



# Grammar, Bahasa dan Hirarki Chomsky

Kuliah Teori Bahasa dan Automata  
Program Studi Ilmu Komputer  
Fasilkom UI

Prepared by:

Rahmad Mahendra

Revised by:

Suryana Setiawan

# Review: Grammar

- Salah satu model komputasi adalah *rewrite system* / *production system* / *rule-based system*.
- Grammar menyatakan suatu *rewrite system* yang digunakan untuk mendefinisikan bahasa.
- Grammar  $G$  dapat ditulis sebagai quadruple  $(V, \Sigma, R, S)$ :
  - $V$ : alfabet *rule* yang terdiri dari simbol non-terminal dan terminal.
  - $\Sigma$ : himpunan simbol terminal (subset dari  $V$ ).
  - $R$ : himpunan *rule* dengan bentuk umum  $\alpha \rightarrow \beta$
  - $S$ : simbol start, dengan  $S \in (V - \Sigma)$ .

# Review: Grammar

- Alfabet  $V$  pada grammar  $G$  dibagi menjadi dua subset.
  - Alfabet terminal ( $\Sigma$ ), simbol yang membentuk string-string pada  $L(G)$
  - Alfabet non-terminal, elemen yang berfungsi sebagai *working symbols* yang akan digunakan ketika *grammar* dioperasikan.
- Rule  $R$  pada grammar  $G$  berbentuk  $\alpha \rightarrow \beta$ 
  - $\alpha$  disebut sebagai *left-hand side*, dan  $\beta$  *right-hand side*
  - Bagaimana ketentuan  $\alpha$  dan  $\beta$  pada regular grammar?

# Pembentukan String pada Grammar

- Diberikan grammar  $G$ , relasi biner *derives in-one-step*,  $x \Rightarrow_G y$  sebagai berikut
  - $\forall x, y \in V^*$  ( $x \Rightarrow_G y$  iff  $x = \alpha A \beta$ ,  $y = \alpha \gamma \beta$ , dan terdapat *rule*  $A \rightarrow \gamma$ )
- Bentuk  $w_0 \Rightarrow_G w_1 \Rightarrow_G w_2 \Rightarrow_G \dots \Rightarrow_G w_n$  disebut derivasi pada  $G$ .
- $\Rightarrow_G^*$  disebut relasi *derive*.
- $\Rightarrow_G^*$  merupakan penutup refleksif transitif dari  $\Rightarrow_G$ .  
**Mengapa?**
- Bahasa yang dapat dibangkitkan oleh (*generated by*)  $G$  adalah  $L(G) = \{w \in \Sigma^* : S \Rightarrow_G^* w\}$ .  
 $L(G)$  adalah himpunan seluruh kemungkinan string yang dapat diturunkan dari  $S$  dengan menerapkan serangkaian *rule* pada grammar  $G$ .

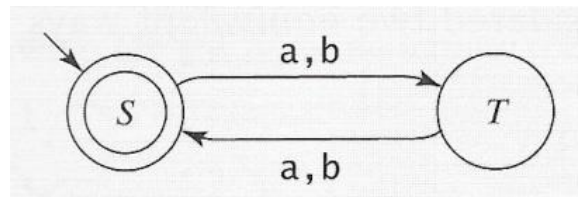
# Regular Grammar

- Bahasa  $L$  adalah *bahasa reguler* jika dan hanya jika  $L$  dapat dibentuk oleh suatu *regular grammar* (RG)  $G$ .
- Regular grammar adalah grammar dengan setiap rule-nya mengambil salah satu dari 3 format rule berikut
  - $A \rightarrow a$
  - $A \rightarrow aB$
  - $A \rightarrow \varepsilon$

dengan  $A, B \in (V - \Sigma)$  dan  $a \in \Sigma$ , dan  $\varepsilon$  string kosong

# Contoh

- $L = \{w \in \{a, b\}^* : |w| \text{ bilangan genap}\}$
- Ekspresi reguler untuk  $L$  adalah  $(aa \cup ab \cup ba \cup bb)^*$
- FSM untuk  $L$  adalah



