



# Push Down Automata (2)

Kuliah Teori Bahasa dan Automata  
Program Studi Ilmu Komputer  
Fasilkom UI

Prepared by:  
Rahmad Mahendra

# Membangun PDA dari Grammar

- Teorema

Diberikan suatu CFG  $G = (V, \Sigma, R, S)$ , terdapat sebuah PDA  $M$  sehingga  $L(M) = L(G)$

- Pembuktian dengan cara konstruksi (dua alternatif cara)

- *Top-down parsing*

Mulai dari  $S$ , menerapkan sejumlah *rule*  $R$ , dan memeriksa apakah ada derivasi dari  $G$  ke  $w$

- *Bottom-up parsing*

Mulai dari  $w$ , menerapkan sejumlah *rule*  $R$  *backward*, dan memeriksa apakah  $S$  dicapai

# CFG ke PDA top down

- Algoritma *CFGtoPDAtopdown*

Diberikan CFG  $G = (V, \Sigma, R, S)$

PDA  $M = (\{p, q\}, \Sigma, V, \Delta, p, \{q\})$  dapat dibentuk, yang mana  $\Delta$  mengandung:

- Transisi  $((p, \varepsilon, \varepsilon), (q, S))$ , *push start symbol S ke stack* dan pindah ke *state q*
- Transisi  $((q, \varepsilon, X), (q, \gamma_1\gamma_2 \dots \gamma_n))$  untuk setiap *rule*  
 $X \rightarrow \gamma_1\gamma_2 \dots \gamma_n$
- Transisi  $((q, c, c), (q, \varepsilon))$  untuk setiap  $c \in \Sigma$

# CFG ke PDA bottom up

- Algoritma *CFGtoPDAbottomup*

Diberikan CFG  $G = (V, \Sigma, R, S)$

PDA  $M = (\{p, q\}, \Sigma, V, \Delta, p, \{q\})$  dapat dibentuk, yang mana  $\Delta$  mengandung:

- Transisi *shift*  $((p, c, \varepsilon), (p, c))$  untuk setiap  $c \in \Sigma$
- Transisi *reduce*  $((p, \varepsilon, (\gamma_1\gamma_2 \dots \gamma_n)^R), (p, X))$  untuk setiap rule  $X \rightarrow \gamma_1\gamma_2 \dots \gamma_n$
- Transisi  $((p, \varepsilon, S), (q, \varepsilon))$

- *Shift-reduce parser*

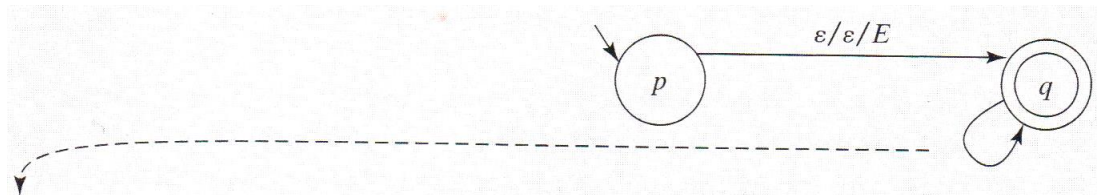
# Contoh-1

- $G = (\{E, T, F, \text{id}, +, *, (, )\}, \{\text{id}, +, *, (, )\}, R, E)$   
 $R = \{$   
$$E \rightarrow E+T \mid T$$
$$T \rightarrow T*F \mid F$$
$$F \rightarrow (E) \mid \text{id}$$
$$\}$$

Buatlah PDA yang ekuivalen dengan grammar di atas!

