CS 611: Theory of Computation

Hongmin Li

Department of Computer Science
California State University, East Bay

Emptiness Problem

Given a CFG G with start symbol S, is $L(G)$ empty?

Emptiness Problem

Given a CFG G with start symbol S, is $L(G)$ empty?
Solution: Check if the start symbol S is generating.

Emptiness Problem

Given a CFG G with start symbol S, is $L(G)$ empty?
Solution: Check if the start symbol S is generating. How long does that take?

Determining generating symbols

Algorithm

Gen = $\{\}$
for every rule $A \to x$ where $x \in \Sigma^*$
    Gen = Gen $\cup$ $\{A\}$
repeat
    for every rule $A \to \gamma$
        if all variables in γ are generating then
            Gen = Gen $\cup$ $\{A\}$
until Gen does not change

Determining generating symbols

Algorithm

```
Gen = {}
for every rule A → x where x ∈ Σ*
    Gen = Gen ∪ {A}
repeat
    for every rule A → γ
        if all variables in γ are generating then
            Gen = Gen ∪ {A}
until Gen does not change
```

- Both for-loops take $O(n)$ time where $n = |G|$.

Determining generating symbols

Algorithm

```
Gen = {}
for every rule A → x where x ∈ Σ*
    Gen = Gen ∪ {A}
repeat
    for every rule A → γ
        if all variables in γ are generating then
            Gen = Gen ∪ {A}
until Gen does not change
```

- Both for-loops take $O(n)$ time where $n = |G|$.
- Each iteration of repeat-until loop discovers a new variable. So number of iterations is $O(n)$. And total is $O(n^2)$.

Membership Problem

Given a CFG $G = (V, \Sigma, R, S)$ in Chomsky Normal Form, and a string $w \in \Sigma^*$, is $w \in L(G)$?

Membership Problem

Given a CFG $G = (V, \Sigma, R, S)$ in Chomsky Normal Form, and a string $w \in \Sigma^*$, is $w \in L(G)$?
Central question in parsing.

"Simple" Solution

"Simple" Solution

- Let $|w| = n$. Since G is in Chomsky Normal Form, w has a parse tree of size $2n-1$ iff $w \in L(G)$

"Simple" Solution

- Let $|w| = n$. Since G is in Chomsky Normal Form, w has a parse tree of size $2n - 1$ iff $w \in L(G)$
- Construct all possible parse (binary) trees and check if any of them is a valid parse tree for w

"Simple" Solution

- Let $|w| = n$. Since G is in Chomsky Normal Form, w has a parse tree of size $2n-1$ iff $w \in L(G)$
- Construct all possible parse (binary) trees and check if any of them is a valid parse tree for w
- Number of parse trees of size $2n-1$ is k^{2n-1} where k is the number of variables in G. So algorithm is exponential in n!

"Simple" Solution

- Let $|w| = n$. Since G is in Chomsky Normal Form, w has a parse tree of size $2n - 1$ iff $w \in L(G)$
- Construct all possible parse (binary) trees and check if any of them is a valid parse tree for w
- Number of parse trees of size $2n - 1$ is k^{2n-1} where k is the number of variables in G. So algorithm is exponential in n!
- We will see an algorithm that runs in $O(n^3)$ time (the constant will depend on k).

First Ideas

Notation

Suppose $w = w_1 w_2 \cdots w_n$, where $w_i \in \Sigma$. Let $w_{i,j}$ denote the substring of w starting at position i of length j. Thus, $w_{i,j} = w_i w_{i+1} \cdots w_{i+j-1}$

First Ideas

Notation

Suppose $w = w_1 w_2 \cdots w_n$, where $w_i \in \Sigma$. Let $w_{i,j}$ denote the substring of w starting at position i of length j. Thus, $w_{i,j} = w_i w_{i+1} \cdots w_{i+j-1}$

Main Idea

For every $A \in V$, and every $i \leq n$, $j \leq n+1-i$, we will determine if $A \stackrel{*}{\Rightarrow} w_{i,j}$.

First Ideas

Notation

Suppose $w = w_1 w_2 \cdots w_n$, where $w_i \in \Sigma$. Let $w_{i,j}$ denote the substring of w starting at position i of length j. Thus, $w_{i,j} = w_i w_{i+1} \cdots w_{i+j-1}$

Main Idea

For every $A \in V$, and every $i \leq n$, $j \leq n+1-i$, we will determine if $A \stackrel{*}{\Rightarrow} w_{i,j}$.

Now, $w \in L(G)$ iff $S \stackrel{*}{\Rightarrow} w_{1,n} = w$; thus, we will solve the membership problem.

First Ideas

Notation

Suppose $w = w_1 w_2 \cdots w_n$, where $w_i \in \Sigma$. Let $w_{i,j}$ denote the substring of w starting at position i of length j. Thus, $w_{i,j} = w_i w_{i+1} \cdots w_{i+j-1}$

Main Idea

For every $A \in V$, and every $i \leq n$, $j \leq n + 1 - i$, we will determine if $A \stackrel{*}{\Rightarrow} w_{i,j}$.

