can be assured by the unification constraint in (46a). Correctly matching the subcategorization information with the fronted element is more complex. The constraints (46b-e) assure that the appropriate subcat information is placed into the `[moved:A-bar]`[10] list; that the subcat list is stepped one position (as illustrated in section 2.2.2); and that the other information on moved elements, as well as the subcategorization restrictions, is passed up.

```
(46) V' = V'_2 NP-trace
       a.   <V'_2 subcat comp first cat> = <NP>
       b.   <V' moved A-bar first>       = <V'_2 subcat comp first>
       c.   <V' moved A-bar rest>        = <V'_2 moved A-bar>
       d.   <V' subcat comp>             = <V'_2 subcat comp rest>
       e.   <V' subcat subj>             = <V'_2 subcat subj>
```

To assure that the subcategorized complement matches the fronted element, the contents of `[moved]` is passed all the way to the top. The rule and constraints which assure the matching for a questioned NP are given in (47). (A similar rule accounts for a questioned PP.) (47a) requires that the NP be `[type[wh:+]]`. (47b) requires that the features on the NP unify with the feature structure of the first element in the `[moved:A-bar]` list. Then (47c) steps the list for the CP, whose `[moved]` feature is required to be `none` for all types of movement at the root level.

```
(47) CP = NP C'
       a.   <NP type wh>           = +
       b.   <C' moved A-bar first> = <NP>
       c.   <C' moved A-bar rest>  = <CP moved A-bar>
```

3.2.2 [Down Subcat] and [Up Subcat] Features

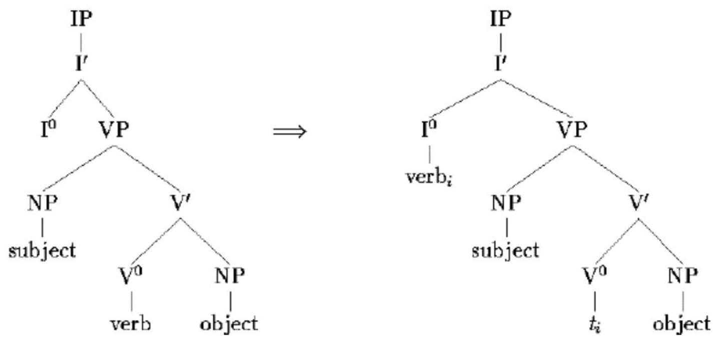
Head movement of the auxiliary to C^0 provides an additional challenge. Auxiliaries come in a certain order and each one requires that the verb following it be in a certain form. This is accounted for by subcategorization for a particular form of VP. In a question, however, the first auxiliary is in C^0 before the subject, not in I^0 or V^0 where local subcategorization can take place. The V-trace is in the crucial position, but the `[subcat]` information is up with the auxiliary. This `[subcat]` information must be passed down to the position of the V-trace.

For English it turns out to be a simple matter of placing constraints on the appropriate phrase structure rules to assure that the `[subcat]` information gets passed down to the rule containing the V-trace, so that the complements can be checked normally. The phrase structure rules and constraints that accomplish this are shown in (48). Also shown are the constraints which assure that the V-trace matches the moved V.[11]

```
(48) a. C' = V IP
       <V subcat comp>    = <IP subcat comp>
       <IP moved head>    = <V>
```

While verb movement is quite restricted in English, in some other languages it occurs in almost every sentence. This is true for VSO languages like Quiegolani Zapotec (QZ). The analysis is that the D-structure is SVO order with the subject in the specifier of VP rather than the specifier of IP position (49a) (Black 1994, following Koopman & Sportiche 1991, McCloskey 1991). The verb then moves to I^0 at S-structure to obtain VSO order (49b).[12]

(49) a. D-structure b. S-structure



The desire to maintain the stacked VP structure and the list notation for `[subcat]` discussed in section 2.2.2 for QZ introduced an additional complication. Recall that the list is stepped through as the complements are matched going up the tree from the verb. Therefore, the `[subcat]` information needs to be moved down the tree from I^0 to the position of the V-trace also. The features `[down]` and `[up]` were introduced to hold the relevant `[subcat]` information. The information being passed down from the verb in I^0 is in `[down subcat]`, while the linked list of complements is being stepped through in `[up subcat]`. The two feature structures in `[up subcat]` and `[down subcat]` are required to unify in the rule introducing V-trace. Sample rules from the QZ grammar with the relevant constraints are shown in (50).

(50) a.
```
rule{normal position of V}
I' = V VP
  <VP moved head>             = <V>
  <V subcat>                  = <VP down subcat>
```

b.
```
rule{subject NP}
VP = NP V'
  <VP moved>                  = <V' moved>
  <VP down subcat subj first> = <NP>
  <V' down subcat subj>       = <VP down subcat subj rest>
  <VP down subcat comp>       = <V'down subcat comp>
  <V' up  subcat comp>        = none
```

c.
```
rule{NP complement}
V' = V'_2 NP
  <V' moved>                  = <V'_2 moved>
  <V' down subcat comp>       = <V'_2 down subcat comp>
  <V'_2 up subcat comp first> = <NP>
  <V' up subcat comp>         = <V'_2 up subcat comp rest>
```

d.
```
rule{trace of fronted verb}
V' = V-trace
  <V' moved head>             = [cat: V]
  <V' down subcat>            = <V' up subcat>
```

4 Constructions Involving Movement to Subject Position

Constructions such as passives, unaccusatives, and raising require movement to the subject position. The constraints on movement discussed in the last section still hold. In addition, semantic roles and the lack of subcategorization for a subject are needed.