# Contoh-1 (top-down)

- PDA yang dibentuk dengan *top-down parsing*



(1)  $(q, \varepsilon, E), (q, E + T)$

(2)  $(q, \varepsilon, E), (q, T)$

(3)  $(q, \varepsilon, T), (q, T * F)$

(4)  $(q, \varepsilon, T), (q, F)$

(5)  $(q, \varepsilon, F), (q, (E))$

(6)  $(q, \varepsilon, F), (q, id)$

(7)  $(q, id, id), (q, \varepsilon)$

(8)  $(q, (, ( ), (q, \varepsilon)$

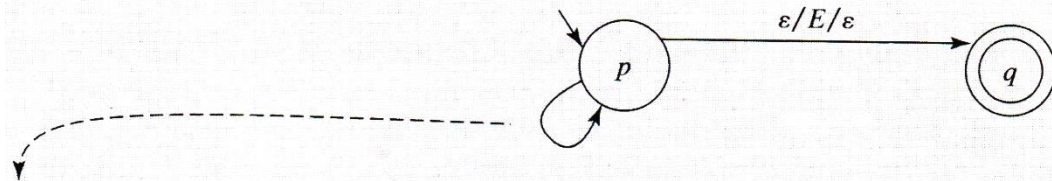
(9)  $(q, ), ) ), (q, \varepsilon)$

(10)  $(q, +, +), (q, \varepsilon)$

(11)  $(q, *, *), (q, \varepsilon)$

# Contoh-1 (bottom-up)

- PDA yang dibentuk dengan *bottom-up parsing*



- (1)  $(p, \text{id}, \varepsilon), (p, \text{id})$
- (2)  $(p, (, \varepsilon), (p, ($
- (3)  $(p, ), \varepsilon), (p, ))$
- (4)  $(p, +, \varepsilon), (p, +)$
- (5)  $(p, *, \varepsilon), (p, *)$
- (6)  $(p, \varepsilon, T + E), (p, E)$
- (7)  $(p, \varepsilon, T), (p, E)$
- (8)  $(p, \varepsilon, F * T), (p, T)$
- (9)  $(p, \varepsilon, F), (p, T)$
- (10)  $(p, \varepsilon, )E ( ), (p, F)$
- (11)  $(p, \varepsilon, \text{id}), (p, F)$



# Restricted Normal Form

- Suatu PDA  $M$  *restricted normal form* **iff**
  - $M$  memiliki *start state*  $s$  yang hanya melakukan *push* karakter spesial ( $\#$ ) ke dalam *stack*, kemudian pindah ke *state* berikut di mana komputasi dimulai.
  - $M$  memiliki *accepting state* tunggal  $a$ . Seluruh transisi menuju  $a$  tidak membaca input dan melakukan *pop*  $\#$  dari *stack*.
  - Setiap transisi dalam  $M$ , kecuali transisi dari  $s$ , *pop* tepat satu simbol dari *stack*.



# Algoritma *PDAtoRestricted*

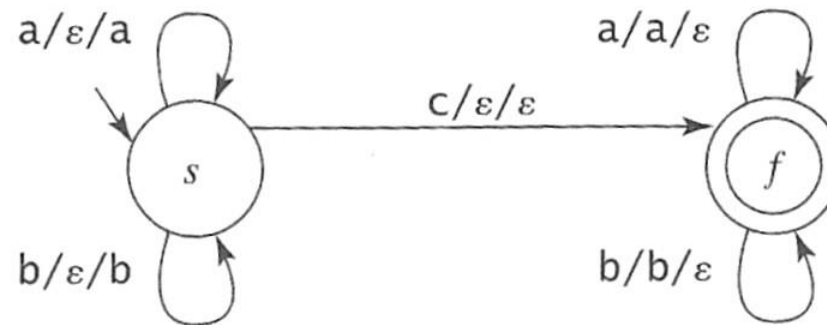
- Input: PDA  $M = (K, \Sigma, \Gamma, \Delta, s, \{q_1, q_2, \dots, q_n\})$
  - Output : PDA *restricted normal form*  $M'$
1.  $M' = M$
  2. Buat start state baru  $s'$  untuk  $M'$
  3. Tambahkan transisi  $((s', \varepsilon, \varepsilon), (s, \#))$
  4. Buat accepting state baru  $a$  untuk  $M'$
  5. Untuk setiap state  $q_k$ 
    - a. Tambahkan transisi  $((q, \varepsilon, \#), (a, \varepsilon))$
    - b. Ubah  $q_k$  menjadi non-accepting state

