Chap. 6 Simplification of Context-free Grammars and Normal Forms

→ parsing 하기 좋은 형태 (정규 형태)

$w \in L(G)$?

$S \Rightarrow \cdots \Rightarrow w$?

parsing

parsing

Agenda of Chapter 6

- Methods for Transforming Grammars
  - A useful substitution rule
  - Removing useless productions
  - Removing λ productions
  - Removing unit-productions

- Two Important Normal Forms
  - Chomsky normal form
  - Greibach normal form

- A Membership Algorithm for Context-Free Grammars

λ – free languages

- L : any context-free language
- G=(V,T,S,P) : a CFG for L-{λ}
- Grammar for L is given by
  - Adding new variable S_0 and a production rule $\boxed{S_0 \rightarrow S \mid \lambda}$

    "start" $S \overset{*}{\Rightarrow} w, \neq \lambda$
- From this, we can say that…
  - Any nontrivial conclusion for L-{λ} will almost certainly transfer to L.
  - No practical difference between CFL including λ and those without λ.
- Restriction on discussions in this chapter
  - Consider only λ-free languages.

A Useful Substitution Rule(1/2)

Theorem 6.1] Consider two grammar G, G'

- Let G=(V,T,S,P) be a CFG

  P contains a production $\boxed{A \rightarrow x_1Bx_2}$,

  delete. → 대체.

  - A, B : different variables
  - $B \rightarrow y_1 \mid y_2 \mid \dots \mid y_n$: all productions with B in the left side.

- Let G'=(V,T,S,P') be a grammar

  P' is constructed by

  - deleting $A \rightarrow x_1Bx_2$,

    사전에 어떻게 바꿀지 집어넣어준다.

  - adding $A \rightarrow x_1y_1x_2 \mid x_1y_2x_2 \mid \dots \mid x_1y_nx_2$.

- Then, $\boxed{L(G')} = \boxed{L(G)}$

Proof) Show that $\forall w \in L(G)$, $w \in L(G')$, and vice versa.

- Suppose $S \overset{*}{\Rightarrow}_G w$.
- For the derivation, $S \overset{*}{\Rightarrow}_G u_1Au_2 \Rightarrow_G u_1\boxed{x_1Bx_2}u_2 \Rightarrow_G u_1\boxed{x_1y_jx_2}u_2$,
- With G', we have $S \overset{*}{\Rightarrow}_{G'} u_1Au_2 \Rightarrow_{G'} u_1x_1y_jx_2u_2$
- Similarly, we can show vice versa.

A Useful Substitution Rule(2/2)

$A \to a \mid aaA \mid abaaaAc \mid abbc$
$B \to aaaA \mid b$

Ex6.1]

- $G=(\{A,B\},\{a,b,c\}, A,P)$

  $A \to a \mid aaA \mid abBc$ → delete

  $B \to aaa\,A \mid b$

  → 찾을 수 있는건 여기밖에 없다.

- A new grammar G' which is equivalent G

  delete $A \to abBc$

  add $A \to ab\,aaaAc \mid abbc$

  P: $A \to a \mid aaA \mid abaaaAc \mid abbc$

  $B \to aaaA \mid b$ → 무조건 써줘야 한다.

- Derivation of aaabbc

  G: $A \to aaA \Rightarrow aaabBc \Rightarrow aaabbc$

  G': $A \to aaA \to aaabbc$ → parsing에 더 유리

- Note) G' still has the variable B.

Removing Useless Productions(1/3)

[Definition] Usefulness & uselessness of variable and productions

- G=(V,T,S,P) : a CFG

$S \overset{*}{\Rightarrow} A \overset{*}{\Rightarrow} w$

↳ 중간에 무조건 나와야 한다.

- A **useful** variable A∈V
  - there is at least one w∈L(G) such that
  - $S \overset{*}{\Rightarrow} xAy \overset{*}{\Rightarrow} w$, with x, y ∈ $(V \cup T)^*$.

    derivation 할 때 쓰인다.
  - A variable is useful iff it occurs in at least one derivation.
- A **useless** variable : the one that is not useful. derivation 에서 사용 안됨.
- A **production is useless** if it involves any useless variable.