4.1 GB: Semantics in the Syntax

Passive constructions are the most well-known constructions involving movement to the subject position. The old Transformational Grammar analysis begins with a transitive deep structure, then creates a passive surface structure by moving the subject to the *by*-phrase (or omitting it completely), moving the object to the subject position, adding the passive *be* and changing the verb form appropriately. This movement analysis of passives captured the generalizations that:

(51) a. Most transitive verbs have passive alternants.

b. No intransitive verbs have passive alternants.

c. The 'subject' of a passive verb corresponds to the object of its transitive alternant.

GB does not account for passives in exactly the same way as Transformational Grammar did, since movement of the object to a previously occupied subject position would violate the principle of No Loss of Information. The generalizations in (51) must still be accounted for, and GB goes a step further than (51c) to account for the synonymy between the active and passive sentences, as seen in (52).

(52) a. The kids invited Sue to the party.

b. Sue was invited to the party by the kids.

General semantic roles, such as Agent, Theme, Recipient, Goal, Locative, etc., can be linked to arguments within the lexical entries to capture this synonymy. For example, the lexical entry for *invite* would be:[13]

```
invite, V,     [ _ NP (PP[to]) ]
                   |    |
invite'    <AGT, THEME, GOAL>
```

where each syntactic complement (called internal arguments) must be linked one-to-one with a semantic role, and one additional role may be linked to the external argument (or subject). Lexical entries apply at D-structure, so not all verbs assign a role to the subject position.[14]

Further assumptions about lexical entries in GB include the desire that the related forms of a word share a single subcategorization frame and that there should not be any cross linking of syntactic arguments and semantic roles. For example - directly relevant to the analysis of Passive - the THEME should not be assigned to the object in one case and to the subject in a related entry. This assumption is formalized under the Uniformity of Theta Assignment Hypothesis (Baker 1988:46), allowing the broader claim that the THEME role should always be assigned to the direct object when it is assigned, since that is its position in normal transitive verbs; the RECIPIENT role is assigned to the indirect object, etc.

Though the semantic roles are assigned at D-structure through the lexical entries, further assumptions forbid movement from changing the linking between syntactic arguments and semantic roles. Therefore, movement out of most positions is allowed, as we have seen with *wh*-question formation, but nothing else can move into that position. The semantic role stays with the original position rather than moving with the phrase; the semantic role is not part of the tree but part of the lexical subcategorization that goes with the D-structure position. Movement into a position linked to a semantic role is not allowed, since it would cause the moved element to take on that semantic role, and thus alter the original linking. The coindexed trace provides the link between the moved element and the position it occupied at D-structure. Therefore, both semantic role and subcategorization requirements are still recoverable at S-structure.

Given these assumptions, the GB account of passive must be partly done in the lexicon and partly by movement. A lexical rule, such as that given in (53), is used to capture generalizations (51a-b) that passive verbs are related to transitive verbs. In order to account for alignment of the semantic roles between entries as much as possible and follow the Uniformity of Theta Assignment Hypothesis, the object NP remains in position at D-structure.

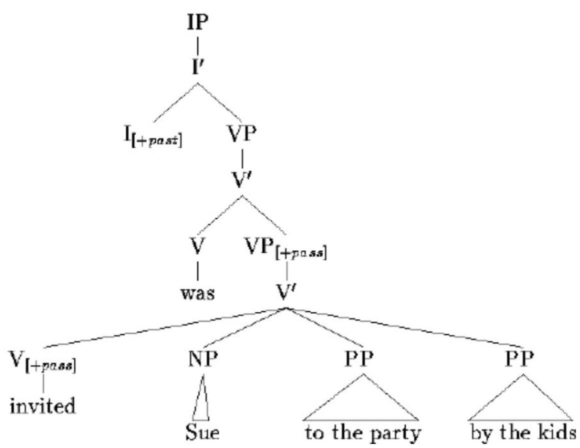
(53) V, [_ NP X] ---> V-en$_{[+pass]}$, [_ NP X (PP$_{[by]}$)]

The full lexical entry for *invited*, including semantic roles, is:

```
invited,    V[+pass]   [ _ NP  (PP[to])  (PP[by]) ]
                         |      |         |
invited'              <THEME,  GOAL,    AGENT>
```

