Lecture 6
Simplification of Context-Free Grammars and Normal Forms

COSE215: Theory of Computation

Seunghoon Woo

Fall 2023

Review: Chomsky normal forms

- **In lecture 6-1 (CFG simplification)**
  - λ-productions can be eliminated only if S is not nullable
- **In lecture 6-2 (CNF step (1))**
  - We need to eliminate λ-productions even S is nullable
- **Is this a contradiction?**

Review: Chomsky normal forms

- **In lecture 6-1 (CFG simplification)**
  - λ-productions can be eliminated only if S is not nullable
  - Here, the goal is eliminating all λ-productions from the grammar
  - If S contains λ-production, it is infeasible to remove λ-production from the grammar
    - Because $S \rightarrow \lambda$ should be contained in the grammar (production rules)
    - If we eliminate it, the generated grammar is not equivalent to the original grammar

Review: Chomsky normal forms

- **In lecture 6-2 (CNF step (1))**
  - We need to eliminate λ-productions even S is nullable
  - Here, the goal is to generate CNF
  - If S is nullable?
    - ❖ First ignore this and remove all λ-productions
    - ❖ This is considered in the last step of CNF conversion steps
    - ❖ Therefore, S in the final CNF is still nullable

Review: Chomsky normal forms

- **In summary, if S is nullable,**
  1. We cannot eliminate all λ-productions in CFG simplification
  2. We can still convert a CFG to CNF by first eliminating all λ-productions and adding $S \rightarrow \lambda$ at the very end of the conversion step

Contents

- **A membership algorithm for CFG**

A membership algorithm for CFG

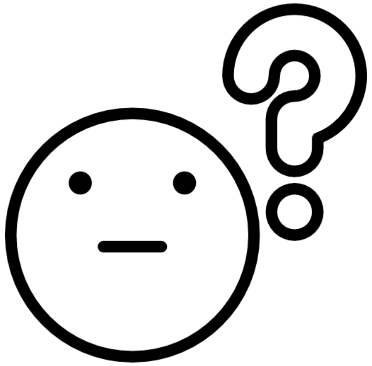
- **The problem of determining whether a string belongs to the language generated by a given CFG**
  - Example
    - Determine whether the string $w = aabbb$ is in the language generated by the grammar
      - $S \rightarrow AB$
      - $A \rightarrow BB \mid a$
      - $B \rightarrow AB \mid b$

A membership algorithm for CFG

- **CYK algorithm**
  - A parsing algorithm by J. Cocke, D.H. Younger, and T. Kasami
  - By using the CYK algorithm, we can address the membership problem
  - This can only be used when the CFG is in **CNF**

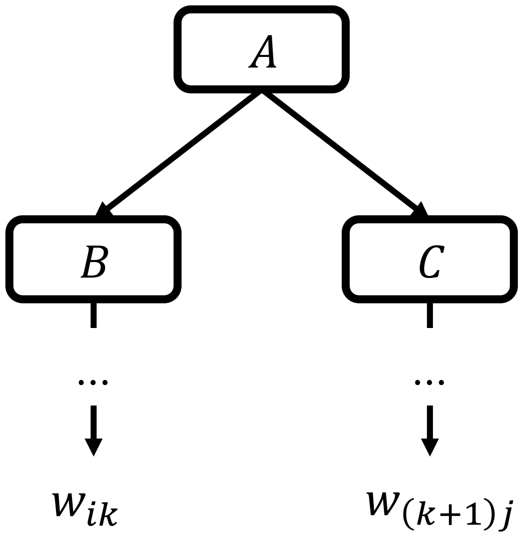
A membership algorithm for CFG

- **CYK algorithm: core idea**
  - $G = (V, T, S, P)$ in CNF
  - Assume an input string as $w = a_1 a_2 \ldots a_n$
    - ❖ We define substrings: $w_{ij} = a_i \ldots a_j$
    - ❖ Subsets of V: $V_{ij} = \{ A \in V : A \overset{*}{\Rightarrow} w_{ij} \}$
    - ❖ $V_{ij} = \cup \{ A : A \rightarrow BC, with\ B \in V_{ik},\ C \in V_{(k+1)j} \}$
  - Then, $w \in L(G)$ if and only if $S \in V_{1n}$



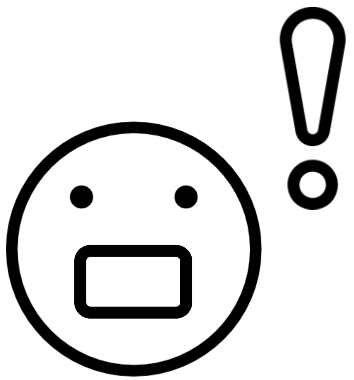
A membership algorithm for CFG

- **CYK algorithm: core idea**
  - A variable that can derive a string w_{ij} has the following production rule
    - ❖Concatenation of a variable that can derive w_{ik} and a variable that can derive $w_{(k+1)j}$



➡

$$A \in V_{ij}$$
$$(A \overset{*}{\Rightarrow} BC \overset{*}{\Rightarrow} w_{ij})$$



A membership algorithm for CFG

- **CYK algorithm example**
  - Determine whether the string $w = aabbb$ is in the language generated by the grammar
    - ❖ $S \rightarrow AB,\ A \rightarrow BB \mid a,\ B \rightarrow AB \mid b$

A membership algorithm for CFG

- **CYK algorithm example**
  - Determine whether the string $w = aabbb$ is in the language generated by the grammar
    - ❖ $S \rightarrow AB,\ A \rightarrow BB \mid a,\ B \rightarrow AB \mid b$
  - $V_{11} = \{A\}, V_{22} = \{A\}, V_{33} = \{B\}, V_{44} = \{B\}, V_{55} = \{B\}$