→ derivation 시 안 나타남.

useless

$A \overset{*}{\Rightarrow} w$ 불가능.

- Useless variable and/or production
  - ① S → aSb | λ |A, A → aA
  - ② S → A, A → aA| λ, B → bA

$S \Rightarrow A \Rightarrow aA \Rightarrow aaA \Rightarrow aaaA \Rightarrow \cdots \Rightarrow \cancel{w}$

→ terminal로 안바뀜

useless 하면 위 과정이 derivation 과정에서 나타나지 않는데, 위 2가지 생각해 봐

useless

S에서 B로 갈 수 없음.

$\cancel{S \overset{*}{\Rightarrow}} B \Rightarrow bA \Rightarrow b$

Removing Useless Productions(2/3)

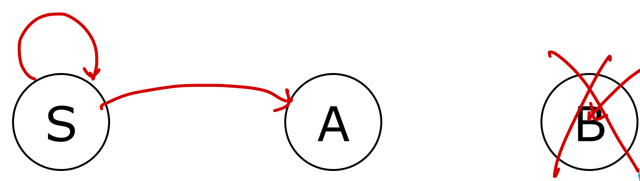
Ex6.3] G=({S,A,B,C},{a,b},S,P)

- $S \rightarrow aS \mid A \mid C,\ A \rightarrow a,\ B \rightarrow aa,\ C \rightarrow aCb$
- Find variables that cannot derive terminal string : C.
- Get new grammar G_1= ({S,A,B},{a,b},S,P_1) $\boxed{P_1: S \rightarrow aS \mid A \quad A \rightarrow a. \quad B \rightarrow aa.}$

useless production : S → C , C → aCb delete

- Find variables that cannot be reached from S
  - Use Dependency Graph for variables

S→aS



S에서 path가 없는 node.
→ useless

→ 안쓰이는 terminal도 빼야지.

- Get final grammar G'= ({S,A},{a,b},S,P')

P' = S→aS|A , A→a.

S ⇒

Removing Useless Productions(3/3)

Theorem 6.2] Let G =(V,T,S,P) be a CFG
Then there exist an equivalent grammar G'
without useless variables or productions.

Proof) Give an algorithm making G' from G

1. Find $G_1=(V_1,T_2,S,P_1)$ having only variables A for which $A \overset{*}{\Rightarrow} w \in T^*$.

   terminal을 만드는 variable은 V_1로.

   a. Set $V_1=\{\}$ empty 에서 시작 ⇒ ① {A,B} ⇒ ② {A,B,S}

   하나씩 채워 나간 상황

   b. Repeat until no more variables are added to V_1 ;
      For every A with a production $A \rightarrow x_1x_2 \ldots x_n$, ($x_i \in V_1 \cup T$),
      add A to V_1

      {A,B,S} → G 업데이트

   c. P_1: set of all productions with only symbols in ($V_1 \cup T$).

   {S,A,B}만 처리. 결과 반영.

2. Get G' from G_1 by deleting followings :
   - all variables that cannot be reached from S → dependency graph를 그린다.
   - all productions involving the variables.
   - all terminals that does not occur in some useful productions.

Removing λ - Productions(1/3)

- λ-production
  - Any production of a CFG of the form $A \rightarrow \lambda$.

parsing 하는데에 매우 안좋음.