- RG untuk  $L$  adalah:
  - $S \rightarrow \varepsilon \quad S \rightarrow aT \quad S \rightarrow bT$   
 $T \rightarrow aS \quad T \rightarrow bS$
  - Dapat disingkat:  
 $S \rightarrow \varepsilon \mid aT \mid bT \quad T \rightarrow aS \mid bS$
- String  $aaba$  dibentuk oleh grammar sbb:  
 $S \Rightarrow aT \Rightarrow aaS \Rightarrow aabT \Rightarrow aabaS \Rightarrow aaba$

# Context-Free Grammar

- Bahasa  $L$  adalah *context-free* jika dan hanya jika  $L$  dapat dibentuk oleh suatu *context-free grammar* (CFG)  $G$ .
- Pada CFG, *left-hand side* pada setiap *rule* harus berupa simbol non-terminal tunggal. Sedangkan *right-hand side* bisa berupa urutan simbol apapun (non-terminal maupun terminal, boleh string kosong).
  - Apa hubungan antara CFG dengan regular grammar?
- Contoh rule yang valid pada CFG
  - $S \rightarrow a$                        $S \rightarrow bSb$
  - $S \rightarrow T$                        $S \rightarrow aaSSbT$
- Contoh rule yang **tidak valid** pada CFG
  - $aSb \rightarrow aTb$                $a \rightarrow \varepsilon$
  - $ST \rightarrow bb$

# Contoh 1

- Bahasa berupa himpunan string-string yang dibentuk dari alfabet  $\Sigma = \{a, b\}$  di mana frekuensi kemunculan simbol 'a' sama dengan simbol 'b'
- $L = \{w \in \{a, b\}^* : \#_a w = \#_b w\}$
- Rule CFG untuk  $L$  adalah:

$$S \rightarrow aSb$$

$$S \rightarrow SS$$

$$S \rightarrow bSa$$

$$S \rightarrow \varepsilon$$

- String “ba”, “aabb”, dan “abbaba” dibentuk oleh CFG dengan proses derivasi sebagai berikut:

$$S \Rightarrow bSa \Rightarrow ba$$

$$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aabb$$

$$S \Rightarrow SS \Rightarrow aSbbSa \Rightarrow abbaSba \Rightarrow aababa$$



# Contoh 2

- *Balanced parentheses language*
- $Bal = \{w \in \{(), \{\}^*: \text{tanda kurung seimbang}\}$
- Rule CFG untuk  $L$  adalah:

$$S \rightarrow (S)$$

$$S \rightarrow SS$$

$$S \rightarrow \varepsilon$$

Tunjukkan bahwa ekspresi di bawah ini merupakan anggota bahasa  $Bal$

- $()()$
- $((()((()()))))$

# Context Sensitive Grammar

- Context sensitive grammar  $G = (V, \Sigma, R, S)$  adalah suatu unrestricted grammar di mana  $R$  memenuhi batasan:
  - Left-hand side (LHS) setiap rule mengandung paling sedikit satu simbol non terminal.
  - Jika  $R$  mengandung *rule*  $S \rightarrow \varepsilon$ , maka  $S$  tidak muncul pada right-hand side (RHS) *rule* manapun.
  - Kecuali  $S \rightarrow \varepsilon$ , setiap *rule*  $\alpha \rightarrow \beta$  pada  $R$  harus memenuhi properti  $|\alpha| \leq |\beta|$ .

# Contoh 1 (lanjutan)

- $A^nB^nC^n = \{a^n b^n c^n : n \geq 0\}$
- Grammar yang disajikan pada halaman 3 BUKAN context-sensitive grammar.
- $A^nB^nC^n$  bisa di-generate dari context-sensitive grammar dengan rule sebagai berikut

$$S \rightarrow T \mid \varepsilon$$

$$T \rightarrow aTBc \mid abc$$

$$cB \rightarrow WB$$

$$WB \rightarrow WX$$

$$WX \rightarrow BX$$

$$BX \rightarrow Bc$$

$$bB \rightarrow bb$$

- Tunjukkan derivasi string “aabbcc”!