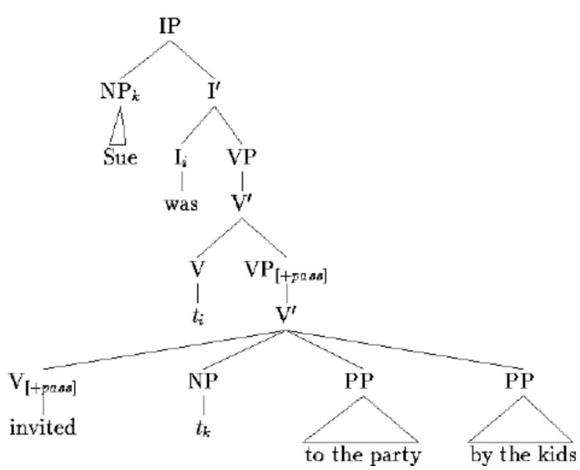
Note that no external argument is assigned by the passive verb. The D-structure for the passive sentence is as shown in (54):

(54) D-structure for passive



Movement of the object to subject position is still needed. This movement can take place since there is no semantic role linked to the subject position at D-structure and the position was not lexically filled, so the principle of No Loss of Information is not violated. A coindexed trace is left behind to maintain the linking of the object to its semantic role. So S-structure looks like (55):

(55) S-structure for passive



The assumptions made about semantic roles in the lexical entries also requires us to distinguish between types of intransitive verbs, following the Unaccusative Hypothesis (Perlmutter 1978). This means, for example, that while verbs like *sleep* would have an AGENT in subject position at D-structure, verbs like *die* would have an empty subject position at D-structure with a THEME object, as shown in the lexical entries. Movement of the THEME to subject position for the unaccusative verb *die* is exactly parallel to the passive movement.

```
sleep,  V   [ _ ]            die,  V [ _ NP ]
                                          |
sleep'   <AGT>               die'      <THEME>
```

The third type of A-movement (movement to an argument position) involves raising predicates, such as *seem* and *likely*. These predicates take either a finite or a nonfinite clause complement and do not assign a semantic role to their own subject position. The lack of a semantic role assigned to the subject position can be seen when there is a finite clause complement by the presence of the dummy *it* (56a). In the case of a nonfinite clause complement, the subject of the lower clause must raise to the main clause subject position.

(56) a. It is likely that Sue will come.

b. Sue is likely to come.

c. *It is likely Sue to come.

Case Theory (see section 5.1) provides the motivation for A-movement of the particular NP in passives, unaccusatives, and raising constructions.

4.2 PC-PATR Implementation

The implementation in PC-PATR accounts for the A-movement of the object in passives and unaccusatives or the lower clause subject in raising constructions to the main clause subject position in an analogous way to $\bar{A}$-movement. These details are briefly covered before moving to the issue of how the changes in the lexical entries are made.

4.2.1 NP-A-trace and the [Moved: A] Feature

As far as the phrase structure rules are concerned, A-movement is handled within PC-PATR in a completely parallel way to the other types of movement discussed in section 3.2.1. The trace of A-movement is distinguished from the $\bar{A}$-traces by being called `NP-A-trace`. The moved category is stored by the `[moved:A]` feature. The rule responsible for introducing the A-trace for both passives and unaccusatives is given in (57a), while the A-trace for raising is introduced by rule (57b).

(57) a.
```
rule{passives, unaccusatives}
  V' = V'_2 NP-A-trace
    <V'_2 subcat comp first cat> = NP
    <V'_2 subcat comp first>     = <V' moved A first>
    <V' moved A rest>            = none
    <V' subcat comp>             = <V'_2 subcat comp rest>
    <V' subcat subj>             = <V'_2 subcat subj>
    <V'_2 subcat subj>           = none
```

b.
```
rule{lower clause of raising}
  IP = NP-A-trace I'
    <I' subcat subj first>       = <IP moved A first>
    <IP moved A rest>            = none
    <IP subcat subj>             = <I' subcat subj rest>
```

At the top of the tree, one of the options for the subject position requires that no subject was subcategorized for by the predicate in that clause (58a) and that the `<moved A first>` path contains `[cat:NP]` (58b). This option accounts for the surface subject of all three types of A-movement.

(58)
```
rule{raised subject option}
    IP = NP I'
  a.   <I' subcat subj>    = none
  b.   <I' moved A first> = <NP>
  c.   <IP moved A>        = <I' moved A rest>
```

4.2.2 Features, Templates and Disjunction

The PC-PATR implementation differs from GB in that the syntactic form of the subject is subcategorized for (although it is always either an NP or `none`, which is consistent with GB). Passives, unaccusatives, raising verbs, and auxiliaries have `[subcat:[subj:none]]` within their feature structure and thus will have a subject only via A-movement.

Putting in the semantic roles seemed to just add the additional feature of `[role]` for each NP, slowing processing without appreciable gain. In order to use the semantic roles to rule out incorrect parses, we need to specify which semantic roles a given lexical item or phrase can fill and then unify this set of roles with the subcategorization requirements of the predicate. Which phrases can be adjuncts and which must be complements must also be restricted. These restrictions are not currently implemented; a negation mechanism allowing specification of what a certain atomic value could not be would be helpful.

The relationship between transitive and passive forms of a particular verb is handled via disjunctive templates.[15] For example, a verb like *broken* would have the feature description `+en trans/pass`, while the form *broke* is marked `+past trans/unacc`. The relevant templates are given in (59). These templates are then matched with the number of arguments present in the sentence, etc., to determine whether a correct parse can be given.

