



Membangun Grammar

Kuliah Teori Bahasa dan Automata
Program Studi Ilmu Komputer
Fasilkom UI

Prepared by:
Suryana Setiawan

Revised by:



Regular Grammar

- Dikenal juga sebagai linear grammar.
- 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 ε string kosong

- **Teorema:** Bahasa L adalah *bahasa reguler* jika dan hanya jika L dapat dibentuk oleh suatu *regular grammar* (RG) G .



Grammar Reguler dan Bahasa Reguler

- Bahasa yang dapat degenerate oleh grammar regular adalah bersifat regular.
 - $A \rightarrow \varepsilon \mid a \mid aB$
 - Variabel A menghasilkan bahasa sederhana $\{\varepsilon\}$ atau $\{a\}$ atau $\{a\}L_B$, jika L_B dihasilkan B .
 - $A \rightarrow \alpha \mid \beta$
 - Variabel A menghasilkan $L_A = L_\alpha \cup L_\beta$ jika L_α dihasilkan α dan L_β dihasilkan β .
- Implikasi
 - $A \rightarrow \varepsilon \mid aA$
 - Variabel A menghasilkan $L_A = \{a\}^*$.
 - $A \rightarrow a \mid aA$
 - Variabel A menghasilkan $L_A = \{a\}^+$.

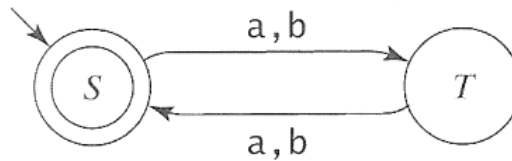


DFSM \rightarrow Grammar Regular

- Dari suatu DFSM M , grammar regular dari Bahasa yang diterima M dapat dibentuk sebagai berikut:
 - $V = K$
 - Start symbol S adalah start state dalam M .
 - Untuk setiap transisi $((A, c), B)$ buat rule $A \rightarrow cB$
 - Untuk setiap accepting state E buat rule $E \rightarrow \varepsilon$.
 - Menambahkan Σ ke dalam V .
- Catatan: mesin M harus deterministic karena transisi- ε tidak ada padanan langsungnya (jika ada, harus dihilangkan dahulu dengan konversi NDFSM \rightarrow DFSM).

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

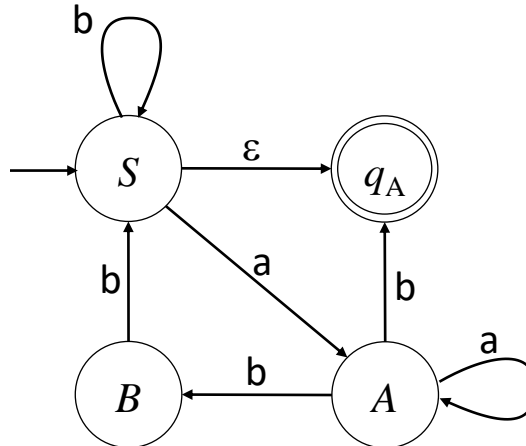


Grammar Regular \rightarrow NDFSM

- Dari suatu grammar regular G , suatu mesin M yang dapat menerima bahasa yang dihasilkan G dapat dibentuk sebagai berikut:
 - $K = (V - \Sigma) \cup \{q_A\}$ dengan q_A adalah accepting state.
 - Start state S start symbol dalam V .
 - Untuk setiap rule $A \rightarrow cB$ buat transisi $((A, c), B)$.
 - Untuk setiap rule $A \rightarrow c$ buat transisi $((A, c), q_A)$.
 - Untuk setiap rule $A \rightarrow \varepsilon$ buat transisi $((A, \varepsilon), q_A)$.
- Note: Mesin yang dihasilkan adalah NDFSM karena ada kemungkinan grammar berisi $A \rightarrow cB \mid cD$, terjadinya undefined transition (implisit ke dead-state), dan transisi- ε .

Contoh

- Gramamr untuk L mempunyai rule-rule:
 - $S \rightarrow bS \mid aA \mid \varepsilon$
 $A \rightarrow aA \mid bB \mid b$
 $B \rightarrow bS$
- Mesin yang menerima Bahasa L adalah:





Latihan

- Dapatkan grammar untuk $\neg L$ jika grammar untuk L adalah (grammar yang sama dari contoh sebelumnya).
 - $S \rightarrow bS \mid aA \mid \varepsilon$
 $A \rightarrow aA \mid bB \mid b$
 $B \rightarrow bS$



Varian lain Untuk Grammar Reguler

- Definisi grammar regular yang telah dibahas adalah varian **right linear grammar**.
- **Left linear grammar** didefinisikan dengan rule-rule berformat:
 - $A \rightarrow Bb$
 - $A \rightarrow b$
 - $A \rightarrow \varepsilon$
- Perbedaan “hanya” pada saat meng-generate string maka simbol string ditulis dari belakang.
- *Pertanyaan: apakah bahasa yang dihasilkan grammar varian ini juga regular?*