Automata dan Teori Bahasa

Pertemuan 2.3
Formal Language

Formal Language

- Adalah sekumpulan kata, yaitu string simbol terbatas yang diambil dari alfabet untuk mendefinisikan bahasa.

Jika

- $\Sigma_1 = \{ 0, 1 \}$
- $\Sigma_2 = \{ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 \}$
- $\Sigma_3 = \{ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C , D, E , F \}$
- $\Sigma_4 = \{ a, b, c, . . . , z \}$

Maka

- $1010 \in \Sigma_1$
- $123 \in \Sigma_2$
- $hello \in \Sigma_4$

Notasi

- ε : empty string
- v , w , x, y , z, : denote strings

Panjang

- |w | adalah panjang dari w
  - |a| = 1
  - |125| = 3
  - |ε| = 0

k-th power

- $\Sigma^k = \{a_1 \cdots a_k \mid a_1, \ldots, a_k \in \Sigma\}$
  - $\Sigma^0 = \{\varepsilon\}$ for any Σ
  - $\Sigma_1^1 = \{0, 1\}$
  - $\Sigma_1^2 = \{00, 01, 10, 11\}$

Kleene closures

$$\Sigma^* \stackrel{\text{def}}{=} \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \cdots = \bigcup_{i=0}^{\infty} \Sigma^i$$

$$\Sigma^+ \stackrel{\text{def}}{=} \Sigma^* \setminus \{\varepsilon\} = \bigcup_{i=1}^{\infty} \Sigma^i$$

String operations

- Concatenation → dibagi menjadi
- Substring → bagian dari
- Prefix → berada di awal
- Suffix → berada di akhir

Operasi pada Language

- $L_1 \cup L_2 = \{w \mid w \in L_1 \vee w \in L_2\}$
- $\overline{L_1} = \{w \in \Sigma_1^* \mid w \notin L_1\}$
- $L_1 L_2 = \{w_1 w_2 \mid w_1 \in L_1 \wedge w_2 \in L_2\}$
- $L_1^* = \{\varepsilon\} \cup L_1 \cup L_1^2 \cup \cdots$
- $L_1 \cap L_2 = \{w \mid w \in L_1 \wedge w \in L_2\}$

Grammar

- Rule : Subject → Verb → Object
  - Subject : Saya, Dia, Mereka
  - Verb : Makan, Membeli, Menjual
  - Object : Jeruk, Semangka

Grammar dalam Tuple

- Tuple $G = (V, T, S, P)$
  - V is a finite, non-empty set of symbols called variables (or non-terminals or syntactic categories)
  - T is an alphabet of symbols called terminals
  - $S \in V$ is the start (or initial) symbol of the grammar
  - P is a finite set of productions $\alpha \rightarrow \beta$ where $\alpha \in (V \cup T)^{+}$ and $\beta \in (V \cup T)^{*}$

Contoh

- V = {Sentence, Subject, Verb, Object},
- T = {I, You, Eat, Buy, Pen, Apple},
- S = {Sentence},
- P = {Sentence → Subject Verb Object, Subject → I | You, Verb → Eat | Buy, Object → Pen | Apple}

Notasi

- $A, B, C, \ldots \in V$ represent non-terminal symbols
- $a, b, c, \ldots \in T$ represent terminal symbols
- $X, Y, Z, \ldots \in V \cup T$ represent generic symbols
- $u, v, w, x, \ldots \in T^*$ are strings over T
- $\alpha, \beta, \gamma, \delta, \ldots \in (V \cup T)^*$ are strings over $V \cup T$

Contoh 1

- Grammar G1 – ({S, A, B}, {a, b}, S, {S $\rightarrow$ AB, A $\rightarrow$ a, B $\rightarrow$ b})
  - S, A, and B are Non-terminal symbols;
  - a and b are Terminal symbols
  - S is the Start symbol, S $\in$ V
  - Productions, P : S $\rightarrow$ AB, A $\rightarrow$ a, B $\rightarrow$ b

Contoh 2

- Grammar G2 − (({S, A}, {a, b}, S,{S $\rightarrow$ aAb, aA $\rightarrow$ aaAb, A $\rightarrow$ ε })
  - S and A are Non-terminal symbols.
  - a and b are Terminal symbols.
  - ε is an empty string.
  - S is the Start symbol, S $\in$ N
  - Production P : S $\rightarrow$ aAb, aA $\rightarrow$ aaAb, A $\rightarrow$ ε

Derivations

- String dapat diturunkan dari string lain menggunakan produksi dalam Grammar.
- Jika grammar G memiliki production $\alpha \rightarrow \beta$, kita dapat mengatakan bahwa x α y menurunkan x β y dalam G.
- $x \alpha y \Rightarrow G\ x \beta y$

Contoh

- G2 = ({S, A}, {a, b}, S, {S $\rightarrow$ aAb, aA $\rightarrow$ aaAb, A $\rightarrow$ ε })
  - S $\Rightarrow$ aAb using production S $\rightarrow$ aAb
  - $\Rightarrow$ aaAbb using production aA $\rightarrow$ aAb
  - $\Rightarrow$ aaaAbbb using production aA $\rightarrow$ aAb
  - $\Rightarrow$ aaabbb using production A $\rightarrow$ ε

Klasifikasi Grammar

General	no restrictions
Monotone	$\alpha \to \beta \Rightarrow \lvert\alpha\rvert \leq \lvert\beta\rvert$
Context-dependent	$\gamma A \delta \to \gamma \beta \delta$
Context-free	$A \to \beta$
Linear	$A \to uBv$
Right-Linear	$A \to wB$
Left-Linear	$A \to Bw$

Derived Grammar

- Bahasa yang dihasilkan oleh Grammar G adalah subset yang ditentukan secara formal oleh :
  - $L(G)=\{W|W \in \sum^*, S \Rightarrow G\ W\}$

Contoh 1

- G: N = {S, A, B} T = {a, b} P = {S $\rightarrow$ AB, A $\rightarrow$ a, B $\rightarrow$ b}
- Here S produces AB, and we can replace A by a, and B by b. Here, the only accepted string is ab,
- L(G) = {ab}

Contoh 2

- G: $N = \{S, A, B\}$ $T = \{a, b\}$
  $P = \{S \rightarrow AB, A \rightarrow aA|a, B \rightarrow bB|b\}$
- $L(G) = \{ab, a^2b, ab^2, a^2b^2, \ldots\ldots\ldots\}$
  - $= \{a^m b^n \mid m \geq 1 \text{ and } n \geq 1\}$

Contoh 3

- S → aS , S → B, B → b and B → bB
- S → B → b (Accepted)
- S → B → bB → bb (Accepted)
- S → aS → aB → ab (Accepted)
- S → aS → aaS → aaB → aab(Accepted)
- S → aS → aB → abB → abb (Accepted)

Klasifikasi Chomsky

Chomsky	Grammars	Automata	Languages
Type-0	unrestricted	Turing	L_0: recursively enumerable[a]
Type-1	context-sensitive	linear-bounded[b]	L_1: context-sensitive[c] $(a^n b^n c^n)$
Type-2	context-free	pushdown	L_2: context-free[d] $(a^n b^n)$
Type-3	r./l.-linear	finite-state	L_3: regular[e] $(a^n b^m)$

$$L_3 \subset L_2 \subset L_1 \subset L_0 \subset \wp(T^*)\ ^f$$