Membership Testing in Context-Free Languages

CSE 322: Introduction to Formal Models in Computer Science
March 5, 2002

1 Introduction

Given a context-free language L and a string w, it is not clear how to write an efficient (deterministic) algorithm that decides whether or not $w \in L$. If L is represented by a context-free grammar, there may be many choices of rules that could be used to expand the leftmost nonterminal symbol, and the naive approach of trying them all at each step of the derivation leads to an algorithm that may need time exponential in the length of the derivation. If instead L is represented by a pushdown automaton, the nondeterminism may provide many choices of transitions that apply to any given configuration, and the naive approach of trying them all at each step leads to an algorithm that may need time exponential in the number of moves.

The purpose of this note is to provide an efficient algorithm for this problem. This algorithm was discovered independently by Cocke, Kasami, and Younger, c. 1965, and is also discussed in Theorem 7.14 of the text.

2 An Algorithm for Deciding Membership

Assume without loss of generality that L is specified by a context-free grammar $G = (V, \Sigma, R, S)$ that is in Chomsky normal form. Let $w = a_1 a_2 \cdots a_n$, where $a_i \in \Sigma$ for $1 \leq i \leq n$. For $1 \leq i \leq j \leq n$, define

$$M_{i,j} = \{A \in V \mid A \Rightarrow_G^* a_i a_{i+1} \cdots a_j\}.$$

The algorithm presented in Figure 1 computes all these sets, starting with $M_{i,i}, M_{i,i+1}, \ldots$, until it finally computes $M_{1,n}$. At that point, $w \in L(G)$ if and only if $S \in M_{1,n}$. The method used by this algorithm is called "dynamic programming".

3 Example

Consider the grammar $G = (V, \Sigma, R, S)$, where

$$\begin{aligned}
V &= \{A, B, C, D, S, T, X, a, b\}, \\
\Sigma &= \{a, b\}, \text{ and} \\
R &= \{S \to TT,\ S \to AC,\ T \to AC,\ T \to DA,\ T \to AB,\ T \to BA, \\
&\quad\ C \to XB,\ D \to BX,\ X \to TT,\ X \to AB,\ X \to BA,\ A \to a,\ B \to b\}.
\end{aligned}$$

Notice that G is in Chomsky normal form, so that the algorithm can be applied to it directly.

```
comment: Determine if a_1 a_2 ··· a_n is generated by G = (V, Σ, R, S) ;

comment: Compute M_{i,i} ;
for i from 1 to n do
  begin
    M_{i,i} ← φ ;
    for all A ∈ V do
      if (A → a_i) ∈ R
        then add A to M_{i,i} ;
  end ;

comment: Compute all other sets M_{i,j} ;
for d from 1 to n − 1 do
  for i from 1 to n − d do
    begin
      j ← i + d ;
      M_{i,j} ← φ ;
      for k from i to j − 1 do
        for all A ∈ V do
          if (A → BC) ∈ R for some B ∈ M_{i,k} and C ∈ M_{k+1,j}
            then add A to M_{i,j} ;
    end ;

if S ∈ M_{1,n} then return "w ∈ L(G)" else return "w ∉ L(G)" ;
```

Figure 1: Algorithm to determine if $a_1 a_2 \cdots a_n$ is generated by $G = (V, \Sigma, R, S)$

For the input $w = baabab$, the algorithm computes the sets $M_{i,j}$ for $i = j$ as follows:

	j=1	j=2	j=3	j=4	j=5	j=6
$i=1$	$\{B\}$					
$i=2$		$\{A\}$				
$i=3$			$\{A\}$			
$i=4$				$\{B\}$		
$i=5$					$\{A\}$	
$i=6$						$\{B\}$

Successive iterations of the "**for** d ..." loop produce the following sets $M_{i,j}$.

$d = 1:$

	j=1	j=2	j=3	j=4	j=5	j=6
$i=1$	$\{B\}$	$\{T,X\}$				
$i=2$		$\{A\}$	ϕ			
$i=3$			$\{A\}$	$\{T,X\}$		
$i=4$				$\{B\}$	$\{T,X\}$	
$i=5$					$\{A\}$	$\{T,X\}$
$i=6$						$\{B\}$

$d = 2:$

	j=1	j=2	j=3	j=4	j=5	j=6
$i=1$	$\{B\}$	$\{T,X\}$	ϕ			
$i=2$		$\{A\}$	ϕ	ϕ		
$i=3$			$\{A\}$	$\{T,X\}$	ϕ	
$i=4$				$\{B\}$	$\{T,X\}$	$\{C,D\}$
$i=5$					$\{A\}$	$\{T,X\}$
$i=6$						$\{B\}$

$d = 3:$

	j=1	j=2	j=3	j=4	j=5	j=6
$i=1$	$\{B\}$	$\{T,X\}$	ϕ	$\{S,X\}$		
$i=2$		$\{A\}$	ϕ	ϕ	ϕ	
$i=3$			$\{A\}$	$\{T,X\}$	ϕ	$\{S,T,X\}$
$i=4$				$\{B\}$	$\{T,X\}$	$\{C,D\}$
$i=5$					$\{A\}$	$\{T,X\}$
$i=6$						$\{B\}$

$d = 4:$

	j=1	j=2	j=3	j=4	j=5	j=6
$i=1$	$\{B\}$	$\{T,X\}$	ϕ	$\{S,X\}$	ϕ	
$i=2$		$\{A\}$	ϕ	ϕ	ϕ	ϕ
$i=3$			$\{A\}$	$\{T,X\}$	ϕ	$\{S,T,X\}$
$i=4$				$\{B\}$	$\{T,X\}$	$\{C,D\}$
$i=5$					$\{A\}$	$\{T,X\}$
$i=6$						$\{B\}$

$d = 5:$

	j=1	j=2	j=3	j=4	j=5	j=6
$i = 1$	$\{B\}$	$\{T, X\}$	ϕ	$\{S, X\}$	ϕ	$\{S, X\}$
$i = 2$		$\{A\}$	ϕ	ϕ	ϕ	ϕ
$i = 3$			$\{A\}$	$\{T, X\}$	ϕ	$\{S, T, X\}$
$i = 4$				$\{B\}$	$\{T, X\}$	$\{C, D\}$
$i = 5$					$\{A\}$	$\{T, X\}$
$i = 6$						$\{B\}$

As an example of how the algorithm proceeds, T is added to $M_{3,6}$ as a result of $d = 3$, $i = 3$, $k = 3$, and the rule $T \to AC$:

	j=1	j=2	j=3	j=4	j=5	j=6
$i = 1$						
$i = 2$						
$i = 3$			A			T
$i = 4$						C
$i = 5$						
$i = 6$						

On the other hand, X is added to the same set $M_{3,6}$ as a result of $d = 3$, $i = 3$, $k = 4$, and the rule $X \to TT$:

	j=1	j=2	j=3	j=4	j=5	j=6
$i = 1$						
$i = 2$						
$i = 3$				T		X
$i = 4$						
$i = 5$						T
$i = 6$						