$A \rightarrow x$, $S \rightarrow A$). $S \Rightarrow A \Rightarrow x$

nullable variable

- Nullable variable

λ로 바뀔 수 있는 variable 이다

  - Any variable A for which the derivation $A \overset{*}{\Rightarrow} \lambda$ is possible.
- When a grammar with λ-production generates a language without λ, the λ-production can be removed.

null

Ex6.4] $S \rightarrow aS_1b \quad S_1 \rightarrow aS_1b \mid \lambda$

$S \Rightarrow aS_1b$

- Generating language : $L(G) = \{a^n b^n \mid n > 0\}$: λ-free Lang

$\lambda \in L(G)$

↳ λ를 없애는 production을 찾을 수 있다

- Removing λ-production :

$S \rightarrow aS_1b \mid ab$

$S_1 \rightarrow aS_1b \mid ab$.

$S \rightarrow aSb \mid ab$.

→ nullable 은 미리 null로 바꿔 주자

Removing λ - Productions(2/3)

Theorem 6.3]

Let G ={V,T,S,P} : a CFG with $\lambda \notin L(G)$.

Then there exist an equivalent grammar G′ without λ-productions

Proof)

1. Find the set V_N of all nullable variables of G. $V_N = \{\ \}$.
   a. For all $A \rightarrow \lambda$, put A into V_N. $=\{A\}$.
   b. Repeat until no more variables are added to V_N :
      For all $B \rightarrow A_1A_2 \ldots A_n$, with all $A_i \in V_N$, put B into V_N.
2. For each production $A \rightarrow x_1x_2 \ldots x_m$, ($m \geq 1$, x_i in $V \cup T$), put followings into P′.
   모든 production에 대해서
   - The production itself.
   - All productions generated by replacing variables in V_N with λ in all possible combinations.
     → production 안에 V_N에 속하는 variable이 있으면 λ로 바꾸자.
   - One exception : if all x_i are nullable, $A \rightarrow \lambda$ is not put into P′.
     모든 variable이 null이 되었을 때의 예외 상황

Removing λ - Productions(3/3)

Ex6.5]

nullable x.

nullable

- $S \to ABaC,\ A \to BC,\ B \to b \mid \lambda,\ C \to D \mid \lambda,\ D \to d.$
- Find nullable variables $V_N = \{B, C, A\}$ nullable → λ로 바꿔주기.
- Find P'

① $S \to ABaC \mid a \mid BaC \mid AaC \mid ABa \mid aC \mid Aa \mid Ba$

동시에 λ로 다 바꾼 게 아니므로 모든 combination 들을 집어넣어야 한다.

② $A \to BC \mid B \mid C \mid \lambda$ → delete.

③ $B \to b$

④ $C \to D$

⑤ $D \to d$

Removing Unit-Productions(1/2)

variable이 포함된 바꾼다.

- Unit-production
  - Any production of a CFG of the form $A \rightarrow B$, $(A, B \in V)$.

λ는 다 없애 놓고 생각해야 한다.

Theorem 6.4]
Let G ={V,T,S,P} : a CFG without λ-productions. Then there exist an equivalent grammar G′ without unit-productions

Proof)

$A \rightarrow B$
$B \rightarrow C$

1. For each variable A, find all variables B such that $A \overset{*}{\Rightarrow} B$ $A \Rightarrow B \Rightarrow C$
   - use dependency graph with only edges for unit-productions.
2. Put all non-unit productions of P into P′.
3. For all A, B found in step 1, $A \Rightarrow B$ → delete 하는 대신 추가

substitution
add $A \rightarrow y_1 \mid y_2 \mid \ldots \mid y_n$ into P′
where $B \rightarrow y_1 \mid y_2 \mid \ldots \mid y_n$ is all rules in P′ with B on the left.

①에서 찾은 모든 관계에 대해

Removing Unit-Productions(2/2)

- Example 6.6) Remove all unit-production from
  - S → Aa | B, B→ A|bb, A→ a|bc|B

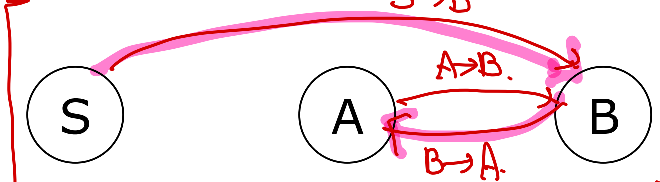
    A→λ 있으면 λ부터 먼저 없애야 함

    S→B
  - Dependency graph

    ○ $\overset{*}{\Rightarrow}$ ○

    A → B

    unit production 만 edge 표시



S⇒B ... ①
B⇒A ... ②
A⇒B ... ③
$S \overset{*}{\Rightarrow} A$ ... ④

  - Put all non-unit production into P'.

