A Logic-Based Government-Binding Parser for Mandarin Chinese

Hsin-Hsi CHEN

Department of Computer Science and Information Engineering
National Taiwan University
Taipei, Taiwan 10764, R.O.C.
NTUT046@TWNMOE10.BITNET

Abstract

Mandarin Chinese is a highly flexible and context-sensitive language. It is difficult to do the case marking and index assignment during the parsing of Chinese sentences. This paper proposes a logic-based Government-Binding approach to treat this problem. The grammar formalism is specified in a formal way. Uniform treatments of movements, arbitrary number of movement non-terminals, automatic detection of grammar errors beforehand, and clear declarative semantics are its specific features. Many common linguistic phenomena of Chinese sentences are represented with this formalism. For example, topic-comment structures, the ba-constructions, the bei-constructions, relative clause constructions, appositive clause constructions, and serial verb constructions. A simple pronoun resolution is touched upon. The expressive capabilities and the design methodologies show this mechanism is also suitable for other flexible and context-sensitive languages.

1. Introduction

Chinese is a highly flexible language. The same meaning may be represented in many different Chinese patterns. In other words, Chinese provides many ways for the native speakers to express their feelings. For example, a sentence like "I have told Mr. Lee that they want these books" in English, we can form multiple different patterns in Chinese:

(a) 我告訴過[$_{np}$ 李先生] [$_{s}$ 他們要這些書]。
I have told [$_{np}$ Mr. Lee] [$_{s}$ that they want these books].

(b) [$_{np}$ 李先生]$_i$,我告訴過 t_i [$_{s}$ 他們要這些書]。
[$_{np}$ Mr. Lee]$_i$, I have told t_i [$_{s}$ that they want these books].

(c) 我告訴過[$_{np}$ 李先生] [$_{s}$ [$_{np}$ 這些書]$_j$ 他們要 t_j]。
I have told [$_{np}$ Mr. Lee] [$_{s}$ that [$_{np}$ these books]$_j$ they want t_j].

(d) [$_{np}$ 李先生]$_i$, 我告訴過 t_i [$_{s}$ [$_{np}$ 這些書]$_j$ 他們要 t_j]。
[$_{np}$ Mr. Lee]$_i$, I have told t_i [$_{s}$ that [$_{np}$ these books]$_j$ they want t_j].

(e) [$_{np}$ 李先生]$_i$, [$_{np}$ 這些書]$_j$, 我告訴過 t_i [$_{s}$ 他們要 t_j]。
[$_{np}$ Mr. Lee]$_i$, [$_{np}$ these books]$_j$, I have told t_i [$_{s}$ that they want t_j].

In reality, it shows the specific pattern: topic-comment structure in Mandarin Chinese. Topicalization may be deemed one of the movement transformations. Examples (b) and (c) specify an object is moved to the topic position. Examples (d) and (e) are sentences with multiple topics. We can realize that the more predicates a sentence includes, the more topic positions it has. And thus, the more complicated patterns may be generated. It is good for the language users, however, it is difficult to process this type of languages in computer.

Chinese is also a highly context-sensitive language. There are so many phenomena, e.g. index assignment, case marking, *etc.*, depending on the context information even within a Chinese sentence. The index assignments in the topic-comment patterns shown above explain this point. Examples (d) and (e) are legal interpretations. However, their bindings are different. The former is a serial binding, and the latter is a crossed binding. Serial binding is not always true. For example, the index assignment cannot be

* [$_{np}$ 李先生]$_i$, [$_{np}$ 這些書]$_j$, 我告訴過 t_j [$_{s}$ 他們要 t_i]。

* [$_{np}$ Mr. Lee]$_i$, [$_{np}$ these books]$_j$, I have told t_j [$_{s}$ that they want t_i].

This is because the object that someone told must be an animate. Therefore, the index assignment, which is a necessary step toward correct interpretation of natural language sentences, is difficult in computer.

This paper proposes a Government-Binding approach to deal with these highly flexible and context-sensitive languages such as Mandarin Chinese. It is organized as follows. Section 2 specifies the concepts of Government-Binding Theory. Section 3 gives a formal definition of Government-Binding based logic grammars. Section 4 demonstrates a Chinese parser from several context-sensitive constructions, and touches on the simple pronoun resolution within a Chinese sentence. Section 5 concludes the remarks.

