19/10/87

Formal Languages and Natural Languages

In trying to place natural languages in the hierarchy of formal languages described in HO1, we will be subject to two opposing considerations. On the one hand, we would like to be able to place them as high (weak) as possible, for several reasons:

↳ describe a set of strings

(1) The weaker the type of grammar, the stronger the claims we make about possible languages

(2) The weaker the type of grammar, the greater the efficiency of the parsing procedure

On the other hand, we may be forced to use a stronger grammar for various reasons:

↳ associate structure(s) with strings.

(1) To capture the bare facts about actual languages

(2) To provide for more perspicuous or elegant analyses, or to make for more compact grammars

clear in expression

Of course, perspicuity and elegance are in the eye of the beholder, but we will introduce some concepts here to clarify the idea.

Of itself, a formal description of a language, like a grammar or automaton, defines only a property of strings, i.e. set membership. However, the use of non-terminal symbols in a grammar that is sufficiently constrained as to rewrite only a single symbol imposes a hierarchical structure on strings that reflects the course of derivation.

The capacity of a generative grammar to describe a set of strings is called its **weak generative capacity.** For any given language, there are an infinite number of grammars that characterise it, i.e. of equivalent weak generative capacity.

The capacity of a generative grammar to associate structures with strings is called its **strong generative capacity.** This is not a notion that has a precise mathematical definition, but rather a linguistic notion that is motivated by considerations of ambiguity, paraphrase and the grammatical relations that hold between parts of a sentence. An elegant grammar is one whose strong generative capacity corresponds to our linguistic intuitions about such questions.

1. Finite State Languages

It is fairly obvious that finite state devices are inadequate in strong generative capacity to describe natural languages. A regular expression for simple subject-verb-object sentences in English, viz:

subject object

((det adj* noun) | pn) verb ((det adj* noun) | pn)

obviously fails to capture the common structure of the subject and object noun phrases. What is less obvious (and more contentious) is that they are also weakly inadequate, as demonstrated by an argument that goes as follows (reported in Gazdar and Pullum, 1985):

The set (1):

(1) {A white male (whom a white male hired)^n (hired)^n [illegible] another white male. | n > 0}

is the intersection of English with the regular set (2):

(2) A white male (whom a white male)* (hired)* another white male.

But (1) is not regular; and the regular sets are closed under intersection; hence English is not a regular language.

It is centre-embedding that causes a language to be non-regular, and centre-embedded constructions are notoriously difficult for humans to parse. Contrast the acceptability of the centre-embedded (3) with the right branching (4) and left-branching (5) structures:

(3) The bath the plumber the firm your mother recommended sent put in is cracked.

(4) On the table by the cupboard under the stairs in my house in London.

(5) My best friend's mother's hairdresser's Afghan hound's fur coat.

Nevertheless, there are good reasons for treating natural languages as of greater than regular power. First, there are several Central Sudanic languages in which centre-embedding appears to be commoner and more acceptable than in English. Secondly, it is difficult to justify imposing any strict bounds on the depth of centre-embedding, as acceptability seems to trail off, rather than cut-off, as it increases. Thirdly, if we were to impose an upper limit, say 4, on the depth of embedding, and to write a finite state grammar that would weakly characterise the string set thus obtained, such a grammar would be much larger, and less strongly adequate, than the equivalent context-free grammar. One approach that has been advocated is to explain the difficulty of processing by assuming that humans have a context-free grammar that is interpreted by a machine with a limited stack, and hence provably equivalent to a finite state grammar. The limit on the size of the stack is explicable in terms of limited human memory. Hence we have a distinction between a **competence** grammar, and its behaviour in **performance.**

2. Deterministic Context-free Languages

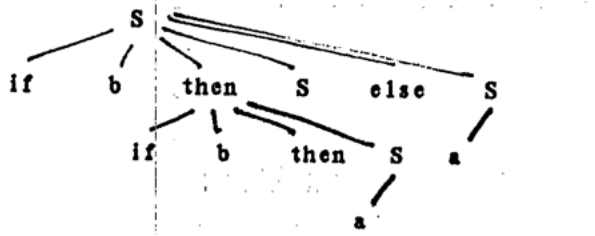
Finite state languages are not the only languages that can be recognised in linear time. A larger class of such languages is one that has been greatly studied in Computer Science, the **Deterministic Context-free Languages (DCFLs).** The standard argument that natural languages are not DCFLs is that they are ambiguous. However, in terms of string sets, the crucial question is not whether NLs are ambiguous, but whether they are inherently ambiguous. An inherently ambiguous language is one for which there is no unambiguous grammar. Note that a grammar for a language may be ambiguous, that is, associate more than one distinct derivation tree with a string in the language, without that language being inherently ambiguous:

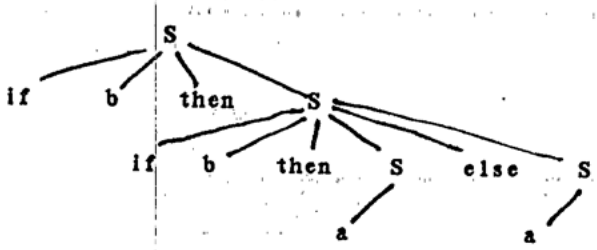
S → if b then S else S
S → if b then S
S → a

gives the following two derivation trees for the sentence:

if b then if b then a else a

19/10/97





(1) However, it is a simple matter to give an unambiguous grammar that generates the same language:

S1 → if b then S1
S1 → if b then S2 else S1
S1 → a

S2 → if b then S2 else S2
S2 → a

Note the following two results. It is undecidable whether an arbitrary CFG is ambiguous, and it is undecidable whether a given ambiguous CFG generates an inherently ambiguous language.

