Kg. strukdis.

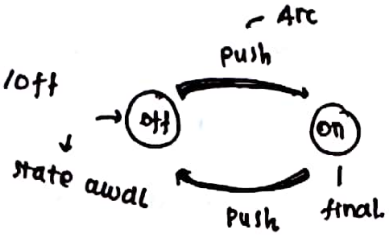
Teori Otomata & DFA.

- Teori otomata: studi tentang peralatan / "mesin" komputasi abstrak.
- Finite Automata (FA)

  model yg berguna untuk macam** perangkat keras & lunak

  contoh:

1. model switch on/off



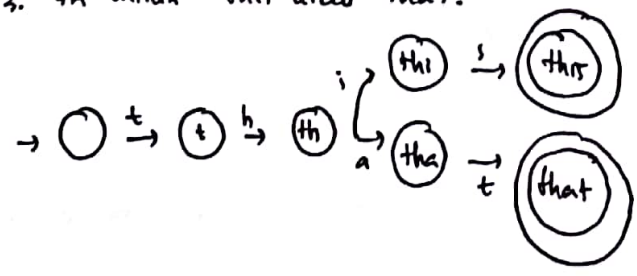
- Komponen dalam FA.

1. State. → dinyatakan oleh lingkaran
2. Arc → diantara state diberi label "input" yg menyatakan pengaruh eksternal pada sistem
3. Start state / initial state
   - state dimana sistem berada dalam keadaan awal
4. Final / accepting state.
   - dinyatakan dalam lingkaran ganda

contoh:

2. FA untuk mengenali kata "then"

3. FA untuk this atau that.



- Grammar.

digunakan dalam perancangan perangkat lunak yg memproses data dg struktur rekursif.

- contoh aturan gramatikal.

$$E \rightarrow E + E.$$

ekspresi dapat dibentuk dari dua ekspresi yg dihubungkan dg tanda +.

- Ekspresi Regular

↳ menyatakan struktur dari data, khususnya string teks.

contoh: `[A-Z][a-z]* [ ] [A-Z][A-Z]`

(`[ ]` ↓ spasi. `[A-Z]` ↓ range karakter.)

- Ithaca NY ✓
- I NY ✓
- Alto Palo CA ✗.

catatan:

* → menandakan boleh null, ≥ 0 boleh berulang

\+ → minimal 1 karakter, tdk boleh null.

[ ] → spasi

( ) → untuk mengelompokkan ekspresi

- Konsep dalam teori Otomata.

1. Alphabet.

   himpunan berhingga dan tak kosong.

   simbol Σ.

   contoh:

   - $\Sigma = \{0, 1\}$, alphabet biner.
   - $\Sigma = \{a, b, \dots, z\}$ himpunan huruf kecil.

2 String.

↳ deretan simbol berhingga yg dapat dipilih dari alphabet.

ε → epsilon → string kosong.

contoh: 011011 & 1111 → string dari alphabet biner $\Sigma = \{0,1\}$

✓ Panjang string

contoh: 01101 → panjangnya 5.

- notasi standar untuk panjang string w adalah $|w|$

contoh: $|011| = 3$, $|\varepsilon| = 0$.

- x adalah substring dari string lain y jika ada string w dan z, keduanya dapat berupa string kosong.

$y = wxz$.

contoh: car (w) — substring dari :

1. carry. = y
   (w bisa x/z/xz.)

cat: intinya y itu gabungan 3 string, boleh null

✓ Pangkat dan Alphabet.

- $\Sigma^0 = \{\varepsilon\}$, untuk alphabet apapun
- Jika $\Sigma = \{0,1\}$ maka

$\Sigma^1 = \{0,1\}$

$\Sigma^2 = \{00, 01, 10, 11\}$

$\Sigma^3 = \{000, 001, 010, 011, 100, 101, 110, 111\}$ dst.

- semua string pada alphabet Σ dinotasikan Σ^*.