(59)
```
Let trans       be [subcat: [ comp: [npobj]
                              subj: [npsubj] ]]

Let unacc       be [type:   [ passive: -]
                    subcat: [ subj: none
                              comp:[npobj]]]

Let trans/unacc be { [trans] [unacc] }

Let trans/pass  be {[type:   [ passive: +]
                     subcat: [ comp: [first: [cat : NP
                                              type:[ pred: -]
                                              role:  Theme ]
                                      rest :{[first:[cat :PP
                                                     role: Agent
                                                     type:[animate:+]
                                                     head:[lex:by]]
                                              rest : none]
                                             none }]
                               subj: none ]]
                    [trans]}
```

As mentioned above, this same implementation of passives and unaccusatives would be possible without the addition of semantic roles at all for English. Semantic roles are more interesting and necessary in the implementation of a syntactic parser for Tatana' in Malaysia, where the case of each nominal is morphologically marked and the voice of the verb determines the required ordering and semantic role of the arguments.[16] The complex disjunctive template for Patient voice (one of three voices: Actor, Patient, Referent) is shown in (60). This template unifies with the `[trans]`, `[intrans]`, or `[ditrans]` templates on a particular verb to limit the options available. Note that the basic word order in Tatana' is VSO, but a nominative NP may be fronted before the verb.

(60)
```
Let PV    be [agr    : [voice : patient ]               subcat :{                        | Three options:                  |
                        |-------------------------------------------|
                        | Option 1: Clause has three NPs            |
                        |           First NP is genitive case       |
                             [first : [cat  : NP
                                       case : genitive
                                       role : ACTOR ]
                              rest  :{
                                | Two sub-options:                  |
                                |-----------------------------------|
                                | Normal word order: V GEN NOM DAT  |
                                     [first: [cat  : NP
                                              case : nominative
                                              role : PAT ]
                                      rest : [first: [cat  : NP
                                                      case : dative
                                                      role : REC ]
                                              rest : none ]]
                                |-----------------------------------|
                                | Fronted NP: NOM V GEN DAT         |
```

```
          |   note: also allows V GEN DAT NOM      |
                [first: [cat  : NP
                         case : dative
                         role : REC ]
                 rest : [first: [cat  : NP
                                 case : nominative
                                 role : PAT ]
                        rest : none ]] } ]
  |------------------------------------------------|
  | Option 2: Clause has two NPs                   |
  |           Either normal order: V GEN NOM       |
  |           or fronted NP:       NOM V GEN       |
       [first : [cat  : NP
                 case : genitive
                 role : ACTOR ]
        rest  : [first: [cat  : NP
                         case : nominative
                         role : PAT ]
                 rest : none ]]
  |------------------------------------------------|
  | Option 3: Clause has only one NP               |
  |           Either normal order: V NOM           |
  |           or fronted NP:       NOM V           |
       [first : [cat  : NP
                 case : nominative
                 role : PAT ]
        rest: none ] } ]
  |------------------------------------------------|
```

5 Case and Agreement

In the areas of both case and agreement, English is deficient compared to some other languages. For both GB and PC-PATR, the features are assumed to be present, whether or not there is distinct morphological marking.

5.1 GB: Case Filter and Specifier-Head Agreement

GB assumes that all NPs must have Case (abstract, not necessarily morphological case) at S-structure. This is expressed by the Case Filter (61). Nominative Case is assumed to be assigned to the subject position (the specifier) by $I_{[+fin]}$. Similarly, Genitive Case is assigned to the possessor position (the specifier) by the noun. Accusative Case is assigned by a verb to its NP complements, by a preposition to its complement, and by the complementizer *for* to the subject (the specifier) of the nonfinite clause which follows it.[17]

(61) Case Filter

*NP if it does not have Case at S-structure.

The Case Filter, coupled with Burzio's Generalization (Burzio 1986) that predicates that do not assign a semantic role to their external argument (i.e. do not subcategorize for a subject) cannot assign Case to their complements, provides the motivation for movement to the subject position for passives, unaccusatives, and raising constructions. These NPs cannot receive Case in their D-structure positions, so they must move to a position which assigns Case: the specifier of $I_{[+fin]}$.

Agreement is also seen to be linked to a particular position or to a relationship between positions in the tree structure. An $I_{[+fin]}$ agrees with its specifier (the subject), and it can also subcategorize for a VP with the proper agreement features (including person, number, and tense).

5.2 PC-PATR: [Agr] Features and Constraints on NPs

The case and agreement features of lexical items, such as `[person]`, `[number]`, `[gender]`, `[case]`, and `[vform]` (for verb form, including tense), are introduced as subfeatures under `[agr]` via templates. Structural case and structural agreement are then constrained via the phrase structure rules. For example, the constraint `<NP agr case>=nominative` on the rule which introduces subjects requires that the subject is in nominative case. Similarly, NP objects are required to have accusative case via a constraint on the rule `V' = V'_2 NP`. Agreement between the subject and the verb is enforced by percolating the `[agr]` features on the verb up to the I', and then requiring that `<I' agr>=<NP agr>`.