# *PDA to Restricted* (lanjutan)

6. Untuk setiap transisi yang *pop*  $k$  simbol,  $k > 1$ , ganti dengan sejumlah  $k$  transisi yang masing-masing *pop* tepat satu simbol tunggal
  - a. Tambahkan  $(k-1)$  state baru, misalkan state  $qq_1, qq_2, \dots, qq_{k-1}$
  - b. Ganti transisi  $((q_1, c, \gamma_1\gamma_2 \dots \gamma_n), (q_2, \gamma_P))$  dengan
    - $((q_1, \varepsilon, \gamma_1), (qq_1, \varepsilon)),$
    - $((qq_1, \varepsilon, \gamma_2), (qq_2, \varepsilon)),$
    - $\dots,$
    - $((qq_{k-1}, c, \gamma_n), (q_2, \gamma_P))$
7. Untuk setiap transisi  $((q_1, c, \varepsilon), (q_2, \gamma))$ , ganti dengan sejumlah  $|\Gamma \cup \{\#\}|$  transisi
  - a. Untuk setiap simbol  $\alpha \in \Gamma \cup \{\#\}$ , tambahkan transisi  $((q_1, c, \alpha), (q_2, \gamma\alpha))$
  - b. Hapus transisi  $((q_1, c, \varepsilon), (q_2, \gamma))$

## Contoh-2

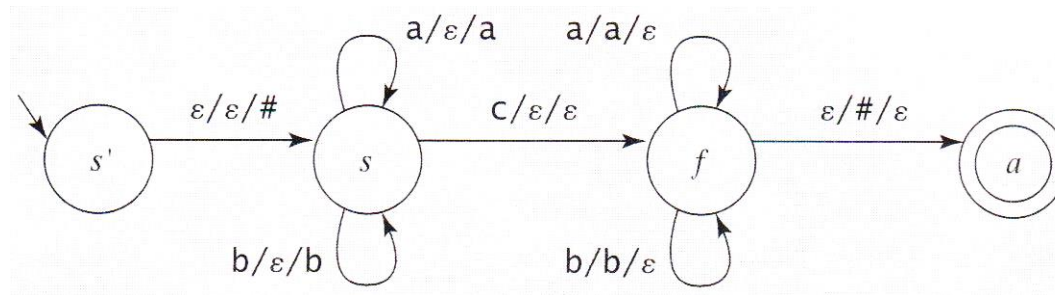
- $WcW^R = \{wcw^R: w \in \{a, b\}^*\}$



Konversi PDA di atas ke dalam *restricted normal form*!

# Contoh-2 (lanjutan)

- Buat *start state* dan *accepting state* yang baru



- Tidak ada transisi pada  $M$  yang melakukan *pop* lebih dari satu simbol
- 3 transisi dari  $s$  yang tidak melakukan *pop* diubah menjadi 9 transisi sebagai berikut

$((s, a, \#), (s, a\#))$ ,  $\#((s, a, a), (s, aa))$ ,  $\#((s, a, b), (s, ab))$ ,  
 $((s, b, \#), (s, b\#))$ ,  $\#((s, b, a), (s, ba))$ ,  $\#((s, b, b), (s, bb))$ ,  
 $((s, c, \#), (f, \#))$ ,  $\#((s, c, a), (f, a))$ ,  $\#((s, c, b), (f, b))$ .

# Membangun Grammar dari PDA

- Teorema

Diberikan suatu PDA  $M = (K, \Sigma, \Gamma, \Delta, s, A)$ , terdapat sebuah CFG  $G$  sehingga  $L(G) = L(M)$

- Pembuktian dengan cara konstruksi

- Ubah PDA ke dalam *restricted normal form*
- Bangun grammar

# Algoritma *PDAtoCFG*

- Input: PDA *restricted normal form*  $M$
- Output yang diharapkan: CFG  $G = (V_G, \Sigma_G, R, S)$ 
  1.  $\Sigma_G = \Sigma_M$
  2. Start symbol  $G = S$
  3. Rule  $R$ 
    - a.  $s'$  adalah *start state*,  $s$  adalah state yang dituju dari  $s'$  dan  $a$  adalah accepting state  
Tambahkan rule  $S \rightarrow \langle s, \#, a \rangle$
    - b. Untuk setiap transisi yang tidak *push* simbol apapun  $((q, c, \gamma), (r, \varepsilon))$  dan setiap state  $w$  selain  $s'$   
Tambahkan rule  $\langle q, \gamma, w \rangle \rightarrow c \langle r, \varepsilon, w \rangle$
    - c. Untuk setiap transisi yang *push* satu simbol tunggal  $((q, c, \gamma), (r, \alpha))$  dan setiap state  $w$  selain  $s'$   
Tambahkan rule  $\langle q, \gamma, w \rangle \rightarrow c \langle r, \alpha, w \rangle$

# Algoritma *PDA to CFG* (lanjutan)

## 3. Rule R

- d. Untuk setiap transisi yang *push* dua simbol sekaligus  $((q, c, \gamma), (r, \alpha\beta))$  serta setiap pasangan state  $v$  dan  $w$  ( $v \neq s', w \neq s'$ )