2. Government-Binding Theory

Government-Binding (GB) Theory /Chomsky 1981, Sells 1985/ is the descendant of Transformation Grammars /Radford 1981/. Its simplified organization is shown in Figure 1. Move - α , which is a general operation, moves anything anywhere between d-structure and s-structure, and between s-structure and logical form. GB Theory includes a series of modules that contain constraints and principles which govern the movement transformation.

The Projection Principle preserves the syntactic information and the semantic information at each level (d-structure, s-structure, and logical form) during the movement transformation. Trace Theory postulates that there exist various empty categories at various levels of mental representation.

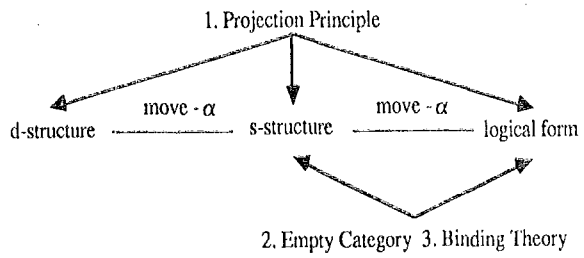

Figure 1. Government-Binding Theory

Thus, we must have the capabilities to verify the relationship between the moved constituent and the empty constituent. GB Theory provides several mechanisms for the verification. The Empty Category Principle (ECP) says "*A trace must be properly governed.*" That is, we must find some α that c-commands the trace β . And α binds β iff (a) α c-commands β , and (b) α and β are co-indexed. Their definitions are based on C-Command Condition. The C-Command Condition states the following:

α c-commands β if and only if the first branching node dominating α also dominates β , and α does not itself dominate β .

It states a co-reference relation between a moved element and its trace. The Subjacency Condition is given in the following:

Any application of Move - α may not cross more than one bounding node.

It specifies island constraints on the moved constituents.

The Binding Theory /Sells 1985/ shown below is used for simple pronoun resolution:
(Principle A) An anaphor is bound in its Governing Category.
(Principle B) A pronominal is free in its Governing Category.
(Principle C) An R-expression is free.
Anaphors include reflexive and reciprocals, pronominals include pronouns, and R-expressions include all other noun phrases.

3. A Government-Binding Based Logic Grammar Formalism

The formal definition of Government-Binding based Logic Grammars (GBLGs) is specified incrementally in the following.

Definition 1. A *Government-Binding based Logic Grammar* is a 6-tuple GBLG = (T,Σ,B,S,C,R) where:

(1) T is the set of lexical terminals. Each lexical terminal is denoted by an atomic formula with lexical category as its predicate symbol.

(2) Σ is the set of non-terminals. $\Sigma = \Sigma_P \cup \Sigma_V \cup \Sigma_M \cup \Sigma_G$ where:

(a) Σ_P is the set of phrasal non-terminals. Each phrasal non-terminal is represented by an atomic formula with phrasal category as its predicate symbol.

(b) Σ_V is the set of virtual non-terminals. Each virtual non-terminal is specified by an atomic formula.

(c) Σ_M is the set of movement non-terminals. A movement non-terminal is one of the following two forms:

A <<< B or B >>> A where $A \in T \cup \Sigma_P \cup \Sigma_V$, and $B \in \Sigma_V$. Σ_{LM} and Σ_{RM} denote the set of non-terminals A <<< B and the set of non-terminals B >>> A, respectively.

(d) Σ_G is the set of goals. Each goal is denoted by a literal.

(3) $B \subset \Sigma_P$ is the set of bounding non-terminals. A bounding non-terminal is a phrasal non-terminal with bounding node as its predicate symbol.

(4) $S \in \Sigma_P$ is the start non-terminal.

(5) C is the set of logic connectives 'and' and 'or' that are denoted by ',' and ';' respectively. A *grammar element* is defined recursively in terms of logic connectives as follows:

(a) A lexical terminal $L \in T$ is a grammar element.

(b) A phrasal non-terminal $P \in \Sigma_P$ is a grammar element.

