Chap. 7
Pushdown Automata

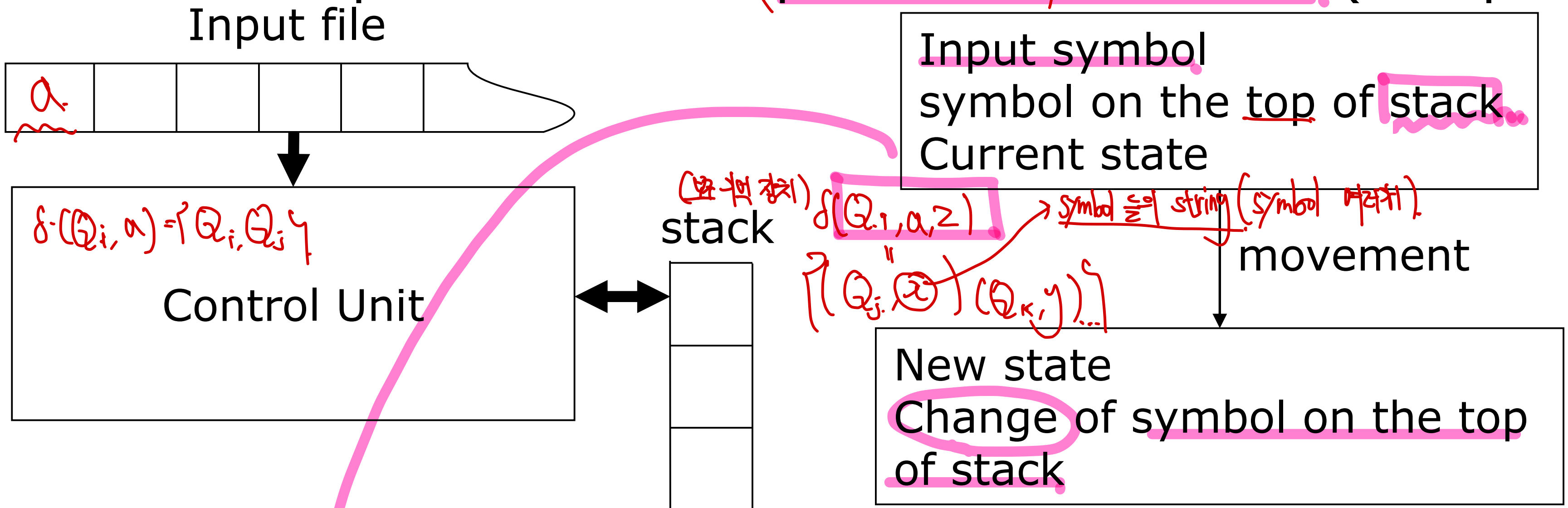
What we look at in this section

- Nondeterministic pushdown automata (NPDA)
- Pushdown automata and context-free languages
- Deterministic pushdown automata (DPDA) and deterministic context-free languages (DCFL)
- Grammars for DCFL

Definition of a Pushdown Automata(1/2)

NPDA

- Schematic representation of pushdown automata (accepter)



Definition] Nondeterministic pushdown accepter (npda)

– $M=(Q, \Sigma, \Gamma, \delta, q_0, z, F)$ 새로 추가 된 것

– Γ: stack alphabet (a finite set of symbols)

– $\delta : Q\times(\Sigma \cup \{\lambda\})\times\Gamma \rightarrow$ finite subset of $Q\times\Gamma^*$ $(2^{Q\times\Gamma^*})$

δ의 domain

stack symbol 의 string

– $Z \in \Gamma$: stack start symbol

시작 시점에 stack의 top에 있는 symbol

현재 state

input symbol

stack 의 start symbol

Definition of a Pushdown Automata(2/2)

Ex7.2] DPDA로 판단해 줄 수 없다. → DCFL.