    S→Aa   B→bb   A→a|bc.
  - Add new rules

    ① S⇒B 에서 S→bb

    ② B⇒A 에서 B→a|bc.

    ③ A⇒B 에서 A→bb

    ④ $S \overset{*}{\Rightarrow} A$ 에서 S→a|bc.
  - Finally obtained rules:

    S→Aa|bb|a|bc. , B→bb|a|b|c , A→a|bc|bb.
  - Any useless variable and/or production?

    $S \overset{*}{\Rightarrow} B \overset{*}{\Rightarrow} w$

Transforming a Context-Free Grammar

Theorem 6.5]

Let L be a CFL without λ.

Then there exists a CFG generating L without useless productions, λ-productions, or unit-productions.

Proof)

From theorems 6.2, 6.3, and 6.4,

we can remove all undesirable productions using the following sequence of steps:

1. Remove λ-productions
2. Remove unit-productions
3. Remove useless productions.

★순서를 지켜라★

Chomsky Normal Form(1/3)

$A \to x$ $\xrightarrow{x \in (V \cup T)^*}$ Context-free.

Chomsky Normal Form (CNF)

- A normal forms with limits on the number of symbols on the right of a production.
- A CFG with which all productions are of the form
- $A \to BC$, or $A \to a$ $(A, B, C \in V,\ a \in T)$

Variable. 2개 ok.

terminal symbol 1개.

Ex6.7]

- $S \to AS \mid a$, $A \to SA \mid b$ ⇒ CNF
- $S \to AS \mid AAS$, $A \to SA \mid aa$

X   X ⇒ CNF X

Chomsky Normal Form(2/3)

Theorem 6.6]

For any CFG $G=(V,T,S,P)$ without λ in $L(G)$,

There exist an equivalent grammar $G'=(V',T',S,P')$ in CNF

모든 context free grammar 가 CNF로 표현 가능하다.

Proof) Construction of G' from G.

저번 시간에 배웠던 내용

Assume G has no λ-production and unit-production.

1. Find $G_1=(V_1,T,S,P_1)$ with which all productions have the form

P_1

$A \rightarrow a$, or $A \rightarrow C_1C_2 \ldots C_n$ (C_i in V_1, a in T) → 일단 이런 형태로. (중간 형태)

- For all productions $A \rightarrow x_1x_2 \ldots x_n$ ($x_i \in V \cup T$),

└ 임의의 production

- if n=1, put the production into P_1. $A \rightarrow a$ → 그대로 P_1에 추가.
- else

$\cancel{A \rightarrow B}$ → unit production 이기 때문에 없는 거라 가정

1) introduce new variables B^a for each a in T.

2) put $A \rightarrow C_1C_2 \ldots C_n$ ($C_i=x_i$ for x_i in V, $C_i=B^a$ for $x_i=a$) into P_1.

3) Put $B^a \rightarrow a$ into P_1.

$A \rightarrow abC$

$V^b \rightarrow b$

$A \rightarrow V^aV^bC$

$V^a \rightarrow a$

Chomsky Normal Form(2/3)

Proof continued)

2. Add variables to reduce the length of right side.
   - For all productions $A \to a$, and $A \to C_1C_2$: put them into P'.
   - For productions $A \to C_1C_2 \ldots C_n$, n>2 :
     - Introduce $D_1, D_2 \ldots$ and add productions $A \to C_1D_1$, $D_1 \to C_2D_2$, … , $D_2 \to C_{n-1}C_n$.

2개가 남을 때 까지.