(c) A virtual non-terminal $V \in \Sigma_V$ is a grammar element.

(d) A movement non-terminal $M \in \Sigma_M$ is a grammar element.

(e) A goal $\in \Sigma_G$ is a grammar element.

(f) If A and B are grammar elements, then (A,B) and (A;B) are grammar elements.

The first five types are called *basic grammar elements*, and the last one is a *compound grammar element*. Let G_I and G_E be the set of basic grammar elements and the set of compound grammar elements, respectively.

(6) R is the set of production rules. A production rule is of the following form:

$$X_0 \text{ --> } X_1\ C_1\ X_2\ C_2 \ldots C_{(m-1)}\ X_m$$

where $X_0 \in \Sigma_P$,

$X_i \in G_E$ for $1 \le i \le m$, and

$C_i \in C$ for $1 \le i \le (m-1)$.

It is obvious each production rule can be translated into a sequence of production rules with the logical operator 'and' only.

An example written with this formalism is shown as follows. It captures the relative clauses in English like "The man who he met is a teacher."

(r1) s --> np, vp.
(r2) np --> pronoun.
(r3) np --> det, noun.
(r4) np --> det, noun, rel.
(r5) vp --> tv, np.
(r6) vp --> tv, trace.
(r7) vp --> iv.
(r8) rel --> rel_pronoun <<< trace, s.

where T = {pronoun, det, noun, tv, iv, rel_pronoun},
Σ_P = {s, np, vp, rel},
Σ_V = {trace},
Σ_M = {rel_pronoun <<< trace}, and
B = {s, np}.

The rule (r8) describes a constituent in phrase structure *s* is extraposed to the *rel_pronoun* position. Which constituent may be moved from which position is specified by rule (r6).

Definition 2. For $X \in \Sigma_P$, $Y \in \Sigma_V$ and TR is a transitive relation, X TR Y if

(1) X is the rule head of a production rule, and Y is a grammar element in its rule body, or

(2) X is the rule head of a production rule, $I \in \Sigma_P$ is a grammar element in its rule body, and I TR Y, or

(3) there exist $I_1, I_2, \ldots$, and $I_n \in \Sigma_P$, such that X TR I_1

TR I_2 TR ... TR I_n TR Y.

The transitive relation TR is also a dominate relation. This is because TR is a dominate relation between a phrasal non-terminal and a virtual non-terminal.

Definition 3. A production rule X_0 --> X_1, X_2, ..., X_m (where $X_i \in G_I$ for $1 \le i \le m$) is *significant* if it satisfies the extra restrictions:

(1) for any grammar element $X_i = (A <<< B) \in \Sigma_{LM}$, there must exist some X_j, $i < j \le m$, such that $(X_j, B) \in TR$.

(2) for any grammar element $X_i = (B >>> A) \in \Sigma_{RM}$, there must exist some X_j, $1 \le j < i$, such that $(X_j, B) \in TR$.

A logic grammar GBLG is *significant* if each production rule $\in R$ is significant. The above sample grammar is significant for the following reasons:

(1) The rules (r1) - (r7) are significant trivially.

(2) The rule

rel --> rel_pronoun <<< trace, s

is significant because there exists a transitive relation TR_1 such that *s* TR_1 *vp* TR_1 *trace.*

Proposition 1. The c-command condition is embedded implicitly in GBLGs if these grammars are significant.

Proof. For a significant production rule:

X_0 --> X_1, X_2, ..., X_m

if $X_i = (A <<< B) \in \Sigma_{LM}$ then there must exist some X_j ($i < j \le m$), such that X_j dominates the virtual non-terminal B in the other production rule. The phrasal non-terminal X_0 is the first branching node that dominates A and X_j, and thus also dominates B. Therefore, A c-commands B. $X_i = (B >>> A) \in \Sigma_{RM}$ has the similar behavior.

This property can be used to check the correctness of grammars automatically before parsing.

Definition 4. The transitive relation $TR_{subjacency}$ is a subset of TR and satisfies the restrictions: for $X \in \Sigma_P$, $Y \in \Sigma_V$, X $TR_{subjacency}$ Y if X TR I_1 TR I_2 TR ... TR I_n TR Y, and there does not exist more than one I_j such that $I_j \in B$.