상태가 Q_0를 final state로

NPDA 지만 DPDA x

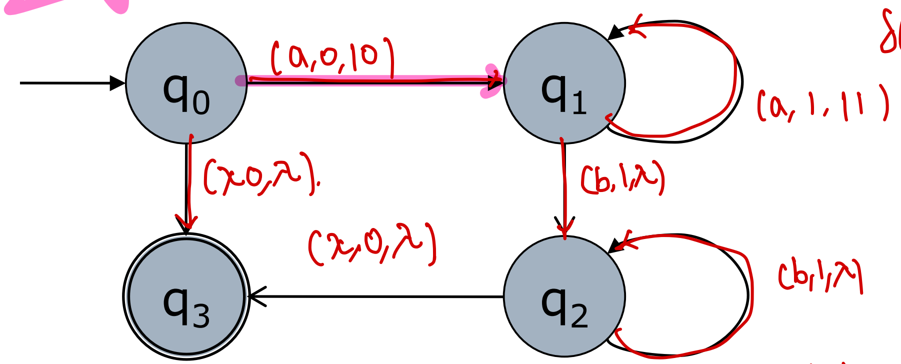
$Q=\{q_0, q_1, q_2, q_3\}$, $\Sigma=\{a,b\}$, $\Gamma=\{0,1\}$, $z=0$, $F=\{q_3\}$ ← 때문에

- $\delta(q_0, a, 0)=\{(q_1,10), (q_3,\lambda)\}$, $\delta(q_0, \lambda, 0)=\{(q_3,\lambda)\}$,
- $\delta(q_1, a, 1)=\{(q_1,11)\}$, $\delta(q_1, b, 1)=\{(q_2,\lambda)\}$,
- $\delta(q_2, b, 1)=\{(q_2,\lambda)\}$, $\delta(q_2, \lambda, 0)=\{(q_3,\lambda)\}$.
- $L=\{a^n b^n \mid n \geq 0\} \cup \{a\}$

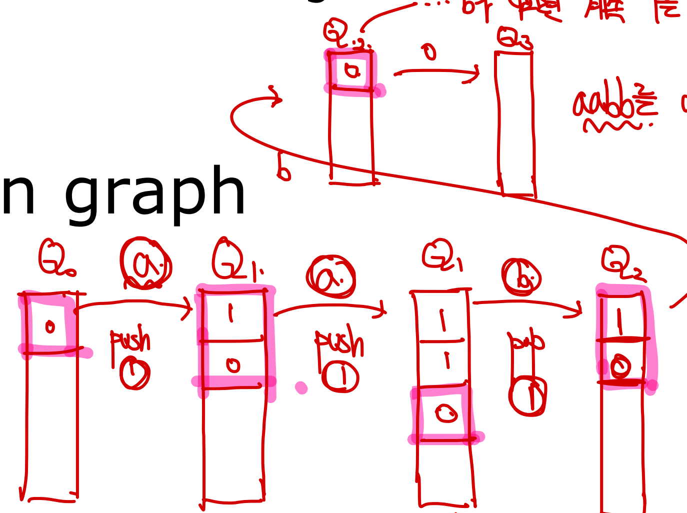
… b가 나오면 계속 1을 pop

aabb를 accept

Ex7.3] Representation by transition graph



δ(Q0,a,0)



… a가 나오면 계속 1을 push.

stack의 변화도 나타내 주기.

Language Accepted by a PDA

- Some notations for NPDA configurations and transition
  - stack의 top에 있는것 오토마타에 주어진 상황
  - 현재 input
  - (q_i, aw, bx): current configuration with state q_i, input string aw, and stack string bx
  - $(q_i, aw, bx) \vdash (q_j, w, yx)$: a transition with $\delta(q_i, a, b) \ni (q_j, y)$
  - $(q_i, w_1w_2, x) \vdash^* (q_j, w_2, y)$: transitions with $\delta(q_i, w_1, x) \ni (q_j, y)$

Definition]Language accepted by M$=(Q, \Sigma, \Gamma, \delta, q_0, z, F)$

- $L(M)=\{w \in \Sigma^* \mid (q_0, w, z) \vdash^*_M (p, \lambda, u), p \in F, u \in \Gamma^*\}$

finite state 중 하나.

stack에 남아 있는건 어떠거나 상관없다.

w를 다 처리.

stack 안의 symbol로 이루어진 것.