- Σ^+ → tidak ada himpunan kosong pada string.

sehingga: $\Sigma^+ = \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cup \dots$

$\Sigma^* = \Sigma^+ \cup \{\varepsilon\}$

✓ Perangkaian String.

misal x dan y adalah string, maka xy perangkaian dari x dan y,

- x disusun oleh i simbol

$x = a_1 a_2 .. a_i$

y disusun oleh j simbol

$y = b_1 b_2 ... b_j$

maka xy string dg panjang i+j

$xy = a_1 a_2 \dots a_i b_1 b_2 \dots b_j$

- untuk string w,

$\varepsilon w = w\varepsilon = w$ dipenuhi.

✓ Bahasa

↳ String yg diperlih dari Σ^*, dimana Σ adalah alphabet. $L \subseteq \Sigma^*$, L adalah bahasa pada Σ.

contoh:

1. Bahasa dari semua string yg berisi n buah 0 dan diikuti oleh n buah 1, n≥0
   → artinya, jumlah 0 dan 1 nya sama.

$\{\varepsilon, 01, 0011, 000111, \dots\}$

ϕ → bahasa kosong.

$\{\varepsilon\}$ → bahasa yg mengandung string kosong.

$\phi \neq \{\varepsilon\}$

Kg. Struktur

contoh:

Bahasa pada $\Sigma = \{a, b\}$:

1. $L_1 = \{\varepsilon, a, aa, aab\}$

2. $L_2 = \{x \in \{a,b\}^* \mid |x| \leq 8\}$

↓

maksudnya kombinasi a dan b.
maksimum panjang stringnya 8.

Rekursif

- Definisi Rekursif dari L^*:
  - a) $\varepsilon \in L^*$
  - b) untuk $x \in L^*$ dan suatu $y \in L$, $xy \in L^*$.

contoh:

1. Diberikan definisi rekursif untuk bahasa L pada $\Sigma = \{0,1\}$ berikut:
   - $\varepsilon \in L$.
   - untuk setiap $x \in L$, $0x$ dan $0x1$ adalah dalam L.

   Jawab: $\{\varepsilon, 0, 01, 00, 001, 0011, \ldots\}$

   $x = \varepsilon \rightarrow 0\varepsilon$ dan $0\varepsilon1 = 0$ dan 01

   $x = 0 \rightarrow 00$ dan (001) — tdk ditulis 2 kali karena sama.

   $x = 01 \rightarrow 001$ dan 0011

   ⋮

   dst.

Latihan:

1. contoh string dalam bahasa $L = \{bxb \mid x \in \{a,b\}^*\}$
   - bb ✓
   - abbb ✗ salah karena tdk drawali b.
   - bbbab ✓ (x = bab)
   - baaba ✗ tidak diakhiri b.
   - babab ✓ (x = bab)

2. diberikan bahasa $L = \{001, 0001, 00001, 000001, 0000011, 00000011, 000000011, \ldots\}$

   Tentukan bentuk umum dari semua string dalam bahasa tersebut!

   Jawab: $L = \{0^i 1^j \mid i \geq 2j\}$

3. diberikan bahasa $L = \{01, 0011, 001, 00111, 00011, 0001, 000011111, 00001111, 000011, 00001, \ldots\}$

   tentukan bentuk umum dari semua string dalam bahasa tersebut!

   $L = \{0^i 1^j \mid i \geq j > 0\}$

4. Diberikan bahasa $L = \{xx, xyx, xxx, xyxyx, xyyyyx, xxxxxx, xyyxxx, \ldots\}$ tentukan bentuk umum dari semua string dalam bahasa tersebut!

   Jawab: $L = \{xax \mid a \in \{x,y\}^*\}$

Perangkaian Bahasa

Jika L_1, L_2 bahasa, L_1 dan $L_2 \subseteq \Sigma^*$

notasi:

$L_1L_2 = \{xy \mid x \in L_1 \text{ dan } y \in L_2\}$

contoh:

{hope, fear} {less, fully} = {hopeless, hopefully, fearless, fearfully}.

$L\{\varepsilon\} = \{\varepsilon\}L$ karena $x\varepsilon = \varepsilon x = x$.

Deterministic Finite Automata (DFA)

DFA akan berada pada suatu state tunggal setelah pembacaan dari serangkaian input.