Proposition 2. A significant logic grammar is a restrictive context sensitive grammar. This is because the truth value of a movement non-terminal depends on the appearance of a virtual non-terminal preceding or following it.

/Chen 1988/ proposes a bottom-up parsing system for GBLGs. Figure 2 shows the execution of our sample grammar for the sentence "The man who he met is a teacher". The label on the arc indicates the step number during parsing. The empty constituent *trace* is generated in phrase *vp*, then passed to phrase *s*, and finally cut in phrase *rel*. Compared with other logic programming approaches /Matsumoto 1983, McCord 1987, Pereira 1981, Stabler 1987/, especially RLGs /Stabler 1987/, GBLGs have the following features:

(1) the uniform treatments of leftward movement and the rightward movement,

(2) the arbitrary number of movement non-terminals in the rule body,

(3) automatic detection of grammar errors before parsing.

The former two features are useful to express the highly flexible languages like Chinese.

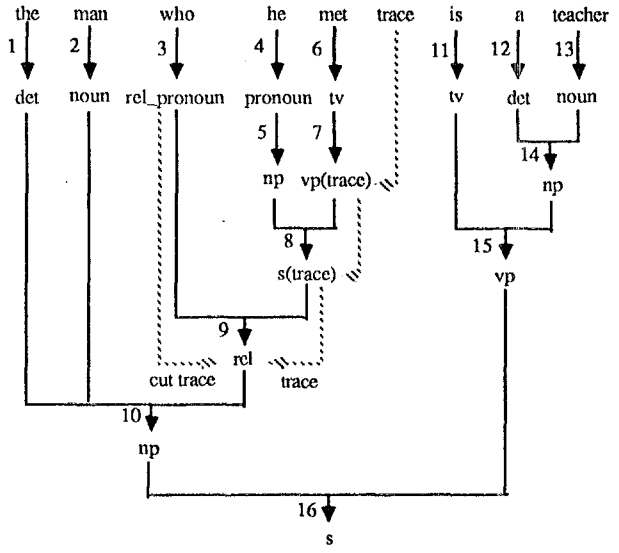

Figure 2. Sample Parsing

4. A Chinese Parser

4.1 Topic-comment Structures

Topic-comment structure is one of the specific features in Mandarin Chinese. There are several interesting linguistic phenomena concerning these structures:

(1) Topic may be moved from the argument positions in the comment - as subject, direct object, or indirect object.

(2) Many categories may appear in the topic position, e.g. n", s', v", or p".

(3) There may be multiple topics in a sentence.

(4) The comment may not contain a constituent which is anaphorically related to the element in the topic.

Under the above observations, topic may be represented as:

```
topic(topic(N2bar),n2bar,Semantic,Index,Case) -->
    n2bar(N2bar,Semantic,Index,Case,Classifier).
```

The second argument of predicate *topic* specifies the phrasal category of the topic, i.e., *n2bar* in this example. It is important for the parser to decide whether the constituent may co-index with a trace.

Next, the production rules for generating sentences are shown as follows:

```
s1bar(s1bar(Topic1,Topic2,S)) -->
    topic(Topic1,Cat1,S1,I1,Case1)
        <<< trace(topic,info(Cat1,S1,I1,Case1)),
    topic(Topic2,Cat2,S2,I2,Case2)
        <<< trace(topic,info(Cat2,S2,I2,Case2)),
    s(S).
s1bar(s1bar(Topic,S)) -->
    topic(Topic,Cat,S,I,Case)
        <<< trace(topic,info(Cat,S,I,Case)),
    s(S).
s1bar(s1bar(S)) --> s(S).
```

Of these three production rules, the first two define the "topic-comment" pattern, and the last one is a rule without topic.