Ex7.4] npda for $L=\{w \in \{a,b\}^* \mid n_a(w)=n_b(w)\}$

$M=(\{q_0, q_f\}, \{a,b\}, \{0,1,z\}, \delta, q_0, z, \{q_f\})$

a도 x b도 x.

a가 나오면 0을 push 하 / b가 나오면 0을 pop 하 1을 push / 1은 pop.

- $\delta(q_0, \lambda, z)=\{(q_f, z)\}$ $\quad$ $\delta(q_0, a, z)=\{(q_0, 0z)\}$ $\quad$ $\delta(q_0, b, z)=\{(q_0, 1z)\}$
- $\delta(q_0, a, 0)=\{(q_0, 00)\}$ $\quad$ $\delta(q_0, b, 0)=\{(q_0, \lambda)\}$
- $\delta(q_0, a, 1)=\{(q_0, \lambda)\}$ $\quad$ $\delta(q_0, b, 1)=\{(q_0, 11)\}$

매칭시켜서 없앤다.

- w=baab

실행

$(q_0, baab, z) \vdash (q_0, aab, 1z) \vdash (q_0, ab, z) \vdash (q_0, b, 0z) \vdash (q_0, \lambda, z) \vdash (q_f, \lambda, z)$

accept 된다.

a와 b에서부터 도달한다.

Language Accepted by a PDA

npda

Ex7.5] npda for $L=\{ww^R \mid w \in \{a,b\}^+\}$

$M=(\{Q_0,Q_1,Q_f\},\{a,b\},\{a,b,z\},\delta,Q_0,z,\{Q_f\})$.

w가 나오는 동안 input을 stack에 push.

→ " stack에서 matching 시켜서 pop.

$[q_0]$ Push w on the stack

$\delta(Q_0,a,z)=\{(Q_0,az)\}$ $\delta(Q_0,a,a)=\{(Q_0,aa)\}$ $\delta(Q_0,a,b)=\{(Q_0,ab)\}$

$\delta(Q_0,b,z)=\{(Q_0,bz)\}$ $\delta(Q_0,b,a)=\{(Q_0,ba)\}$ $\delta(Q_0,b,b)=\{(Q_0,bb)\}$

$[q_0 \rightarrow q_1]$ Guess middle of string:

$\delta(Q_0,\lambda,a)=\{(Q_1,a)\}$. $\delta(Q_1,\lambda,b)=\{(Q_1,b)\}$.

$[q_1]$ Match w^R:

$\delta(Q_1,a,a)=\{(Q_1,\lambda)\}$, $\delta(Q_1,b,b)=\{(Q_1,\lambda)\}$.

$[q_1 \rightarrow q_f]$ Go to final

$\delta(Q_1,\lambda,z)=\{(Q_f,z)\}$.

– w=abba

$(Q_0,abba,z) \vdash^* (Q_f,\lambda,z)$ → accept.

$(Q_0,abbb,z) \vdash^*$? reject.

Pushdown Automata for CFL (1/4)

- For every CFL, there is an npda accepting it.
  - For a language generated by a grammar in GNF, we can construct a npda.

① ② ③

Greibach normal form.
$S \to aV$ ($a \in T$, $V \in V^*$)

Ex7.6] Find a pda for languages generated by $S \to aSbb \mid a$

- Transform to Greibach normal form :

  $S \to aSA \mid a$, $A \to bB$, $B \to b$

$[q_0 \to q_1]$ Put S on the stack by

$[q_1]$ For $S \to aSA$, $S \to a$

$[q_1]$ For $A \to bB$, $B \to b$

$[q_1 \to q_f]$ Completion of derivation

M=

Pushdown Automata for CFL (2/4)

Theorem 7.1] CFG → NPDA

For any CFL L, there exist an npda M such that L=L(M)

Proof) Let G=(V,T,S,P) be a grammar in GNF generating L.

Construction of M such that L(M)=L(G)

$A \rightarrow aBC.$ CFG

→ derivation 도중

Let $M=(\{q_0,q_1,q_f\},T,V\cup\{z\},\delta,q_0,z,\{q_f\}),\ z\notin V$

$Q_1.$

First transition : $\delta(q_0, \lambda, z)=\{(q_1, Sz)\}$

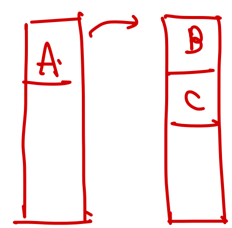
→ stack의 top에 "S"를 push.

