CMSC35000-1 Introduction to Artificial Intelligence **Winter 2007**

Lecture 14: 02/22/2007

Lecturer: Partha Niyogi *Scribe: Zhandos Yessenbayev*

14.1 Context Free Grammar

Let N = {S, X, Y, Z, ...} be a set of non-terminals with distinguished non-terminal S, and T = Σ be a set of words (terminals).
A *Context Free Grammar* is a formal grammar with a finite set of the production rules of the form A $\rightarrow \phi$, where A $\in$ N and $\phi \in \{NUT\}^*$
A language defined by this grammar consists of all possible strings that can be generated by the rules of the grammar.

e.g.: N = $\{S\}$, T = $\{a, b\}$
S $\rightarrow$ aSb
S $\rightarrow$ ab
This grammar produces the strings of the form $a^k b^k$

To formalize the set of rules, we require it to be in the *Chomsky Normal Form*:
A $\rightarrow$ BC
A $\rightarrow$ a,
where A,B,C - non-terminals, and a - teminal symbol. For instance, we can represent the rules in the example above as follows:
S $\rightarrow$ AB
A $\rightarrow$ a
B $\rightarrow$ aSB'
B'$\rightarrow$ b
So, in general, a rule of the form A $\rightarrow \phi$ with $\phi = \phi_1\phi_2$ can be represented by the set of the new rules
A $\rightarrow$ BC
B $\rightarrow \phi_1$
C $\rightarrow \phi_2$
Thus, we can represent recursively any set of rules as a set of rules in a Chomsky normal form by introducing new non-terminal elements. From now on we can assume that all grammars are given in Chomsky normal form
So, given a grammar G and a string S, the next question is "Does there exist a unique parse tree for this sentence?" Since, in general, the answer is "No", then "How many parse trees are there if there is any?" Finding all parse trees is interesting, because we can not only verify and understand the grammatical structure of the sentence but also see if it is ambiguous. For example, consider a sentence like "I saw the man with the binocular". It is not clear whether the phrase "with the binocular" describes "the man" or refers to the mean by which "the man" was seen, but both cases are grammatically correct.

14.1.1 X-bar Theory

Consider a grammar that have three types of phrases:
  Prepositional phrase: with a spoon, over the table
  Noun phrase: the man with the binocular
  Verb phrase: ate the apple, picked the ball
for which the rules are defined as follows:
  XP $\to \overline{X}$ YP
  $\overline{X} \to \overline{X}$ YP
  $\overline{X} \to$ X
Then the claim of this theory is that the head of the X-phrase is $\overline{X}$ (the head of the phrase).
According to this, the sentence
  He ran from there with his money
can be parsed like
  He (ran (from there) (with his money))

If this is the case for the English language, which has SVO (Subject+Verb+Object) structure, it may not be for the other languages, moreover, the sentence itself may be grammatically incorrect.
So, returning to the question whether the given sentence S has a parse tree (equivalently, whether it is grammatically correct) or not, let us estimate the complexity of this question on the following example.

Suppose we have the rules
  S $\to$ Z Z
  Z $\to$ Z Z
  Z $\to$ AB
  B $\to$ CD
  B $\to$ b
  A $\to$ C'D'
  A $\to$ a
  C $\to$ b
  D $\to$ b
  C' $\to$ a
  D' $\to$ b

Clearly, for this grammar the string ab^2 can be obtained in two different way from Z, for instance, either
S $\to$ Z Z, Z $\to$ A B, A $\to$ a, B $\to$ C D, C $\to$ b, D $\to$ b
or
S $\to$ Z Z, Z $\to$ A B, A $\to$ C'D', B $\to$ b, C' $\to$ a, D' $\to$ b

But then it turns out that the string $ab^2ab^2...ab^2$ with k ab^2 substrings in it can have at least 2^k parse trees, since each ab^2 can be obtained from a single Z. And this, consequently, means that it is a quite difficult problem to find a proper parse tree among the exponential number of possible trees for the given sentence.

14.1.2 A Chart Parse

Let S = $\alpha_1\alpha_2...\alpha_n$, where $\alpha_i \in \Sigma$ and $\mid S \mid= n$. Suppose we are given a grammar with M production rules in Chomsky normal form, i.e. the rules are of the form:

A → BC
A → a
where A,B,C $\in \{A_1, A_2, ..., A_L\}$ - a set of all non-terminals, a - terminal symbol.
Then for each non-terminal A_k we define a matrix of size nxn such that $A_k[i,j] = 1 \Leftrightarrow A_k$ covers the subsequence $\alpha_i \alpha_{i+1} ... \alpha_j$ (i $\leq$ j). So, ultimately, we want to verify if S[i,j]=1.
Having said this, we begin building all the matrices recursively as follows:
1) Initialize all the matrises A_k with zeros: $A_k[i,j] = 0 \; \forall i,j,k$;
2) $A_k[i,i] = 1$, if there exists a rule of the form $A_k \rightarrow \alpha_i$, $\forall i, k$;

3) Next we run:

```
for all $k \in \{1, 2, ..., L\}$
    for all $i \in \{1, 2, ..., n\}$
        for all $j \geq i$
            $A_k[i,j] = 1 \Leftrightarrow$ if there exists a rule $A_k \rightarrow BC$ and $i \leq l \leq j$ s.t. $B[i, l-1] = 1$ and $C[l-1, j] = 1$
        end
    end
end
```

So, after this loop we fill up all the upper triangles of each of the matrices A_k. The complexity of the algorithm, as it is easy to see, is of order $O(L \cdot M \cdot n^3)$, that is, the algorithm runs in polynomial time.

14.1.3 A Stochastic Grammar

Another way to approach the problem is to assign a probability distribution P on a given language L $\subset \Sigma^*$ s.t. $\Sigma P(S) = 1$, for all $S \in L$. The rationale of this is that even though a sentence may be perfectly grammatical, it is more unlikely to appear in the common language (e.g. archaisms).
On the other hand, rather than assigning a probability to each sentence in the language, which is impractical because of infinite size of L, we can assign a probability distribution on a set of rules that define the grammar:
$A_1 \rightarrow B_1 C_1$ (p_1)
$A_2 \rightarrow B_2 C_2$ (p_2)
...
$A_m \rightarrow B_m C_m$ (p_m)
where $p_i \geq 0$ for all $i \in \{1, 2, ..., m\}$ and $\Sigma p_i = 1$.
Given this, it is clear that if S has a parse tree then

$$P(parse) = \prod_{node \in parse} p_i(node),$$

and in case of the existence of several parse trees we get:

$$P(S) = \sum_{\forall parse of S} P(parse),$$

Example:
Let G be a grammar with rules:
S → SS (p)
S → a (1-p)

It is obvious that this grammar consists of the strings $a^i (1 \leq i)$, then as we said before

$$\sum_{i=1}^{\infty} P(a^i) = 1,$$

which holds only for $p > \frac{1}{2}$. This fact can be verified from the following recurrence relation:
$S_{h+1} = (1-p) + pS_h, S_1 = 1-p$
where S_h is the probability of generating a tree of height $\leq h$.