Finally, the phrasal non-terminal *s* is introduced.

```
s(s(N2bar,V2bar)) -->
   n2bar(N2bar,Semantic,Index,Case,Classifier),
   v2bar(V2bar,Semantic,Index,Case,subj,nonbei).
s(s(t(Case,Index),V2bar)) -->
   trace(X,info(n2bar,Semantic,Index,Case)),
   v2bar(V2bar,Semantic,Index,Case,subj,nonbei).
s(s(N2bar,V2bar)) -->
   n2bar(N2bar,S,I,C,Classifier)
      <<< trace(bei,info(n2bar,S,I,C)),
   v2bar(V2bar,S1,I1,C1,subj,bei).
s(s(t(C,I),V2bar)) -->
   trace(relative,info(n2bar,S,I,C))
      <<< trace(bei,info(n2bar,S,I,C)),
   v2bar(V2bar,S1,I1,C1,subj,bei).
s(s(V2bar)) --> v2bar(V2bar,_,_,_,nosubj,nonbei).
```

The first *s* rule is a normal case, i.e., no movement. *Semantic* denotes the semantic feature of the head noun. It must be unifiable with the semantic feature provided by the matrix verb with the type tree matching /McCord 1987/. The same logical variable *Case* appears in the phrasal non-terminals *n2bar* and *v2bar*. It means the case of subject is assigned by the matrix verb externally according to θ - theory. The second *s* rule captures one of the movement transformations - relativization, topicalization, ba-transformation, or bei-transformation. An overt noun phrase is moved via the former operation, thus a virtual non-terminal *trace(X,info(n2bar,Semantic,Index,Case))* is left at the empty site. It specifies only *n2bar* can appear here, and what kinds of movements are not concerned. The semantic feature and case are confined by the matrix verb. The third *s* rule deals with bei-transformation. For example,

那個小偷$_i$被警察抓走 t_i 了。

(The thief$_i$ is arrested t_i by the police.)

The thief (那個小偷) is not a logical subject of *v2bar*. The real subject is the object of *bei* ('被'), i.e., the police. Thus, a different group <S,I,C> of variables is used. The *n2bar* acts as the object of *v2bar* or the subject of the embedded sentence. The fourth *s* rule captures double movements for an *n2bar*. For example,

t_i 被警察抓走 t_j 的那個小偷$_{(i,j)}$ 又逃跑了。

(The thief$_i$ arrested t_i by the police escaped again.)

A left-moved constituent (那個小偷', the thief) is moved rightward furthermore. In this rule, two virtual non-terminals appear at both sides of movement operator '<<<'. The fifth *s* rule describes those sentences without subject. An atom *nosubj* instead of *subj* specifies such a situation.

4.2 Noun Phrase

A noun phrase can be a pronoun, a simple noun, or a noun plus other elements that act as pre-modifiers of that noun. Those elements are (1) classifier phrases, (2) associative phrases, and (3) modifying phrases. Only associative phrase, relative clause, and appositive clause are listed in the following. Associative phrase denotes two noun phrases are linked by a special Chinese word *de* ('的'). For example,

中國的人口 (the population of China).

The rule

```
n2bar(n2bar(A,N2bar),Semantic,Index,Case,Classifier)-->
   asc(A),
   n2bar(N2bar,Semantic,Index,Case,Classifier)
```

represents this construction. The definition of associative clause is:

```
asc(asc(N2bar,De)) -->
   n2bar(N2bar,Semantic,Index,Case,Classifier),
   * de(De).
```

Both relative clause and appositive clause are nominalization in the form: *nominalization + head noun*, and are defined as follows:

```
rel(rel(S,De)) --> s(S), * de(De).
app(app(S,De)) --> s(S), * de(De).
```

However, they are different in the restricting the reference of the head noun. The head noun that a relative clause modifies refers to some unspecified participant in the nominalization part. For example,

t_i 種水果的農夫$_i$

(the former$_i$ who t_i grows fruits), and

他們種 t_i 的水果$_i$

(the fruits$_i$ that they grow t_i).

The head noun '水果' (the fruits) refers to an empty constituent (either subject or object) in the relative clause. This type of constructions can be considered a rightward movement. For appositive clause and head noun pair, the head noun does not refer to any entity in the modifying clause, i.e., appositive clause. For example,

我們租房子的事

(the matter concerning our renting a house).