$A \to C_1 \underbrace{C_2C_3C_4}_{D_1}$

$A \to C_1D_1$.
$D_1 \to C_2 \underbrace{C_3C_4}_{D_2}$.

$A \to C_1D_1$
$D_1 \to C_2D_2$
$D_2 \to C_3C_4$

Chomsky Normal Form(3/3)

Ex6.8] Convert the grammar with productions

$S \rightarrow ABa, \quad A \rightarrow aab, \quad B \rightarrow Ac$

λ production, unit production이 있는지.

to Chomsky Normal Form.

- Step 1)

  ① $S \rightarrow ABV_a$

  ② $A \rightarrow V_a V_a V_b$

  ③ $\boxed{B \rightarrow AV_c}$

  D_1, D_2

  $V_a \rightarrow a$,
  $V_b \rightarrow b$,
  $V_c \rightarrow c$,

- Step 2)

  $B \rightarrow AV_c$, $V_a \rightarrow a$, $V_b \rightarrow b$, $V_c \rightarrow c$

  ①′ $S \rightarrow AD_1$, $D_1 \rightarrow BV_a$

  ②′ $A \rightarrow V_a D_2$, $D_2 \rightarrow V_a V_b$

  P

  $\Rightarrow$ CNF.

Greibach Normal Form

Greibach Normal Form (GNF)

- Normal Forms with limits on the positions (위치) in which terminals and variables can appear.
- A CFG with which all productions are of the form
- A → ax (x in V^*, a in T)

  terminal 이 맨 앞에 1개만. variable이 여러개 가 (여러 개 상관) terminal 뒤에

- c.f.) s-grammar (pair가 1개만) → 모든 context free가 될 수 x.

substitution. O X

Ex6.9] S → AB , A → aA|bB|b, B → b

- Applying substitution rule

delete 하는 대신에 add.

$S \to aAB \mid bBB \mid bB$.

(A,b) 쌍 2개 → s-grammar x.

Ex6.10] Convert S → abSb | aa into Greibach normal form

X. V^b. $V^b \to b$. $S \to aV^bSV^b$.

> **Theorem 6.7]**
> For any CFG G=(V,T,S,P) without λ in L(G),
> There exist an equivalent grammar G' in Greibach Normal Form.

CYK algorithm for Chomsky Normal Form

CYK algorithm

$w \in L(G)$?

- Membership algorithm of $w=a_1a_2 \ldots a_n$ for $G=(V,T,S,P)$ in CNF.
- Define substring $w_{ij}=a_i \ldots a_j$ and subset $V_{ij}=\{A \in V \mid A \overset{*}{\Rightarrow} w_{ij}\}$

  Variable의 집합

  w_{ij}를 생성해 낼 수 있는 V들의 집합.
- $w \in L(G)$ if and only if $S \in V_{1n}$.

  "Membership을 알고싶기 때문."

  $w = a_1 a_2 \ldots a_i \cdots a_j \cdots a_{n-1} a_n$ (w_{11}, w_{ij})

  $w_{1n} \in L(G) \Leftrightarrow S \overset{*}{\Rightarrow} w_{1n}$.
- How to find V_{1n}
  1. Find V_{ii} using the fact that $A \in V_{ii}$ iff G contains $A \rightarrow a_i$.

     $V_{11} = \{A \mid A \Rightarrow w_{11} = a_1\}$.

     $V_{1n} = \{A \mid A \overset{*}{\Rightarrow} w_{1n} = w\}$. 을 찾고 싶다.
  2. Find V_{ij} using the formula

     Formal 하게 쓴 식

     $V_{ij} = \bigcup_{k \in \{i,i+1,\ldots,j-1\}} \{A \mid A \rightarrow BC, \text{ with } B \in V_{ik}, C \in V_{k+1,j}\}$

     이 안에 S가 속하는지가 중요하다.
  3. Compute all V_{ij} by proceeding in the sequence
     - Compute V_{11}, V_{22}, …, V_{nn}
     - Compute V_{12}, V_{23}, … $V_{n-1,n}$
     - Compute V_{13}, V_{24}, … $V_{n-2,n}$, and so on.