The implementation of case and agreement restrictions becomes more challenging in dealing with questions. In content questions, the fronted *wh*-phrase retains the case it would have had in its D-structure position. This is enforced at the point of the trace by placing the correct case information within the `[moved:A-bar]` feature matrix, so that this `[case]` feature will have to unify with the `[case]` of the fronted NP. The rules and relevant constraints for subjects and objects being questioned are shown in (62).

(62) a.
```
rule{subject being questioned}
IP = NP-trace I'
     <IP moved A-bar first>          = <I' subcat subj first>
     <IP moved A-bar first agr case> = nominative
```

b.
```
rule{for questioned object}
V' = V'_2 NP-trace
     <V'_2 subcat comp first>        = <V' moved A-bar first>
     <V' moved A-bar first agr case> = accusative
```

The second problem with questions is dealing with the agreement after subject-aux inversion. The verb agreement features are not known at the I' level, as they are in non-questions, since the verb that agrees with the subject is up in the C position. The `[agr]` features of the subject are therefore passed up to the IP for questions, so that agreement can be enforced in the rule `C' = V IP` via the constraint `<IP agr>=<V agr>`.

Binding constructions are another instance of non-local restrictions on agreement. The PC-PATR implementation of these restrictions is discussed in section 6.2.[18]

6 Binding of Reflexives and Equi Constructions

We have now accounted for cases of movement, which GB handles via coindexed traces, by feature passing in PC-PATR with unification constraints at both ends. The binding of reflexives and equi constructions can be modeled in a parallel way, since coindexing is again involved.

6.1 GB: Binding Theory

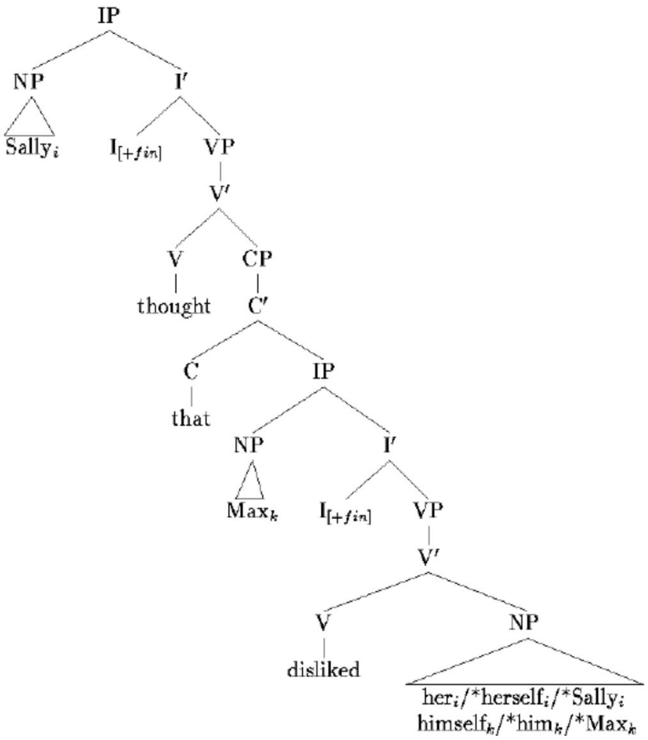
Binding theory seeks to provide an account of where a reflexive versus a pronoun versus a full NP is allowed or required. Relevant examples are given in (63), where intended coreference is indicated by subscripts.

(63) a. $Sally_i$ thought that Max_k disliked her_i/*$herself_i$/$Sally_i$.

b. $Sally_i$ thought that Max_k disliked $himself_k$/*him_k/Max_k.

GB accounts for these distinctions structurally, so consider the tree in (64).

(64)



Mathematical definitions for *c-command, binding,* and *A-binding* are given in (65). Note that *c-command* simply formalizes the notion of 'higher in the tree than'.

(65) a C-COMMANDS b iff

- a. a does not dominate b, and
- b. the first branching node that dominates a also dominates b.

a BINDS b iff

- a. a c-commands b, and
- b. a and b are coindexed.

a A-BINDS b iff

- a. a is in an argument position, and
- b. a binds b.

In the tree (64) *Sally* and *Max* are both in argument positions and they both c-command the object of *disliked* and are coindexed with it. Therefore the object of *disliked* is A-bound.

The principles of Binding Theory determine whether a pronoun or a reflexive anaphor or a full NP is correct in a particular position. Informally, Principle A says that an anaphor can only be used when the position that A-binds it is local enough (usually within the same finite IP): *Max* is close enough to the object of *disliked* so that the anaphor is correct, but *Sally* is too far away to use *herself*.

Principle B says that a pronoun can only be used if it is not A-bound at all, or if its A-binder is far enough away. This is why *him* cannot be used to refer back to *Max* but *her* may refer back to *Sally*.

Finally, Principle C says that nonpronominals may not be A-bound in the sentence at all, which rules out repetition of full nominals.

The Principles of Binding Theory thus partition nominal phrases into three different types. These partitions are characterized by the two features [±anaphoric] and [±pronominal], where pronouns are [-ana,+pro], reflexives and reciprocals are [+ana,-pro], and full nominal phrases are neither pronouns nor anaphors so they are [-ana,-pro].