# Bahasa Context-Sensitive

- Bahasa  $L$  context-sensitive jika  $\forall w \in L$ ,  $w$  bisa di-generate oleh suatu context-sensitive grammar.
- Adakah mesin spesifik yang menerima bahasa context-sensitive?
  - Linear Bounded Automata (LBA)
- LBA adalah mesin Turing di mana tape dibatasi oleh panjang input.
  - LBA adalah mesin Turing yang tidak menggunakan lebih dari  $k \cdot |w|$  tape square, di mana  $w$  adalah input dan  $k$  adalah suatu integer positif (*fixed*).

# Unrestricted Grammar

- Unrestricted grammar  $G$  dapat ditulis sebagai quadruple  $(V, \Sigma, R, S)$ :
  - $V$ : alfabet *rule* yang terdiri dari simbol non-terminal dan terminal.
  - $\Sigma$ : himpunan simbol terminal (subset dari  $V$ ).
  - $R$ : himpunan *rule* dengan bentuk umum  $V^+ \times V^*$
  - $S$ : simbol start, dengan  $S \in (V - \Sigma)$ .
- Right-hand side (RHS) pada unrestricted grammar boleh mengandung lebih dari satu simbol.
- Teorema:

Suatu bahasa  $L$  dapat dibentuk oleh suatu unrestricted grammar **iff** terdapat suatu mesin Turing semi-decide  $L$

# Contoh 1

- $A^n B^n C^n = \{a^n b^n c^n : n \geq 0\}$
- Rule unrestricted grammar untuk  $A^n B^n C^n$  adalah:

$$S \rightarrow aBSc$$

$$S \rightarrow \varepsilon$$

$$Ba \rightarrow aB$$

$$Bb \rightarrow bb$$

$$Bc \rightarrow bc$$

- Derivasi string “abc” dan “aabbcc” sebagai berikut:

$$S \Rightarrow aBSc \Rightarrow aBc \Rightarrow abc$$

$$S \Rightarrow aB\underline{S}c \Rightarrow aB\underline{aB\underline{S}cc} \Rightarrow a\underline{B}a\underline{B}cc \Rightarrow aa\underline{B}\underline{B}cc \Rightarrow aa\underline{B}\underline{b}cc \Rightarrow aabbccc$$

# Contoh 2

- Diberikan rule unrestricted grammar  $G$  sebagai berikut:

$S \rightarrow T\#$  /\* tandai akhir string

$T \rightarrow aTa$  /\* generate  $wCw^R$

$T \rightarrow bTb$

$T \rightarrow C$

$C \rightarrow CP$  /\* generate pusher

$Paa \rightarrow aPa$  /\* push karakter ke kanan

$Pab \rightarrow aPb$

$Pba \rightarrow bPa$

$Pbb \rightarrow bPb$

$Pa\# \rightarrow \#a$  /\* mencapai akhir string

$Pb\# \rightarrow \#b$

$C\# \rightarrow \varepsilon$

- Definisikan  $L(G)$

# Contoh 2 (lanjutan)

- $L(G) = \{ww : w \in \{a, b\}^*\}$
- Strategi generate  $ww$ 
  - Generate  $wCw^R\#$   
di mana  $C$  = temporary middle marker dan  $\#$  = temporary right boundary
  - Reverse  $w^R$
  - Remove  $C$  dan  $\#$
- Tunjukkan derivasi untuk string berikut
  - $\epsilon$
  - abab
  - babbbabb



# Chomsky Hierarchy

- Basis dari Chomsky Hierarchy adalah jumlah dan organisasi memori yang diperlukan untuk memproses suatu bahasa.
  - Type 0 (semi-decidable): tidak ada batasan memori
  - Type-1 (context-sensitive): memori terbatas pada panjang string
  - Type-2 (context-free): memori tidak terbatas tetapi hanya dapat diakses melalui mekanisme state
  - Type-3 (regular): memori berhingga

# Chomsky Hierarchy