Definisi

DFA terdiri dari :

1. Q → state
2. Σ → simbol input. } himpunan berhingga.
3. Fungsi transisi (δ)

   Untuk $q \in Q$ dan $a \in \Sigma$.

   $$\delta : Q \times \Sigma \to Q$$

   misal :
   $\delta(q,a)$ = state p → terdapat arc yg diberi label a dari q ke p.
   q = state
   a = simbol input
4. state awal
5. Final state / accepting state

Notasi : !!

$A = (Q, \Sigma, \delta, q_0, F)$

dengan : A = nama DFA
Q = himpunan state
Σ = simbol atau input
δ = Fungsi transisi
q_0 = start state
F = himpunan accepting state.

Pemrosesan String Oleh DFA.

misal : $a_1, a_2, \ldots a_n$ → adalah rangkaian dari simbol input.

- $\delta(q_0, a_1) = q_1$
  ↳ artinya, berawal dari q_0 dg dibaca nya simbol input a_1, menuju q_1.
- Jika $q_n \in F$ maka input diterima.

contoh :

1. $A = (\{q_0, q_1, q_2\}, \{0,1\}, \delta, q_0, \{q_1\})$

   dimana δ :

   $\delta(q_0,0) = q_2$ $\delta(q_0,1) = q_0$

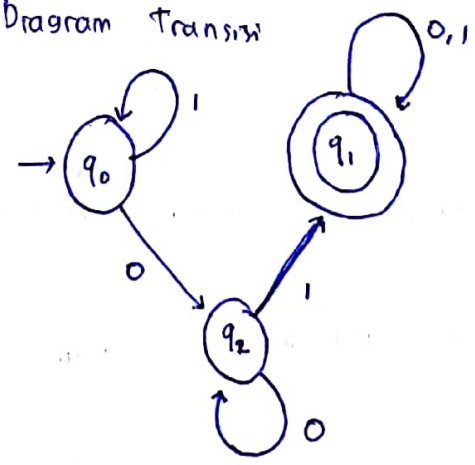
   $\delta(q_2,0) = q_2$ $\delta(q_2,1) = q_1$

   $\delta(q_1,0) = \delta(q_1,1) = q_1$

Tabel transisi

Σ	0	1
→ q_0	q_2	q_0
q_1	q_1	q_1
q_2	q_2	q_1

Diagram Transisi



contoh : 1010110 → diterima.

Perluasan Fungsi Transisi Untuk String.

notasi :

$\hat{\delta}$ adalah fungsi transisi yg diperluas

- Basis

$\hat{\delta}(q, \varepsilon) = q$. berarti jika FA berada dalam state q & membaca tak ada input, FA tetap dalam state q.

- Induksi

anggap w = string dg bentuk xa,
a adalah simbol terakhir dari w.
x adalah semua simbol kecuali yg akhir.

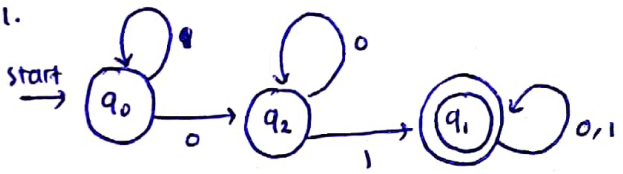
contoh: $w = 1101$ ← $x = 110$, $a = 1$

maka:

$$\hat{\delta}(q, w) = \delta(\hat{\delta}(q, x), a)$$

contoh:

1.



misal: 010

Basis : $\hat{\delta}(q_0, \varepsilon) = q_0$

Induksi

- $w = 0 \rightarrow x = \varepsilon$, $a = 0$

$\hat{\delta}(q_0, w) = \hat{\delta}(q, \varepsilon 0) = \delta(\hat{\delta}(q_0, \varepsilon), 0)$

$= \delta(q_0, 0)$

$= q_2$

- $w = 01 \rightarrow x = \varepsilon 0$, $a = 1$

$\hat{\delta}(q_0, w) = \hat{\delta}(q, \varepsilon 01) = \delta(\hat{\delta}(q_0, \varepsilon 0), 1)$