       ⋮

       V_{1n}.

$V_{ii} = \{A \mid A \rightarrow a_i\}$.

$A \rightarrow a$

$A \rightarrow BC$.

$V_{12} = \{A \mid A \overset{*}{\Rightarrow} w_{12} = a_1 a_2\}$ (B C.)

$BC \overset{*}{\Rightarrow} a_1 a_2$

$a_1 C \Rightarrow a_1 a_2$

$= \{A \mid A \rightarrow BC\}$.

$B = V_{11}$ $C = V_{22}$

$V_{13} = \{A \mid A \overset{*}{\Rightarrow} w_{13} = a_1 a_2 a_3\}$. ($B$ C, B C)

$A \Rightarrow BC$ (V_{11}, V_{23})

$V_{13} = \{A \mid A \rightarrow BC, B \in V_{12}, C \in V_{33}\}$.

$V_{13} = \{A \mid A \rightarrow BC, B \in V_{11}, C \in V_{23}\}$.

$V_{12} = \{A \mid A \rightarrow BC, B \in V_{11}, C \in V_{22}\}$

$V_{i,i+1} = \{A \mid A \rightarrow BC, B \in V_{ii}, C \in V_{i+1,i+1}\}$.

CYK algorithm for Chomsky Normal Form

Ex6.11] Apply CYK algorithm to w=aabbb and → CNF의 형태인지

$S \rightarrow AB,\ A \rightarrow BB|a,\ B \rightarrow AB|b.$

$A \rightarrow a$

V_{ii}	$V_{11}=\{A\}$	$V_{22}=\{A\}$	$V_{33}=\{B\}$	$V_{44}=\{B\}$	$V_{55}=\{B\}$
V_{ii+1}	$V_{12}=V_{11}V_{22}=\emptyset$	$V_{23}=V_{22}V_{33}=\{S,B\}$	$V_{34}=V_{33}V_{44}=\{A\}$	$V_{45}=\{A\}$	
V_{ii+2}	$V_{13}=V_{11}V_{23} \cup V_{12}V_{33}=\{S,B\}$	$V_{24}=V_{22}V_{34} \cup V_{23}V_{44}=\{A\}$	$V_{35}=\{S,B\}$		
V_{ii+3}	$V_{14}=V_{11}V_{24} \cup V_{12}V_{34} \cup V_{13}V_{44}=\{A\}$	$V_{25}=V_{22}V_{35} \cup V_{23}V_{45} \cup V_{24}V_{55}=\{S,B\}$			
V_{ii+4}	$V_{15}=V_{11}V_{25} \cup V_{12}V_{35} \cup V_{13}V_{45} \cup V_{14}V_{55}=\{S,B\} \in L(G)$				

$\emptyset \rightarrow (AA, AB) = \{S\}$ (A B)

$\emptyset \rightarrow (AB, AS)=\{\}$

$\{S,B\}\{B\}$. $\emptyset \rightarrow (SB, BB) = \{A\}$

$\{A\}\ \{S,B\}$

aabbb=w

S 첫 순서에 들어가면 참

CYK algorithm for Chomsky Normal Form

Ex6.11] Apply CYK algorithm to w=aabbb and

$S \rightarrow AB,\ A \rightarrow BB|a,\ B \rightarrow AB|b.$

V_{ii}	$V_{11}=$	$V_{22}=$	$V_{33}=$	$V_{44}=$	$V_{55}=$
V_{ii+1}	$V_{12}=$		$V_{34}=$		
		$V_{23}=$		$V_{45}=$	
V_{ii+2}	$V_{13}=$				
		$V_{24}=$			
			$V_{35}=$		
V_{ii+3}	$V_{14}=$				
		$V_{25}=$			
V_{ii+4}	$V_{15}=$				