The nominalization '我們租房子' (our renting a house) serves as a complement to the head noun '事' (the matter). This type of constructions cannot be regarded as a movement transformation. Two rules are specified for them:

```
n2bar(n2bar(Rel,N2bar),S,I,C,Classifier) -->
   rel(Rel),
   trace(relative,info(n2bar,S,I,C1))
      >>> n2bar(N2bar,S,I,C,Classifier).
n2bar(n2bar(App,N2bar),S,I,C,Classifier) -->
   app(App),
   n2bar(N2bar,S,I,C,Classifier).
```

The only difference between these two rules is a trace has to be found in relative clause. Note the cases of the empty constituent and the overt constituent may be different in *relative clause + head noun* construction. For the sake of space, the *n1bar* is neglected in this paper.

4.3 Verb Phrase

Different from a noun phrase, a verb phrase may have pre-modifiers and post-modifiers. The preverbal specifiers are ba-phrases, bei-phrases, adverbial phrases, degree phrases, preposition phrases, quantifier phrases, aspect, and modal. The postverbal modifiers are sentential constructions, adverbial phrases, quantifier phrases, classifier phrases, prepositional phrases, and aspect. Only Serial Verb Constructions (SVCs) are about to discuss in detail. The rule

```
v2bar(v2bar(Va1bar,Vb1bar),S,I,[C1,C2],subj) -->
   v1bar(Va1bar,S,I,C1,subj),
   v1bar(Vb1bar,S,I,C2,subj)
```

means two separate events juxtaposed together, e.g. 我 [$_V$' 買票] [$_V$' 進去] (I [$_V$' bought a ticket] and [$_V$' went in]). It is one of the SVCs. The two events have the identical subject, but cases may be different. The other groups of SVCs are:

(1) One verb phrase or clause serving as the direct object of another verb, e.g.

我要去學校。 (I want to go to school.)

我要他去學校。 (I want him to go to school.)

(2) Pivotal constructions, e.g.

我委託他辦一件事。
(I entrust him to take care of an affair.)

(3) Descriptive clauses, e.g.

她炒了一道菜我很喜歡吃。
(She cooked a dish that I very much enjoyed eating.)

Only the former two are considered. The verbs with first use are classified into *t2* and *t3*, and the verbs with the second use, i.e., pivotal construction, are classified into *t8*. It is not easy to define descriptive clauses with a rule or a new category, e.g. POSSESSIVE /Yang 1987/. This is because the descriptive clause is optional. Without this clause, the original sentence is acceptable too. Furthermore, many verbs may be used with the descriptive clauses.

The lowest level *v1bar* (v') touches on the uses of the subcategorization frames of the specified verb. According to the frames and ECP, a virtual non-terminal *trace* is placed wherever it is needed. For example,

```
v1bar(v1bar(T1,N2bar),Semantic,Index,Case,HasSubj)-->
    * t1(T1,HasSubj:Semantic:Case,Semantic1:Case1),
    n2bar(N2bar,Semantic1,Index1,Case1,Classifier).
v1bar(v1bar(T1,t(Case1,Index1)),Semantic,Index,Case,
        HasSubj) -->
    * t1(T1,HasSubj:Semantic:Case,Semantic1:Case1),
    trace(X,info(n2bar,Semantic1,Index1,Case1)).
v1bar(v1bar(T2,pseudoS(e(Case1,Index),V2bar)),
        Semantic,Index,Case,subj) -->
    * t2(T2,subj:Semantic:Case),
    v2bar(V2bar,Semantic,Index,Case1,subj).
```

The lexical category *t1* denotes transitive verb. Here, the trace may be generated by any movement transformation. The third rule is for SVCs. Note *v2bar* should have a subject and share it (*Index*) with the matrix verb. Thus, the semantic features of the two are the same. However, cases may be different. That is, one is assigned by the matrix verb, and the other one by the embedded verb. The rules for other lexical categories are omitted in this paper. The details can refer to /Lin 1989/.

4.4 Ba-construction

Ba-construction is usually generated by ba-transformation, which is one of the movement transformations. The direct object is placed immediately after '把' (ba) and before the verb like:

subject '把' (ba) direct object verb.

For example,

我把三本書$_i$都賣 t$_i$ 了。(I sold all three books.)

However, there is another pattern for ba-construction:

subject '把' (ba) object 1 verb object 2.

It is not constructed by movement transformation because some noun phrase appears after verb, i.e., object 2. For example,

我把蘋果吃了三個。(I ate three of apples.)