The chart in (66) shows these featural distinctions and which Principle of Binding Theory applies. Empty categories are also included, since GB claims that the chain coindexing established by movement is equivalent to the coindexing in binding relationships between overt nominals. Four types of empty categories are recognized, corresponding to the four possible feature specifications.

(66) Featural Distinctions for Overt NPs and Empty Categories

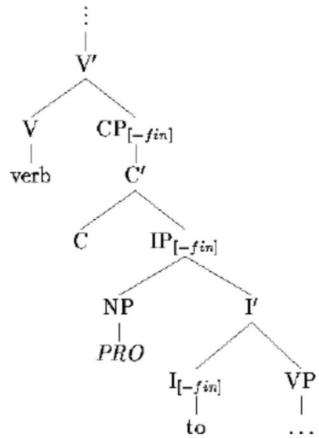
Features	Binding Principle	Overt Nominals	Empty Categories
[-pro,+ana]	A	himself	trace of A-movement
[+pro,-ana]	B	him	*pro*
[-pro,-ana]	C	John	trace of $\bar{A}$-movement
[+pro,+ana]	A/B		*PRO*

Section 4 covered the trace of A-movement in passive and unaccusative constructions. By saying that this trace is anaphoric and subject to Principle A, the movement is restricted to local domains. The trace of $\bar{A}$-movement in the formation of content questions was introduced in section 3. This trace is subject to Principle C in that it cannot be bound by an element in an argument position. Since $\bar{A}$-movement is movement to a nonargument position, this requirement is clearly met.

The two new empty categories are not traces, but empty elements in the lexicon. *pro* is the empty pronoun allowed in *pro*-drop languages, usually because of agreement morphology on the verb to specify the person and number of the subject, e.g. *pro No hablo español*. This empty pronoun shows up in all the same places that an overt pronoun does and is therefore subject to Principle B.

Finally, *PRO* is the empty subject in non-finite clauses, sometimes called controlled *PRO*. Since it is both anaphoric and pronominal, it is subject to both Principles A and B of Binding Theory. From this is derived the fact that *PRO* must be ungoverned and therefore also not receive Case.[19] The only legal position for *PRO* (and only *PRO* can be there) is shown in (67).

(67)



Both *pro* and *PRO* have intended reference, even though they are null. Since they are referential, they bear a semantic role, unlike the dummy *it*, which has phonetic content but is nonreferential.

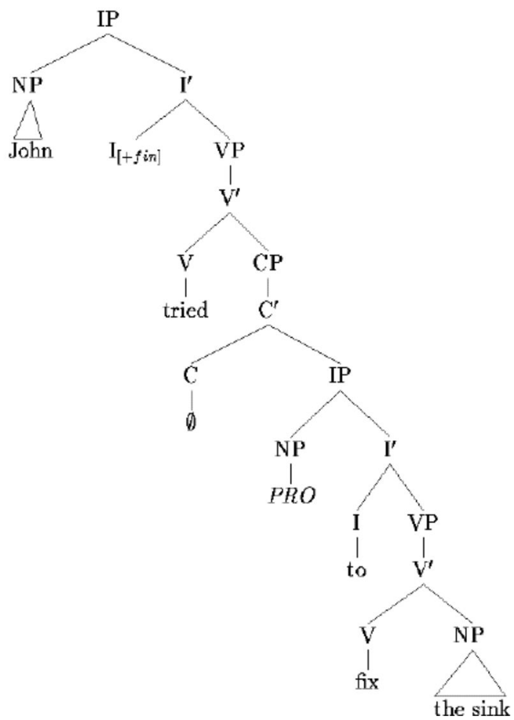
English does not use *pro*, but we can see the use of *PRO* in equi constructions, such as:

(68) a. John tried to fix the sink.

b. Bill wants to win the prize.

Note that both *try* and *want* assign the semantic role of AGENT to their external argument (subject), in contrast to raising predicates like *seem* and *likely*. The verbs in the lower clause, *fix* and *win*, also assign AGENT roles to their external argument. GB requires a one-to-one mapping between NP positions and semantic roles, so a null category is necessary in these constructions, namely, *PRO*. The tree for (68a) is given in (69).

(69)



6.2 PC-PATR: Agreement Features Passed Up in [Bind]

Within PC-PATR, only the Principle A requirement that an anaphor agree with the closest subject is implemented. It is not possible to enforce the requirements of Principles B and C that a pronoun or full NP does not agree, at least for third person pronouns. If negation capabilities were added to PC-PATR, it would be easier to restrict against two first or second person pronouns as subject and object of the same clause.

In order to account for the reflexive agreement, `[gender]` features are added to all of the nouns, pronouns, and reflexives in the lexicon under the `[agr]` feature discussed in section 5.2. Reflexives are further distinguished by the feature `[type:[anaphor:+]]`. All of the phrase structure rules which could introduce an anaphor (including NP objects of V and objects of prepositions which are themselves objects of either V or N) are also constrained to place the `[agr]` features for the anaphor into the `[bind]` feature matrix, which then acts in a parallel way to the `[moved]` feature matrix. The `[bind]` feature percolates as far as the closest subject position to enforce coreference, or unification of the `[agr]` features.