Now, $w \in L(G)$ iff $S \stackrel{*}{\Rightarrow} w_{1,n} = w$; thus, we will solve the membership problem.

How do we determine if $A \stackrel{*}{\Rightarrow} w_{i,j}$ for every A, i, j?

Base Case

Substrings of length 1

Observation

For any A, i, $A \stackrel{*}{\Rightarrow} w_{i,1}$ iff $A \rightarrow w_{i,1}$ is a rule.

Base Case

Substrings of length 1

Observation

For any A, i, $A \stackrel{*}{\Rightarrow} w_{i,1}$ iff $A \rightarrow w_{i,1}$ is a rule.

- Since G is in Chomsky Normal Form, G does not have any ϵ-rules, nor any unit rules.

Base Case

Substrings of length 1

Observation

For any A, i, $A \overset{*}{\Rightarrow} w_{i,1}$ iff $A \rightarrow w_{i,1}$ is a rule.

- Since G is in Chomsky Normal Form, G does not have any ϵ-rules, nor any unit rules.

Thus, for each A and i, one can determine if $A \overset{*}{\Rightarrow} w_{i,1}$.

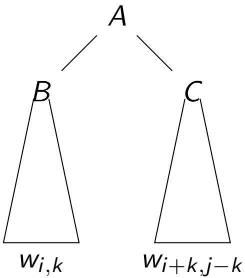
Inductive Step

Longer substrings

Suppose for every variable X and every $w_{i,\ell}$ ($\ell < j$) we have determined if $X \overset{*}{\Rightarrow} w_{i,\ell}$

Inductive Step

Longer substrings

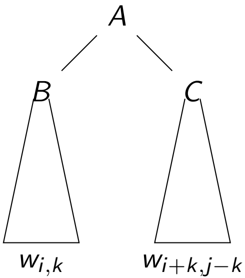


Suppose for every variable X and every $w_{i,\ell}$ ($\ell < j$) we have determined if $X \stackrel{*}{\Rightarrow} w_{i,\ell}$

- $A \stackrel{*}{\Rightarrow} w_{i,j}$ iff there are variables B and C and some $k < j$ such that $A \rightarrow BC$ is a rule, and $B \stackrel{*}{\Rightarrow} w_{i,k}$ and $C \stackrel{*}{\Rightarrow} w_{i+k,j-k}$

Inductive Step

Longer substrings



Suppose for every variable X and every $w_{i,\ell}$ ($\ell < j$) we have determined if $X \stackrel{*}{\Rightarrow} w_{i,\ell}$

- $A \stackrel{*}{\Rightarrow} w_{i,j}$ iff there are variables B and C and some $k < j$ such that $A \rightarrow BC$ is a rule, and $B \stackrel{*}{\Rightarrow} w_{i,k}$ and $C \stackrel{*}{\Rightarrow} w_{i+k,j-k}$
- Since k and $j - k$ are both less than j, we can inductively determine if $A \stackrel{*}{\Rightarrow} w_{i,j}$.

Cocke-Younger-Kasami (CYK) Algorithm

Algorithm maintains $X_{i,j} = \{A \mid A \stackrel{*}{\Rightarrow} w_{i,j}\}$.

```
Initialize:  X_{i,1} = {A | A → w_{i,1}}
for j = 2 to n do
    for i = 1 to n − j + 1 do
        X_{i,j} = ∅
        for k = 1 to j − 1 do
            X_{i,j} = X_{i,j} ∪ {A | A → BC, B ∈ X_{i,k}, C ∈ X_{i+k,j−k}}
```

Cocke-Younger-Kasami (CYK) Algorithm

Algorithm maintains $X_{i,j} = \{A \mid A \overset{*}{\Rightarrow} w_{i,j}\}$.

```
Initialize:  X_{i,1} = {A | A → w_{i,1}}
for j = 2 to n do
    for i = 1 to n − j + 1 do
        X_{i,j} = ∅
        for k = 1 to j − 1 do
            X_{i,j} = X_{i,j} ∪ {A | A → BC, B ∈ X_{i,k}, C ∈ X_{i+k,j−k}}
```

Correctness: After each iteration of the outermost loop, $X_{i,j}$ contains exactly the set of variables A that can derive $w_{i,j}$, for each i.

Cocke-Younger-Kasami (CYK) Algorithm

Algorithm maintains $X_{i,j} = \{A \mid A \stackrel{*}{\Rightarrow} w_{i,j}\}$.

`Initialize:` $X_{i,1} = \{A \mid A \rightarrow w_{i,1}\}$
for j `= 2 to` n **do**
  for i `= 1 to` $n-j+1$ **do**
    $X_{i,j} = \emptyset$
    for k `= 1 to` $j-1$ **do**
      $X_{i,j} = X_{i,j} \cup \{A \mid A \rightarrow BC,\ B \in X_{i,k},\ C \in X_{i+k,j-k}\}$

Correctness: After each iteration of the outermost loop, $X_{i,j}$ contains exactly the set of variables A that can derive $w_{i,j}$, for each i.Time $= O(n^3)$.