Tambahkan rule  $\langle q, \gamma, w \rangle \rightarrow c \langle r, \alpha, v \rangle \langle v, \beta, w \rangle$

- e. Dengan cara yang sama dengan d, tambahkan rule untuk transisi yang *push* lebih dari dua simbol.
- f. Untuk setiap state selain  $s'$ ,

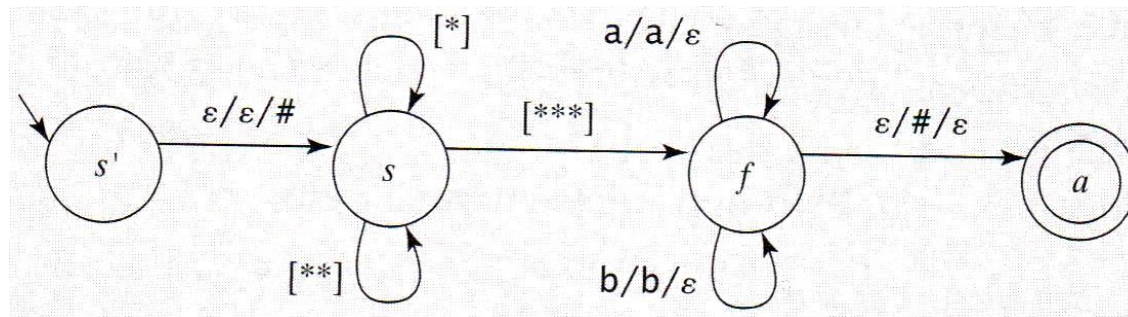
Tambahkan rule  $\langle q, \varepsilon, q \rangle \rightarrow \varepsilon$

- 4.  $V_G = \Sigma_M \cup$  himpunan simbol non terminal dalam rule  $R$  yang ditambahkan pada langkah 3



# Contoh-2 (lanjutan)

- PDA *restricted normal form* untuk  $WcW^R$



- Panah yang dilabeli dengan asterisk bermakna

$[*] ((s, a, \#), (s, a\#)), ((s, a, a), (s, aa)), ((s, a, b), (s, ab)),$   
 $[**] ((s, b, \#), (s, b\#)), ((s, b, a), (s, ba)), ((s, b, b), (s, bb)),$   
 $[***] ((s, c, \#), (f, \#)), ((s, c, a), (f, a)), ((s, c, b), (f, b)).$

# Contoh-2 (lanjutan)

		$S \rightarrow \langle s, \#, a \rangle$	[1]
	$((s', \varepsilon, \varepsilon), (s, \#))$	no rules based on the transition from $s''$	
[*]	$((s, a, \#), (s, a\#))$	$\langle s, \#, s \rangle \rightarrow a \langle s, a, s \rangle \langle s, \#, s \rangle$	[x]
		$\langle s, \#, s \rangle \rightarrow a \langle s, a, f \rangle \langle f, \#, s \rangle$	[x]
		$\langle s, \#, s \rangle \rightarrow a \langle s, a, a \rangle \langle a, \#, s \rangle$	[x]
		$\langle s, \#, f \rangle \rightarrow a \langle s, a, s \rangle \langle s, \#, f \rangle$	[x]
		$\langle s, \#, f \rangle \rightarrow a \langle s, a, f \rangle \langle f, \#, f \rangle$	[x]
		$\langle s, \#, f \rangle \rightarrow a \langle s, a, a \rangle \langle a, \#, f \rangle$	[x]
		$\langle s, \#, a \rangle \rightarrow a \langle s, a, s \rangle \langle s, \#, a \rangle$	[x]
		$\langle s, \#, a \rangle \rightarrow a \langle s, a, f \rangle \langle f, \#, a \rangle$	[2]
		$\langle s, \#, a \rangle \rightarrow a \langle s, a, a \rangle \langle a, \#, a \rangle$	[x]