It is possible that a large subset of NL could be defined by means of an unambiguous grammar. The main problem with this is, as so often, a linguistic one - changing a grammar changes its strong generative capacity, and we would like an English string that is ambiguous in meaning to be associated with alternative derivations.

But again, this is not an absolute consideration. Consider compound nominals such as (6) and (7):

< nouns made from multiple nouns

(6) plastic baby pants

(7) left engine fuel pump suction line - aircraft maintenance manual.

A grammar that will give all possible structures is:

CN → CN CN
CN → N

CN - Compound Noun
N - Noun

The number of parses associated with a compound nominal by this grammar grows as the so-called Catalan series:
1 2 3 4 5 6 7 8
1 2 5 14 42 132 469 1430

giving (7), for instance, 132 parses. We might very well consider using a canonical grammar of the form:

19/10/87

CN → N*

for recognition purposes, and leave the precise structure to be determined by some semantic component.

3. Context-free Languages

So, arguments from ambiguity are at best a demonstration that DCFGs are not strongly adequate for describing NLs. Are there any phenomena that would demonstrate that CFGs are not weakly adequate. Notice that the standard dismissal on the basis of subject-verb agreement is not an argument against even strong adequacy. Given the finiteness of the set of values for the feature *number*, a context-free rule schema like:

S → NP[num=α] VP[num=α]

is a shorthand for a finite number of context-free rules, and seems to capture the facts of the matter in a perfectly general manner.

Other constructions that have been claimed to render NLs weakly greater than context-free power include the respectively construction in English, as in (8)

(8) Harry, Rod and David love Bev, Sandra and Mary respectively.

However, that this a matter of semantics, not syntax, is shown by (9)

(9) The three boys love the two girls and the gerbil respectively.

So that the deviance of (10)

(10) Harry and Rod love Bev, Sandra and Mary respectively.

is of the same type as that in (11)

(11) Our three main weapons are fear, surprise, ruthlessness and a fanatical devotion to the Pope.

Recently, however, certain constructions have been discovered that do appear to show that certain NLs are not even weakly context-free. The first example is that of subordinate infinitivals in Dutch and Swiss German. In Dutch, the translation of the English (12) is as in (13)

(12) ... that John saw Pete help Mary make the children swim.

(13) ... dat Jan Piet Marie de kinderen zag helpen make zwemmen.
maken

Now strictly, the grammatical sentences of Dutch displaying this phenomenon are of the form $a^n b^n$, and thus can be weakly generated by a context-free grammar. However, in Swiss German, certain verbs demand dative rather than accusative objects, and therefore a CFG is not even weakly adequate. We can demonstrate this by arranging for all accusative noun phrases to come before all dative noun phrases, (by intersection with a regular set). Then the language has the form $a^n b^m c^n d^m$, which cannot be generated by a context-free grammar.

Another example is from Bambara, a language spoken in Senegal. In Bambara, compound

19/10/87

nouns are formed by concatenation of noun stems, as in (14) - (16)

(14) wulu - dog

(15) wulu-filela - dog watcher

(16) wulu-filela-nyinila - dog watcher hunter

One can also form a noun 'N - o - N' meaning 'whatever N', so the Bambara vocabulary also contains words of the form (17) - (19)

(17) wulu-o-wulu - whatever dog

(18) wulu-filela-o-wulu-filela - whatever dog watcher

(19) wulu-filela-nyinila-o-wulu-filela-nyinila - whatever dog watcher hunter

That is, Bambara has constructions of the form $\{w\text{-}o\text{-}w \mid w \in \Sigma^* \}$, which is not a context-free language.

4. Indexed Languages

It thus appears that NLs fall outside the CFLs. Recently there has been considerable interest in exploring the space between type 2 and type 1 formalisms, in the hope that a more restricted class of grammars than the context-sensitive (type 1) ones may be adequate. One formulation that appears to be adequate to handle the non-CF cases above, while still being capable of generating only a proper subset of type 1 languages, are the **indexed grammars.** It is not possible to give a characterisation of indexed grammars in terms of the form of a rewrite rule using only atomic symbols. Rather, non-terminal symbols are augmented with a stack, and rules of the following forms are allowed:

$A[..] \rightarrow \alpha[..]$
$A[..] \rightarrow B[i,..]$
$A[i,..] \rightarrow \alpha[..]$
$A[..] \rightarrow \alpha[]\ B[..]\ \beta[]$
$A[..] \rightarrow \alpha[]\ B[i,..]\ \beta[]$
$A[i,..] \rightarrow \alpha[]\ B[..]\ \beta[]$

where A and B are non-terminals, and α, β are strings of terminals and non-terminals.

Then we have a very simple formulation of the language $a^n b^n c^n$:

S[..] → a A[z,..]
A[..] → a A[a,..]
A[..] → B[..]
B[a,..] → b B[..] c
B[z,..] → bc

A[..] → b A[b,..]
A[..] → c A[c,..]
A[..] → B[..]
B[a,..] → B[..] a
B[b,..] → B[..] b
B[c,..] → B[..] c
B[x,..] → a
B[y,..] → b
B[z,..] → c

and of the language $\{ww \mid w \in (a,b,c)^+\}$

S[..] → a A[x,..]
S[..] → b A[y,..]
S[..] → c A[z,..]
A[..] → a A[a,..]

$a^n b^n c^n d^m$