$= \delta(q_2, 1)$

$= q_1$

- $w = 010 \rightarrow x = \varepsilon 01$, $a = 0$

$\hat{\delta}(q_0, \varepsilon 010) = \delta(\hat{\delta}(q_0, \varepsilon 01), 0)$

$= \delta(q_1, 0)$

$= q_1$

$\therefore \hat{\delta}(q_0, 010)\ q_1 \in F.$

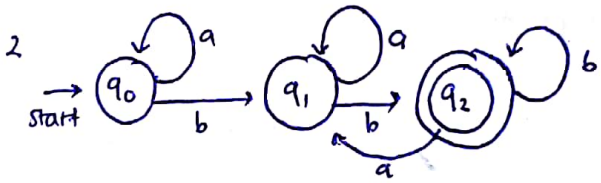
↳ karena menuju final state

- Bahasa dari DFA.

$L(A) = \{ w \mid \hat{\delta}(q_0, w)$ dalam $F \}$

$L(A) \subseteq \Sigma^*$

2



$\hat{\delta}(q_0, \varepsilon) = q_0$ misal: abb.

- Induksi

$w = a \rightarrow x = \varepsilon$, $a = a$

$\hat{\delta}(q_0, w) = \hat{\delta}(q, \varepsilon a) = \delta(\hat{\delta}(q_0, \varepsilon), a)$

$= \delta(q_0, a)$

$= q_0$

- $w = ab \rightarrow x = \varepsilon a$, $a = b$

$\hat{\delta}(q_0, w) = \hat{\delta}(q, \varepsilon ab)$

$= \delta(\hat{\delta}(q_0, \varepsilon a), b)$

$= \delta(q_0, b)$

$= q_1$

- $w = abb \rightarrow x = \varepsilon ab$, $a = b$

$\hat{\delta}(q_0, w) = \hat{\delta}(q, \varepsilon abb)$

$= \delta(\hat{\delta}(q_0, ab), b)$

$= \delta(q_1, b)$

$= q_2 \rightarrow$ accepting state

$\therefore$ diterima.

Nondeterministic Finite Automata

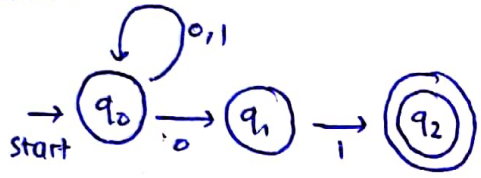
untuk NFA, δ boleh mengembalikan himpunan dari nol, satu / lebih state.

misal: $\delta(q_0, 0) = \emptyset$ → boleh.

$\delta(q_0, 0) = \{q_1, q_2\}$ → boleh

contoh:

1.



tabel transisi:

NFA $A = (Q, \Sigma, \delta, q_0, \{q_2\})$.

Σ	0	1
→ q_0	$\{q_0, q_1\}$	q_0
q_1	$\emptyset$	q_2
q_2	$\emptyset$	$\emptyset$

harus diakhiri 01.

misal: $w_1 = 01001$ ✓

$w_2 = 010111$ ✗.

Fungsi Transisi yg Diperluas.

Basis = $\hat{\delta}(q_0, \varepsilon) = q_0$ → $w = xa$.

Induksi = $\hat{\delta}(q, w) = \bigcup_{i=1} \delta(p_i, a)$,

$p_i \in \hat{\delta}(q, x)$

contoh:

$w = 101$

- Basis = $\hat{\delta}(q_0, \varepsilon) = q_0$
- Induksi:

- $w = \varepsilon 1 \rightarrow x = \varepsilon$, $a = 1$

$\hat{\delta}(q_0, \varepsilon 1) = \bigcup \delta(p_i, 1)$, $p_i \in \hat{\delta}(q_0, \varepsilon)$

$= \delta(q_0, 1)$ $\quad p_i \in \{q_0\}$

$= q_0$

- $w = \varepsilon 10$, $x = \varepsilon 1$, $a = 0$

$\hat{\delta}(q_0, \underbrace{\varepsilon 10}_{xa}) = \bigcup \delta(p_i, 0)$, $p_i \in \hat{\delta}(q_0, \varepsilon 1)$