For $A\rightarrow au$: $(q_1, u) \in \delta(q_1, a, A)$

production 개 ⟺ transition 개

Final state : $\delta(q_1, \lambda, z)=\{(q_f, z)\}$

→ stack에 variable이 모두 사라지면 final



Proof of L(G)⊆L(M) (i.e. M accepts any w∈L(G))

1. Show that if $S\overset{*}{\Rightarrow}w$ then $(q_1, w, Sz) \vdash^* (q_1, \lambda, z)$ by induction.

For $a_1a_2\ldots a_nA_1A_2\ldots A_n \Rightarrow a_1a_2\ldots a_nbB_1B_2\ldots B_kA_2\ldots A_n$,

M has $(q_1, B_1B_2\ldots B_k) \in \delta(q_1, b, A_1)$

Finally, we have $(q_0, w, z) \vdash (q_1, w, Sz) \vdash^* (q_1, \lambda, z) \vdash (q_f, \lambda, z)$

Pushdown Automata for CFL (3/4)

Proof Continued)

Proof of L(M)⊆L(G)

2. Let w be in L(M).

Then $(q_0, w, z) \vdash^* (q_f, \lambda, z)$ by definition.

Since there is only one way from q_0 to q_1 and q_1 to q_f,

we must have $(q_1, w, Sz) \vdash^* (q_1, \lambda, z)$

For any $w=a_1a_2\ldots a_n$,

First step : $(q_1, a_1a_2\ldots a_n, Sz) \vdash (q_1, a_2\ldots a_n, u_1z)$,

then the grammar has $S \to a_1u_1$, so that $S \Rightarrow a_1u_1$.

Repeating this for $u_1=Au_2$: $(q_1, a_2\ldots a_n, Au_2z) \vdash (q_1, a_3\ldots a_n, u_3u_2z)$,

then the grammar has $A \to a_2u_3$, so that $S \overset{*}{\Rightarrow} a_1a_2u_3u_2$.

From this observation, we can see that

$(q_1, a_1a_2\ldots a_n, Sz) \vdash^* (q_1, \lambda, z)$ implies $S \overset{*}{\Rightarrow} a_1a_2\ldots a_n$.

Remark] $\boxed{\text{If } \lambda \in L, \text{ add } \delta(q_0, \lambda, z)=\{(q_f, z)\}}$

Pushdown Automata for CFL (4/4)

Ex 7.7] Find npda M for the grammar

GNF

- $(S \rightarrow aA,\ A \rightarrow aABC \mid bB \mid a,\ B \rightarrow b,\ C \rightarrow c)$
- $M = (\{Q_0, Q_1, Q_f\}, \{a, b, c\}, \{A, B, C, z\}, \delta, Q_0, z, \{Q_f\})$
- First & Last step : $\delta(Q_0, \lambda, z) = \{(Q_1, Sz)\}$, $\delta(Q_1, \lambda, z) = \{(Q_f, z)\}$
- Other transitions

① $\delta(Q_1, a, S) = \{(Q_1, A)\}$

④, ② $\delta(Q_1, a, A) = \{(Q_1, ABC), (Q_1, \lambda)\}$

③ $\delta(Q_1, b, A) = \{(Q_1, B)\}$

⑤ $\delta(Q_1, b, B) = \{(Q_1, \lambda)\}$

⑥ $\delta(Q_1, c, C) = \{(Q_1, \lambda)\}$

npda → CFG x.