A membership algorithm for CFG

- **CYK algorithm example**
  - Determine whether the string $w = aabbb$ is in the language generated by the grammar
    - ❖ $S \rightarrow AB,\ A \rightarrow BB \mid a,\ B \rightarrow AB \mid b$
  - $V_{11} = \{A\}, V_{22} = \{A\}, V_{33} = \{B\}, V_{44} = \{B\}, V_{55} = \{B\}$
  - $V_{12} = \{A: A \rightarrow BC, B \in V_{11}, C \in V_{22}\}$
    - ❖ No production rule for $AA \Rightarrow \emptyset$ ($V_{12} = \emptyset$)

A membership algorithm for CFG

- **CYK algorithm example**
  - Determine whether the string $w = aabbb$ is in the language generated by the grammar
    - ❖ $S \to AB,\ A \to BB \mid a,\ B \to AB \mid b$
  - $V_{11} = \{A\}, V_{22} = \{A\}, V_{33} = \{B\}, V_{44} = \{B\}, V_{55} = \{B\}$
  - $V_{12} = \{A: A \to BC, B \in V_{11}, C \in V_{22}\}$
    - ❖ No production rule for $AA \Rightarrow \emptyset$ ($V_{12} = \emptyset$)
  - $V_{23} = \{A: A \to BC, B \in V_{22}, C \in V_{33}\}$
    - ❖ $S \to AB$ and $B \to AB$ exist
    - ❖ $V_{23} = \{S, B\}$

A membership algorithm for CFG

- **CYK algorithm example**
  - Determine whether the string $w = aabbb$ is in the language generated by the grammar
    - ❖ $S \rightarrow AB,\ A \rightarrow BB \mid a,\ B \rightarrow AB \mid b$
  - $V_{11} = \{A\}, V_{22} = \{A\}, V_{33} = \{B\}, V_{44} = \{B\}, V_{55} = \{B\}$
  - $V_{12} = \emptyset,\ V_{23} = \{S, B\},\ V_{34} = \{A\},\ V_{45} = \{A\}$

A membership algorithm for CFG

- **CYK algorithm example**
  - Determine whether the string $w = aabbb$ is in the language generated by the grammar
    - ❖ $S \rightarrow AB,\ A \rightarrow BB \mid a,\ B \rightarrow AB \mid b$
  - $V_{11} = \{A\}, V_{22} = \{A\}, V_{33} = \{B\}, V_{44} = \{B\}, V_{55} = \{B\}$
  - $V_{12} = \emptyset,\ V_{23} = \{S, B\},\ V_{34} = \{A\},\ V_{45} = \{A\}$
  - $V_{13} = \{S, B\},\ V_{24} = \{A\},\ V_{35} = \{S, B\}$

A membership algorithm for CFG

- **CYK algorithm example**
  - Determine whether the string $w = aabbb$ is in the language generated by the grammar
    - ❖ $S \rightarrow AB,\ A \rightarrow BB \mid a,\ B \rightarrow AB \mid b$
  - $V_{11} = \{A\}, V_{22} = \{A\}, V_{33} = \{B\}, V_{44} = \{B\}, V_{55} = \{B\}$
  - $V_{12} = \emptyset,\ V_{23} = \{S, B\},\ V_{34} = \{A\},\ V_{45} = \{A\}$
  - $V_{13} = \{S, B\},\ V_{24} = \{A\},\ V_{35} = \{S, B\}$
  - $V_{14} = \{A\},\ V_{25} = \{S, B\}$

A membership algorithm for CFG

- **CYK algorithm example**
  - Determine whether the string $w = aabbb$ is in the language generated by the grammar
    - $S \rightarrow AB,\ A \rightarrow BB \mid a,\ B \rightarrow AB \mid b$
  - $V_{11} = \{A\}, V_{22} = \{A\}, V_{33} = \{B\}, V_{44} = \{B\}, V_{55} = \{B\}$
  - $V_{12} = \emptyset,\ V_{23} = \{S, B\},\ V_{34} = \{A\},\ V_{45} = \{A\}$
  - $V_{13} = \{S, B\},\ V_{24} = \{A\},\ V_{35} = \{S, B\}$
  - $V_{14} = \{A\},\ V_{25} = \{S, B\}$
  - $V_{15} = \{S, B\}$
  - Since $S \in V_{15}$, hence, $w \in L(G)$

A membership algorithm for CFG

- **CYK algorithm example**
  - Determine whether the string $w = baaba$ is in the language generated by the grammar
    - ❖ $S \rightarrow AB \mid BC$
    - ❖ $A \rightarrow BA \mid a$
    - ❖ $B \rightarrow CC \mid b$
    - ❖ $C \rightarrow AB \mid a$

A membership algorithm for CFG

- **CYK algorithm example**
  - Determine whether the string $w = baaba$ is in the language generated by the grammar
    - ❖ $S \rightarrow AB \mid BC$
    - ❖ $A \rightarrow BA \mid a$
    - ❖ $B \rightarrow CC \mid b$
    - ❖ $C \rightarrow AB \mid a$

5	{S,A, C}				
4	∅	{S,A, C}			
3	∅	{B}	{B}		
2	{S,A}	{B}	{S, C}	{S,A}	
1	{B}	{A, C}	{A, C}	{B}	{A, C}
	b	a	a	b	a

A membership algorithm for CFG

- **CYK algorithm practice**
  - Determine whether the string $w = abaab$ is in the language generated by the grammar
    - ❖ $S \rightarrow AB \mid BC$
    - ❖ $A \rightarrow BA \mid a$
    - ❖ $B \rightarrow CC \mid b$
    - ❖ $C \rightarrow AB \mid a$

Next Lecture

- **Pushdown Automata**