# Contoh-2 (lanjutan)

	$((s, a, a), (s, aa))$	$\langle s, a, f \rangle \rightarrow a \langle s, a, f \rangle \langle f, a, f \rangle$	[3]
	$((s, a, b), (s, ab))$	$\langle s, b, f \rangle \rightarrow a \langle s, a, f \rangle \langle f, b, f \rangle$	[14]
[**]	$((s, b, \#), (s, b\#))$	$\langle s, \#, f \rangle \rightarrow b \langle s, b, f \rangle \langle f, \#, f \rangle$	[15]
	$((s, b, a), (s, ba))$	$\langle s, a, f \rangle \rightarrow b \langle s, b, f \rangle \langle f, a, f \rangle$	[4]
	$((s, b, b), (s, bb))$	$\langle s, b, f \rangle \rightarrow b \langle s, b, f \rangle \langle f, b, f \rangle$	[16]
[***]	$((s, c, \#), (f, \#))$	$\langle s, \#, f \rangle \rightarrow c \langle f, \#, f \rangle$	[17]
	$((s, c, a), (f, a))$	$\langle s, a, f \rangle \rightarrow c \langle f, a, f \rangle$	[18]
	$((s, c, b), (f, b))$	$\langle s, b, f \rangle \rightarrow c \langle f, b, f \rangle$	[5]
	$((f, \varepsilon, \#), (a, \varepsilon))$	$\langle f, \#, a \rangle \rightarrow \varepsilon \langle a, \varepsilon, a \rangle$	[6]
	$((f, a, a), (f, \varepsilon))$	$\langle f, a, f \rangle \rightarrow a \langle f, \varepsilon, f \rangle$	[7]
	$((f, b, b), (f, \varepsilon))$	$\langle f, b, f \rangle \rightarrow b \langle f, \varepsilon, f \rangle$	[8]
		$\langle s, \varepsilon, s \rangle \rightarrow \varepsilon$	[19]
		$\langle f, \varepsilon, f \rangle \rightarrow \varepsilon$	[9]
		$\langle a, \varepsilon, a \rangle \rightarrow \varepsilon$	[10]

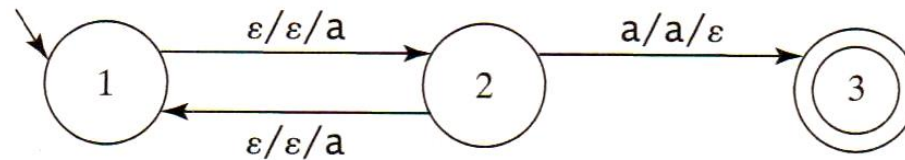
## Contoh 2 (lanjutan)

- Pada halaman 17 slide ini, beberapa rule hasil konversi dari PDA ditandai [x], artinya non produktif atau *unreachable* (review slide 10 : CFG-1)
- Contoh:  $\langle s, \#, a \rangle \rightarrow a \langle s, a, a \rangle \langle a, \#, a \rangle$  non produktif karena simbol  $\langle s, a, a \rangle$  dan  $\langle a, \#, a \rangle$  tidak bisa menurunkan simbol terminal (rule tersebut tidak dapat digunakan untuk derivasi string)
- Catatan: rule-rule yang ditampilkan pada halaman 18 hanya yang produktif saja.
- Contoh: dari transisi  $((s, a, a), (s, aa))$  bisa dibuat 9 rule seperti langkah yang ditunjukkan pada halaman 17. Namun, hanya rule  $\langle s, a, f \rangle \rightarrow a \langle s, a, f \rangle \langle f, a, f \rangle$  yang produktif.

# Non Determinisme dan Halting

- Komputasi  $C$  pada  $M$  *halt* **iff** salahsatu kondisi terpenuhi
  - $C$  adalah *accepting computation*
  - $C$  berakhir dalam suatu konfigurasi di mana tidak ada transisi dalam  $\Delta$  yang dapat diterapkan
- PDA mungkin tidak *halt* atau tidak pernah selesai membaca input.
- Tidak ada algoritma untuk minimisasi PDA.

# Contoh-3



- $L(M) = \{a\}$
- Komputasi  $(1, a, \varepsilon) \vdash_M (2, a, a) \vdash_M (3, \varepsilon, \varepsilon)$  menyebabkan string  $a$  diterima
- Akan tetapi, PDA tidak menerima juga tidak menolak string selain  $a$
- Contoh: PDA tidak pernah selesai membaca input  $aa$   
 $(1, aa, \varepsilon) \vdash_M (2, aa, a) \vdash_M (1, aa, aa) \vdash_M (2, aa, aaa)$   
 $\vdash_M (1, aa, aaaa) \vdash_M (2, aa, aaaaa) \vdash_M \dots$