- w=aaabc

$S \Rightarrow aA \Rightarrow aaABC \Rightarrow aaaBC \Rightarrow aaabC \Rightarrow aaabc$

$(Q_0, aaabc, z) \vdash (Q_1, aaabc, Sz) \vdash (Q_1, aabc, Az) \vdash (Q_1, abc, ABCz) \vdash (Q_1, bc, BCz) \vdash (Q_1, c, Cz)$

$\vdash (Q_1, \lambda, z) \vdash (\boxed{Q_f}, \lambda, z)$

accept

Context-Free Grammars for PDA (1/3)

- Assumptions on npda
  - (a) Single final state q_f that is entered ⇔ the stack is empty
  - (b) All transitions must have the form

$$\delta(q_i, a, A) = \{c_k \mid c_k = (q_j, \lambda) \text{ or } (q_j, BC),\ k=1,\ldots,n\}$$

Note] For any npda, there exist an equivalent npda satisfying (a) & (b)

Theorem 7.2] NPDA → CFL

If L=L(M) for some npda M, then L is a CFL

Proof) **Construction of CFG $G=(V,T,S,P)$ satisfying $L(G)=L(M)$**

[Assumption] $M=(Q, \Sigma, \Gamma, \delta, q_0, z, \{q_f\})$ satisfies (a) &(b)

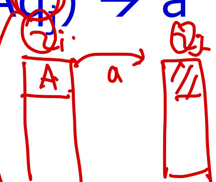
1. For each $\delta(q_i, a, A) = (q_j, \lambda)$, make production $(q_iAq_j) \rightarrow a$

"A가 stack의 top에서 변화되기 시작하는 state"

$A \rightarrow a$

"state의 변화도 써주자"

$\rightarrow (q_iAq_j) \rightarrow a$



state가 바뀌는데 how?

2. For each $\delta(q_i, a, A) = (q_j, BC)$, make productions

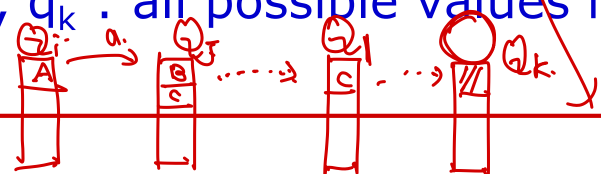
$(q_iAq_k) \rightarrow a(q_jBq_l)(q_lCq_k)$, ($q_l$, q_k : all possible values in Q)

B가 삭제됨.

C가 top에서 삭제됨.

$(q_i A q_k) \rightarrow a (q_j B q_l)(q_l C q_k)$

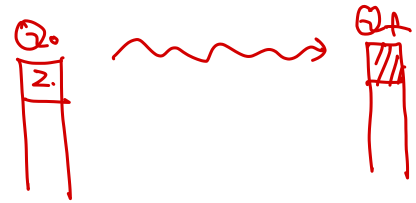
A가 삭제됨 = C가 삭제됨.



"A"가 stack에서 없어진 상태. state.

Context-Free Grammars for PDA (2/3)

(transition의 시작.



Qf는 stack이 empty 상태를 의미.

Proof continued)

3. Take start variable : $\boxed{(q_0 z q_f)}$

Thus, we have $G=(V, \Sigma, S, P)$ with $V=\{(q_i c q_j) \mid q_i, q_j \in Q,\ c \in \Gamma\}$, $S=(q_0 z q_f)$.

npda → CFG를 구할 수 있다.

Proof of L(G)=L(M)

- Show that for all $q_i, q_j \in Q$, $A \in \Gamma$, $X \in \Gamma^*$, $u, v \in \Sigma^*$,

  $(q_i, uv, AX) \overset{*}{\vdash} (q_j, v, X)$ iff $(q_i A q_j) \overset{*}{\Rightarrow} u$

  by using the construction rule and $\boxed{\text{induction}}$.
- From the conclusion, we see that

  $(q_0, w, z) \overset{*}{\vdash} (q_f, \lambda, \lambda)$ iff $(q_0 z q_f) \overset{*}{\Rightarrow} w$,
- This implies L(M)=L(G).

Context-Free Grammars for PDA (3/3)

Ex7.8] Find CFG for npda with transitions (initial: q_0, final: q_2)

(a), (b) ... ?