It shows a part-whole relation between object 1 and object 2. In the well-performed parsing systems, all the two patterns must be treated. It is also easy to represent this construction with our formalism.

4.5 Bei-construction

Bei-construction is a familiar Chinese pattern like the following:

noun phrase 1 '被' (bei) noun phrase 2 verb.

For example,

那隻鳥$_i$ 被(我)放走 t$_i$ 了。

(The bird was let go (by me).)

Bei-construction has disposal shown as below similar to ba-construction:

那個門被(我)踢了一個洞。
(That door was kicked (by me) and a hole is left.)

The rules in Section 4.1 (topic-comment structure) capture the above phenomena.

4.6 Pronoun Resolution

Binding Theory can be rephrased in the following procedures. Assume β is an anaphor, a pronominal, or an R-expression depending on which principle is used. Each element β may have two sets: set of possible pairs and set of impossible pairs. These two sets are denoted by *possible-pair* and *impossible-pair* respectively, and are defined in the following:

possible-pair(β)={ α | α can co-index with β },
impossible-pair(β)={ α | α cannot co-index with β }.

(Principle A) For an acceptable sentence, try to find some α such that α is in β's Governing Category and c-commands β. Each α that is outside of this range should not have a co-index relationship with β. This principle defines two sets for β. For example,

*李先生$_i$ 說 [$_S$ 你看見了自己$_i$]。

(* Mr. Lee$_i$ said [$_S$ that you saw yourself$_i$].)

possible-pair('自己')={'你'}
(possible-pair(self)={you}), and
impossible-pair('自己')={'李先生'}
(impossible-pair(self)={Mr. Lee}).

Both '你' (you) and '李先生' (Mr. Lee) c-command '自己' (self). The former is in the governing category of the reflexive '自己' (self), but the latter is outside. So the index assignment is not acceptable.

(Principle B) Those α s that are in the range of Governing Category and c-command β should not co-index with β. This principle just says which α s cannot be in the candidate set. However, we cannot determine whether those α s that are in its range and do not c-command β, co-index with β or not. If such an α co-indexes with β, it must satisfy other criteria, e.g. other binding principles, the same semantic feature, and so on. Thus, this principle says only the *impossible-pair*. For example,

[$_S$ 李先生$_i$ 看見了 他$_i$]。 ([$_S$ Mr. Lee$_i$ saw him$_i$].)

impossible-pair('他')={'李先生'}
(impossible-pair(him)={Mr. Lee}).

The phrase '李先生' (Mr. Lee) c-commands '他' (him), thus they cannot be co-indexed based on Principle B. Consider another example:

[$_S$ 他$_i$ 看見了 李先生$_i$]。 ([$_S$ He$_i$ saw Mr. Lee$_i$].)

The R-expression does not c-command the pronominal. According to Principle B, we have no way to determine their binding relationship. But if Principle C is applied, it can tell us the index assignment is wrong.

(Principle C) For any α where α c-commands β, α ought not to have co-index relationship with β. This principle says nothing for those α s that do not c-command β. A set *impossible-pair* is defined from this principle. For example,

*他$_i$ 說 [$_S$ 你看見了李先生$_i$]。

(* He$_i$ said [$_S$ that you saw Mr. Lee$_i$].)

impossible-pair('李先生')={'他','你'}
(impossible-pair(Mr. Lee)={he, you }).

The pronominal '他' (he) c-commands '李先生' (Mr. Lee), so they should have different indices.

Based on these three principles, a post-processing routine embedded in the parser is used to determine the co-index relationship between constituents from the parse tree. The algorithm is simple: Traverse the parse tree, generate the relations *possible-pair* and *impossible-pair*. If it is unknown up to now, a relation *unknown* is given temporarily. When a new relation *possible-pair* or *impossible-pair* is got, use it to check all the unknown relations. Retract the unknowns accordingly. Finally, assign the anaphors and pronominals suitable indices based on the relations *possible-pair* and *impossible-pair*.