The most interesting case is the interaction between reflexives and equi and/or raising constructions. These are the only cases which allow apparent long-distance binding of anaphors. However, as shown in (70), the coindexed *PRO* or A-trace serves as the local A-binder of the reflexive and then it takes its reference from the upper clause subject.

(70) a. John$_i$ thought that Sue liked him$_i$/*himself$_i$.

b. John$_i$ tries PRO$_i$ to like himself$_i$/*him$_i$.

c. John$_i$ seemed t_i to like himself$_i$/*him$_i$.

Equi constructions, such as (70b) are handled in PC-PATR via the null category *PRO*, which is introduced into the grammar only by the rule in (71). This rule is restricted to embedded, nonfinite, non-question clauses (71a-c). The subject subcat list is stepped (71e), since *PRO* counts as the subject of the embedded predicate. The crucial constraint allowing the apparent long-distance binding of reflexives is (71f), where the `[bind]` features are percolated up to the IP. This is in contrast to the regular subject rule (72) (used also for the upper clause of an equi or raising construction) where the `[bind]` features must unify with the `[agr]` features on the overt NP subject.

(71)
```
rule{equi-lower clause}
     IP = PRO I'
    a.   <IP type root>       = -
    b.   <IP type fin>        = -
    c.   <IP type q>          = -
    d.   <IP type>            = <I' type>
    e.   <IP subcat subj>     = <I' subcat subj rest>
    f.   <IP bind>            = <I' bind>
```

(72)
```
rule{normal position subject}
     IP = NP I'
    a.   <I' bind agr person> = <NP agr person>
    b.   <I' bind agr number> = <NP agr number>
    c.   <I' bind agr gender> = <NP agr gender>
```

7 Conclusion

This documentation of work in progress is instructive in at least two ways. First, it presents the development of a parser in PC-PATR that accounts for a broad range of data, illustrating the capabilities of the PC-PATR program. Secondly, the attempt to implement GB theory computationally forced precision into the analysis at several points. For example, the implementation showed that 'coindexing' and feature sharing of nonlocal constituents must be passed through all local subtrees, including intermediate projections, whereas GB states only that heads and their maximal projections and heads and their specifiers may share features.

Appendix A: Two Examples

Appendix B: Example Sentences the Grammar Accounts For

Endnotes

1 Neither GB nor PC-PATR can be fully documented here. For more complete coverage, see Black (1996), Sells (1985) or the original works of Chomsky (1981, 1986) on GB and McConnel (1995) on PC-PATR. The PC-PATR software and documentation are available for downloading from http://www.sil.org/pcpatr/.

2 The computer files for the current grammar developed for English and the current English lexicon file are available for downloading:
englishgb.zip (DOS/Windows)
englishgb_sea.hqx (Macintosh)
englishgb.tar-gz (UNIX)

3 Ideas for the presentation of the GB material plus a number of the examples are from courses taught by Sandra Chung at the University of California, Santa Cruz.

4 A * before a word, phrase or sentence indicates ungrammaticality.

5 The order of the elements on the right side of these phrase structure rules is dependent upon the basic word order in the language.

6 Both C^0 and I^0 are functional heads rather than lexical heads. More recently, additional functional heads such as Neg^0, Agr^0, and $Tense^0$ have been proposed to further break down I^0 (Pollock 1989) and D^0 has been proposed as the functional head of the noun phrase, taking NP as its complement (Abney 1987, Stowell 1989). None of these additional functional heads (or the phrases they project) are used in the English implementation in PC-PATR, though the implementation for Quiegolani Zapotec requires a NegP phrase for negation between CP and IP and uses the DP structure for nominal phrases.

7 The original PATR-II formalism allowed X for subcategorized complements (Shieber 1986:72). While the X is desirable in capturing the generalization that whatever follows must be a complement determined by the subcat information, it turns out to not be feasible in a fuller system due to the differing constraints needed for the various complements.

8 GB does not allow subcategorization for subjects directly, but only for whether or not there is a semantic role assigned to the subject (see section 4.1). In the PC-PATR implementation, I allow subcategorization only for an NP subject, which will have a semantic role, or `none`.

9 In the current form of the grammar, the highest auxiliary is assumed to begin in the I^0 position, rather than having to move there, so V-trace fills only that position for questions (45d). For declarative sentences with an auxiliary, the highest auxiliary is placed in the tree via the rule `I' = V VP`, constrained by `<V type aux> = +`.

10 The `[moved]` feature is a personal innovation, based upon the slash features of GPSG (Gazdar, Klein, Pullum & Sag 1985). Following GB, `[moved]` has three subordinate features: `[A-bar]` (for *wh*-movement), `[head]` (for head movement such as movement of the auxiliary), and `[A]` (for argument movement to be discussed in section 4).

11 Rules (48a & c) are restricted to use in root clause questions only, by other constraints not shown here. (48b) is a general rule for the position of the subject as the specifier of IP, but the constraints listed here are included only under the root clause question option for this rule.