① $\delta(q_0, a, z) = \{(q_0, Az)\}$, ② $\delta(q_0, a, A) = \{(q_0, A)\}$

③ $\delta(q_0, b, A) = \{(q_1, \lambda)\}$, ④ $\delta(q_1, \lambda, z) = \{(q_2, \lambda)\}$

(a)

- Satisfy the assumptions :

②' $\delta(q_0, a, A) = (q_3, TA)$ ②'' $\delta(q_3, \lambda, T) = (q_0, \lambda)$

- For transitions of the form $\delta(q_i, a, A) = \{(q_j, \lambda)\}$

③ $(q_0 A q_1) \to b$ ④ $(q_1 z q_2) \to \lambda$ ③' $(q_3 T q_0) \to \lambda$

- For other transitions

① $(q_0 z q_k) \to a (q_0 A q_l)(q_l z q_k)$ $(l, k = 0,1,2,3)$: production 16개.

②' $(q_0 A q_k) \to a (q_3 T q_l)(q_l A q_k)$ (") "

- Start variable $(q_0 z q_2)$
- String aab

$(q_0 z q_2) \overset{①}{\Rightarrow} a (q_0 A q_1)(q_1 z q_2) \overset{②}{\Rightarrow} aa \underbrace{(q_3 T q_0)}_{\lambda} \underbrace{(q_0 A q_1)}_{b} \underbrace{(q_1 z q_2)}_{\lambda} \overset{②''③④}{\vdash} aab$

DPDA and DCFL(1/2)

Definition] Deterministic PDA

- A pushdown accepter that never has a choice in its move
- $M = (Q, \Sigma, \Gamma, \delta, q_0, z, F)$
- For every $q \in Q$, $a \in (\Sigma \cup \{\lambda\})$, $b \in \Gamma$,
  1. $\delta(q,a,b)$ contains at most one element. (δ의 결과가 1개 이하. 0개)
  2. For all $c \in \Sigma$, if $\delta(q,\lambda,b) \neq \emptyset$ then $\delta(q,c,b) = \emptyset$

허용 → 허용 안됨

Note] DPDA vs. DFA

- DPDA still has λ-transition. (Q,b)는 λ-transition 만 가능한 "λ전용" 임.
- DPDA also has dead configuration. → 필요 없는 것은 정의 안해도 됨.

- **Deterministic context-free language** DCFL 과 대응.
  - A language L with a dpda M such that L=L(M).

npda → CFL

$\{a^n b^{2n}\}$.

"$S \to S_1 | S_2$ $\{a^n b^n\} \cup \{a^n b^{2n}\}$.
$S_1 \to aS_1b | \lambda$
$S_2 \to aS_2bb | \lambda$.
CFL 임.

$S \to aSb | \lambda$ CFG $\{a^n b^n | n \geq 0\}$

∴ DPDA, NPDA 다 큼

CFL ⟺ NPDA
DCFL ⟺ DPDA
Reg ⟺ NFA dFA

$(ab)^*$

DPDA and DCFL(2/2)

Ex7.10] $L=\{a^nb^n \mid n\geq 0\}$ → DCFL

A $\boxed{\text{dpda}}$ $M= (\{q_0,q_1,q_2\}, \{a,b\}, \{0,1\}, \delta, q_0, 0, \{q_0\})$

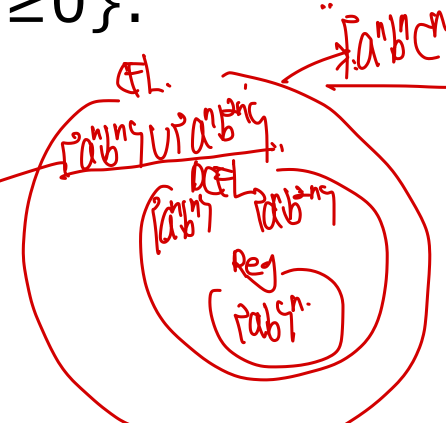
(Ex 7.2의 변형) $\delta(q_0,a,0)=(q_3,\lambda)$ 삭제 $\delta(q_0,\lambda,0)=(q_3,\lambda)$ 삭제 하는 대신 q_0를 final로.

Ex7.11] $L_1=\{a^nb^n \mid n\geq 0\}$, $L_2=\{a^nb^{2n} \mid n\geq 0\}$.

- L_1, L_2 are CFL.
- $L= L_1 \cup L_2$ is also a CFL.
- L is not deterministic CFL. how?