5. Conclusion and Remarks

Many natural languages are flexible and context-sensitive. Mandarin Chinese is a famous example. It is difficult to capture the linguistic phenomena for these languages in computer. This paper adopts GB Theory to deal with this problem. According to GB Theory, the rule of 'move - α' moves anything anywhere, and the universal principles operate interactively to rule out the illegal movements. Thus, the only things should be declared in the grammars are:

(1) which phrases are the possible empty constituents,
(2) which positions are their possible empty sites,
(3) which positions are their possible landing sites,
(4) which phrasal categories are bounding nodes.

In such cases, a robust parser for natural languages can be designed. As an example, we represent many context-sensitive constructions in Mandarin Chinese, and do case marking and index assignment for Chinese sentences. An experimental Chinese parser is running under the environments: (1) Vax-11/785, (2) Quintus Prolog, (3) lexicon with about 200 words (about 33K bytes), and (4) about 150 production rules (about 112K bytes). Besides movement transformation, pronoun resolution is another index assignment. For well treatment of pronoun resolution, the syntactic knowledge is not enough. This is because the Binding Theory tells us much the *impossible pair*, but little the *possible pair*. Much more semantic information should be included.

Moreover, our GB approach is also useful when we would like to compose logical formulae from their syntactic counterparts. The idea is that the mapping between d-structure and s-structure, as well as between s-structure and logical form are treated in the similar way. The movement transformation between d-structure and s-structure tells us the relationship among verb and its accompanying arguments. The skeleton of the given verb is defined in the lexicon, and base-generated in the d-structure. For example,

'買'(Subject,Object) (buy(Subject,Object)).

The index assignment relates '書' (book) to the verb '買' (buy) in the following sentence:

有一本書$_i$ 每個學生都買 t_i 了。

(There is one book$_i$ that every student bought t_i.)

Because the variable of the type '書' (book) and the second argument of the template '買'(Subject, Object) (buy(Subject, Object)) should be the same in the logical form, the index (a unique integer) can be changed into a variable, say X. That is, they share the same variable shown below:

exist(X,'書'(X),forall(Y,'學生'(Y),'買'(Y,X)))
(exist(X,book(X),forall(Y,student(Y),buy(Y,X)))).

The formula tells us the SVO-SOV inversion in the logical form. This phenomenon can be added into our parser easily with our formalism. The details concerning the logical interpretation of Chinese sentences refer to /Chen 1989/.

References

Chen, H.H., I.P. Lin and C.P. Wu (1988) 'A New Design of Prolog-based Bottom-up Parsing System with Government-Binding Theory.' *Proceedings of the 12th International Conference on Computational Linguistics*, pp. 112-116.

Chen, H.H. (1989) 'The Logical Interpretation of Chinese Sentences.' *Computer Processing of Chinese and Oriental Languages* 4(2,3), pp. 171-184.

Chomsky, A.N. (1981) *Lectures on Government-Binding*. Foris Publication, Dordrecht, Holland.

Lin, I.P., S.F. Huang, H.H. Chen and K.W. Chui (1989) *The Study of the Knowledge Base in Mandarin Syntax (II)*. Project Report, Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, R.O.C.

Matsumoto, Y., H. Tanaka, *et al.* (1983) 'BUP: A Bottom-up Parser Embedded in Prolog.' *New Generation Computing* 1(2), pp. 145-158.

McCord, M.C. (1987) 'Natural Language Processing in Prolog.' In: Walker, A. (Editor) *A Logical Approach to Expert Systems and Natural Language Processing*. Addison-Wesley Publishing Company, Inc., pp. 291-402.

Pereira, F. (1981) 'Extraposition Grammars.' *American Journal of Computational Linguistics* 7(4), pp. 243-256.

Radford, A. (1981) *Transformation Syntax*. The Cambridge University Press.

Sells, P. (1985) *Lectures on Contemporary Syntactic Theories*. Stanford, Center for the Study of Language and Information.

Stabler, E.P., Jr. (1987) 'Restricting Logic Grammars with Government-Binding Theory.' *Computational Linguistics* 13(1-2), pp. 1-10.

Yang, Y. (1987) 'Combining Prediction, Syntactic Analysis and Semantic Analysis in Chinese Sentence Analysis.' *Proceedings of the 10th International Joint Conference on Artificial Intelligence*, pp. 679-681.