Example

Example

Consider grammar
$S \rightarrow AB \mid BC,\ A \rightarrow BA \mid a,\ B \rightarrow CC \mid b,\ C \rightarrow AB \mid a$ Let $w = baaba$. The sets $X_{i,j} = \{A \mid A \overset{*}{\Rightarrow} w_{i,j}\}$:

Example

Example

Consider grammar
$S \to AB \mid BC,\ A \to BA \mid a,\ B \to CC \mid b,\ C \to AB \mid a$ Let $w = baaba$. The sets $X_{i,j} = \{A \mid A \stackrel{*}{\Rightarrow} w_{i,j}\}$:

j/i	1	2	3	4	5
5					
4					
3					
2					
1	$\{B\}$	$\{A, C\}$	$\{A, C\}$	$\{B\}$	$\{A, C\}$
	b	a	a	b	a

Example

Example

Consider grammar
$S \rightarrow AB \mid BC,\ A \rightarrow BA \mid a,\ B \rightarrow CC \mid b,\ C \rightarrow AB \mid a$ Let $w = baaba$. The sets $X_{i,j} = \{A \mid A \overset{*}{\Rightarrow} w_{i,j}\}$:

j/i	1	2	3	4	5
5					
4					
3					
2	$\{S, A\}$	$\{B\}$	$\{S, C\}$	$\{S, A\}$	
1	$\{B\}$	$\{A, C\}$	$\{A, C\}$	$\{B\}$	$\{A, C\}$
	b	a	a	b	a

Example

Example

Consider grammar
$S \to AB \mid BC,\ A \to BA \mid a,\ B \to CC \mid b,\ C \to AB \mid a$ Let $w = baaba$. The sets $X_{i,j} = \{A \mid A \stackrel{*}{\Rightarrow} w_{i,j}\}$:

j/i	1	2	3	4	5
5					
4					
3	$\emptyset$	$\{B\}$	$\{B\}$		
2	$\{S, A\}$	$\{B\}$	$\{S, C\}$	$\{S, A\}$	
1	$\{B\}$	$\{A, C\}$	$\{A, C\}$	$\{B\}$	$\{A, C\}$
	b	a	a	b	a

Example

Example

Consider grammar
$S \to AB \mid BC,\ A \to BA \mid a,\ B \to CC \mid b,\ C \to AB \mid a$ Let $w = baaba$. The sets $X_{i,j} = \{A \mid A \overset{*}{\Rightarrow} w_{i,j}\}$:

j/i	1	2	3	4	5
5					
4	$\emptyset$	$\{S, A, C\}$			
3	$\emptyset$	$\{B\}$	$\{B\}$		
2	$\{S, A\}$	$\{B\}$	$\{S, C\}$	$\{S, A\}$	
1	$\{B\}$	$\{A, C\}$	$\{A, C\}$	$\{B\}$	$\{A, C\}$
	b	a	a	b	a

Example

Example

Consider grammar
$S \rightarrow AB \mid BC,\ A \rightarrow BA \mid a,\ B \rightarrow CC \mid b,\ C \rightarrow AB \mid a$ Let $w = baaba$. The sets $X_{i,j} = \{A \mid A \overset{*}{\Rightarrow} w_{i,j}\}$:

j/i	1	2	3	4	5
5	$\{S, A, C\}$				
4	$\emptyset$	$\{S, A, C\}$			
3	$\emptyset$	$\{B\}$	$\{B\}$		
2	$\{S, A\}$	$\{B\}$	$\{S, C\}$	$\{S, A\}$	
1	$\{B\}$	$\{A, C\}$	$\{A, C\}$	$\{B\}$	$\{A, C\}$
	b	a	a	b	a

More Decision Problems

Given a CFGs G_1 and G_2

More Decision Problems

Given a CFGs G_1 and G_2

- Is $L(G_1) = \Sigma^*$?

More Decision Problems

Given a CFGs G_1 and G_2

- Is $L(G_1) = \Sigma^*$?
- Is $L(G_1) \cap L(G_2) = \emptyset$?

More Decision Problems

Given a CFGs G_1 and G_2

- Is $L(G_1) = \Sigma^*$?
- Is $L(G_1) \cap L(G_2) = \emptyset$?
- Is $L(G_1) = L(G_2)$?

More Decision Problems

Given a CFGs G_1 and G_2

- Is $L(G_1) = \Sigma^*$?
- Is $L(G_1) \cap L(G_2) = \emptyset$?
- Is $L(G_1) = L(G_2)$?
- Is G_1 ambiguous?

More Decision Problems

Given a CFGs G_1 and G_2

- Is $L(G_1) = \Sigma^*$?
- Is $L(G_1) \cap L(G_2) = \emptyset$?
- Is $L(G_1) = L(G_2)$?
- Is G_1 ambiguous?
- Is $L(G_1)$ inherently ambiguous?

More Decision Problems

Given a CFGs G_1 and G_2

- Is $L(G_1) = \Sigma^*$?
- Is $L(G_1) \cap L(G_2) = \emptyset$?
- Is $L(G_1) = L(G_2)$?
- Is G_1 ambiguous?
- Is $L(G_1)$ inherently ambiguous?

All these problems are undecidable.