Deterministic 하지 않다.

CFL ≠ DCFL



Proof) Assume that L is a deterministic CFL.

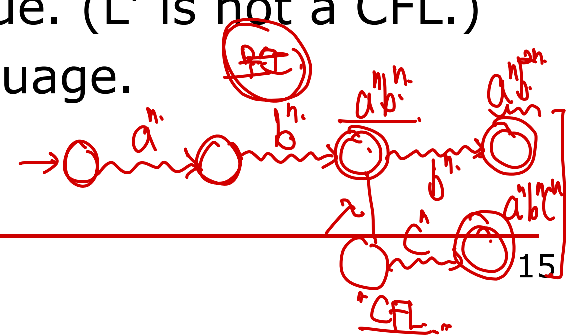
Then, by using the dpda, we can make a npda accepting $L'=L \cup \{a^nb^nc^n \mid n\geq 0\}$.

$\{a^nb^n\}$ DPDA $\{a^nb^{2n}\}$ DPDA $\{a^nb^n\} \cup \{a^nb^{2n}\}$ DPDA 라고 가정

This implies that L′ is a CFL, but this is not true. (L′ is not a CFL.)

Thus, L is not a deterministic context-free language.

- DPDA and NPDA are not equivalent.



Grammars for Deterministic CFL(1/2)

- Why deterministic CFL?
  - For any input, corresponding transition is at most one. DPDA
  - Parsing is efficient.

- Parsing process
  - Scan w from left to right while developing a sentential form whose terminal prefix matches the prefix of w up to scanned symbols.
  - For any current input symbol, correct production can be deterministically chosen.

Input w :	a_1	a_2	a_3	a_4	…	a_n	Scan from left to right
Sentential form :	a_1	a_2	a_3	A	…		
	matched part			yet to be matched			

current position

→ 바로 그 다음 production이 선택.

$A \to a_4 x$ (S-grammar)

Grammars for Deterministic CFL(2/2)

- Grammar suitable for the description of DCFL
  - S-grammar (A→ax·)
  - LL(k)grammar

S-grammar· LL(1) grammar·

The correct production is identified, given the currently scanned symbol and a "look-ahead" of the next k-1 symbols.

총 k개의 symbol.

$a^n b^n$

Ex7.12] S → aSb | ab

"LL(2) grammar"

$S \overset{②}{\Rightarrow} aSb \overset{③}{\Rightarrow} aaSbb$

$① \Rightarrow ab \quad ① \Rightarrow aabb$

Ex7.13] S → SS | aSb | ab

SS가 맞냐 해야 할지 결정할 수 X

$L = \{a^n b^n a^m b^m \cdots\}$

S → aSbS | λ, S_0 → aSbS

current aabbab ← parsing (a → ①, b → ②) (LL(1) grammar)

parsing이 → 편한 형태

$S_0 \Rightarrow aSbS \overset{①}{\Rightarrow} aaSbSbS \overset{②}{\Rightarrow} aabSbS \overset{②}{\Rightarrow} aabbS \overset{①}{\Rightarrow} aabbaSbS \overset{②}{\Rightarrow} aabbabS \overset{②}{\Rightarrow} aabbab.$

Grammars for Deterministic CFL(2/2)

- Grammar suitable for the description of DCFL
  - S-grammar
  - LL(k)grammar

    The correct production is identified, given the currently scanned symbol and a "look-ahead" of the next k-1 symbols.

Ex7.12] $S \rightarrow aSb \mid ab$

Ex7.13] $S \rightarrow SS \mid aSb \mid ab$

$S \rightarrow aSbS \mid \lambda$, $S_0 \rightarrow aSbS$

Grammars for Deterministic CFL(2/2)

- Grammar suitable for the description of DCFL
  - S-grammar
  - LL(k)grammar

    The correct production is identified, given the currently scanned symbol and a “look-ahead” of the next k-1 symbols.

Ex7.12] $S \rightarrow aSb \mid ab$

Ex7.13] $S \rightarrow SS \mid aSb \mid ab$

$S \rightarrow aSbS \mid \lambda$, $S_0 \rightarrow aSbS$