12 In the GB analysis, the aspect marking is in I^0 at D-structure and the morphological subcategorization requirements motivate movement of the verb to adjoin to I^0 (see Black 1994, Rizzi & Roberts 1989). For PC-PATR, only the position of full words at S-structure is modeled. Therefore, the verb with its aspect marking is shown in I^0 at S-structure.

13 The line containing the semantic roles is introduced by the logical semantic predicate which is indicated by the syntactic verb form followed by a prime. The lexical entry thus specifies both syntactic and semantic subcategorization information.

14 For example, auxiliaries do not:
(i) Joe may have hit a home run.

There is only one external argument = *John*. Which verb assigns a semantic role to *John*? It is the AGENT of the verb *hit*; *a home run* is the THEME of *hit*. Auxiliaries do not assign external arguments: there is nothing anyone did or experienced to *may*. Instead they simply select a VP complement that is a type of EVENT or STATE.

15 The PC-PATR documentation includes discussion of a Lexical Rule. Unfortunately, the GB lexical rule assumes different output conditions than PC-PATR does, so I could not use that mechanism for handling passives in GB.

16 The information on Tatana' was obtained from John Dillon at the Computational Linguistics workshop at the 1994 session of the Summer Institute of Linguistics held at the University of North Dakota.

17 These positions which receive Case are the positions that the particular heads *govern*.

18 An area where agreement is particularly tricky is in coordination constructions. The current grammar restricts coordinate constructions to third person, plural agreement. Much more would need to be added to account for the full distribution of agreement patterns in coordinate constructions.

19 This is based upon the formal definition of the local domain of Principle A, called the governing category.

References

Abney, Stephen. 1987. *The English Noun Phrase and its Sentential Aspect*. Ph.D. Dissertation. MIT. Cambridge, Mass.

Baker, Mark C. 1988. *Incorporation: A Theory of Grammatical Function Changing.* University of Chicago Press. Chicago.

Black, Cheryl A. 1994. *Quiegolani Zapotec Syntax.* Ph.D. Dissertation. University of California, Santa Cruz.

Black, Cheryl A. 1996. "A step-by-step introduction to Government and Binding theory of syntax." *Notes on Linguistics.* A seven-part series of articles beginning with 73:5-12.

Bresnan, Joan, ed. 1982. *The Mental Representation of Grammatical Relations*. MIT Press. Cambridge, Mass.

Burzio, Luigi. 1986. *Italian Syntax: A Government-Binding Approach.* Studies in Natural Language and Linguistic Theory. Kluwer. Dordrecht.

Chomsky, Noam. 1981. *Lectures on Government and Binding.* Foris. Dordrecht.

Chomsky, Noam. 1986. *Barriers.* MIT Press.

Gazdar, Gerald, Ewan Klein, Geoffrey Pullum, and Ivan Sag. 1985. *Generalized Phrase Structure Grammar.* Harvard University Press. Cambridge, Mass.

Koopman, Hilda and Dominque Sportiche. 1991. "On the position of subjects" in James McCloskey, ed. *The Syntax of Verb-Initial Languages. Lingua* Special Edition. Amsterdam. 211-258.

McCloskey, James. 1991. "Clause structure, ellipsis and proper government in Irish" in James McCloskey, ed. *The Syntax of Verb-Initial Languages. Lingua* Special Edition. Amsterdam. 259-302.

McConnel, Stephen. 1995. "PC-PATR Reference Manual." Alpha test version 0.97a9 of October, 1995. Academic Computing. Dallas, Tex. (The PC-PATR software and documentation are available at http://www.sil.org/pcpatr/.)

Perlmutter, David. 1978. "Impersonal Passives and the Unaccusative Hypothesis." In J. Jaeger et al, eds., *Proceedings of the Fourth Annual Meeting of the Berkeley Linguistics Society*. University of California, Berkeley.

Pollard, Carl and Ivan A. Sag. 1987. *Information-Based Syntax and Semantics: Volume 1 Fundamentals*. Center for the Study of Language and Information. Stanford, Calif.

Pollard, Carl and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. University of Chicago Press. Chicago.

Pollock, Jean-Yves. 1989. "Verb Movement, UG, and the structure of IP." *Linguistic Inquiry* 20.3:365-424.

Rizzi, Luigi and Ian Roberts. 1989. "Complex inversion in French." *Probus* 1.1:1-30.

Sells, Peter. 1985. *Lectures on Contemporary Syntactic Theories.* Center for the Study of Language and Information. Stanford, Calif.

Shieber, Stuart M. 1986. *An Introduction to Unification-Based Approaches to Grammar.* Center for the Study of Language and Information. Stanford, Calif.

Stowell, Tim. 1989. "Subjects, specifiers, and X-Bar theory" in Mark Baltin and Anthony Kroch, eds. *Alternative Conceptions of Phrase Structure*. The University of Chicago Press. 232-262.

Date created: *17-Dec-1997*
URL: *http://www.sil.org/silewp/1997/006/silewp1997-006.html*
Questions/Comments: *SILEWP@sil.org*

[SILEWP 1997 Contents | SILEWP Home | SIL Home]