$p_i \in \{q_0\}$

$= \delta(q_0, 0)$

$= \{q_0, q_1\}$

- $w = \varepsilon 101$, $x = \varepsilon 10$, $a = 1$

$\hat{\delta}(q_0, \varepsilon 101) = \bigcup \delta(p_i, 1)$, $p_i \in \hat{\delta}(q_0, \varepsilon 10)$

$= p_i \in \{q_0, q_1\}$

$= \delta(q_0, 1) \cup \delta(q_1, 1)$

$= q_0 \cup q_2$

$= \{q_0, q_2\}$

$\therefore \hat{\delta}(q_0, \varepsilon 101) = \{q_0, q_2\} \cap q_2$.

$\neq \emptyset$

diterima //.

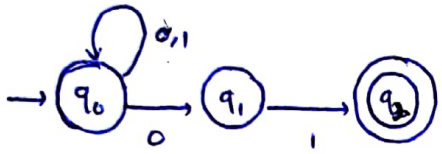
Ekuivalensi dari DFA & NFA.

untuk setiap $S \subseteq Q_N$ dan untuk setiap simbol input $a \in \Sigma$.

$$\delta_D(S,a) = \bigcup_{P \in S} \delta_N(P,a)$$

contoh:

1.



string diakhiri oleh 01.

	$Q_D \backslash \Sigma$	0	1
A	$\emptyset$	$\emptyset$	$\emptyset$
B	$\rightarrow \{q_0\}$	$\{q_0, q_1\}$	$\{q_0\}$
C	$\{q_1\}$	$\emptyset$	$\{q_2\}$
D	$*\{q_2\}$	$\emptyset$	$\emptyset$
E	$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_2\}$
F	$*\{q_0, q_2\}$	$\{q_0, q_1\}$	$\{q_0\}$
G	$\{q_1, q_2\}$	$\emptyset$	$\{q_2\}$
H	$\{q_0, q_1, q_2\}$	$\{q_0, q_1\}$	$\{q_0, q_2\}$

- $\delta_D(S,0) = \bigcup_{P \in \{q_0,q_1\}} \delta_N(P,a)$

  $= \delta(q_0,0) \cup \delta(q_1,0)$

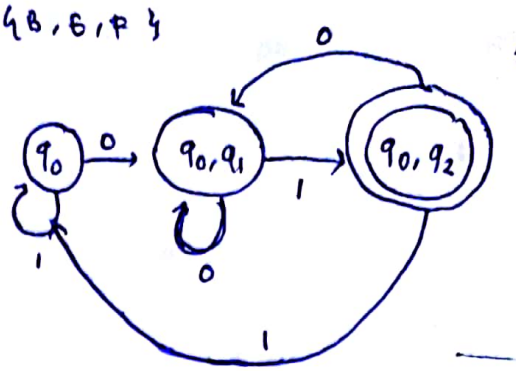
  $= \{q_0, q_1\} \cup \{\emptyset\} = \{q_0, q_1\}$

- $\delta_D(S = \{q_0,q_2\}, 0) = \bigcup_{P \in S} \delta_N(P,a)$

  $= \delta_N(q_0,0) \cup \delta_N(q_2,0)$

  $= \{q_0,q_1\} \cup \{\emptyset\} = \{q_0,q_1\}$

$\{B, G, P\}$



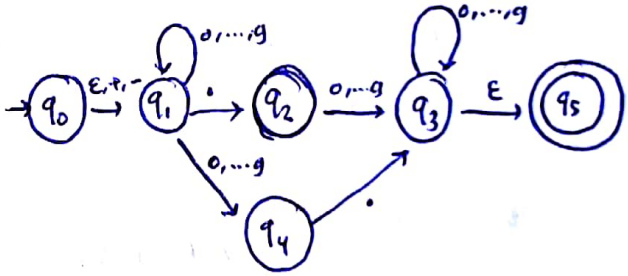
: NFA.

Epsilon NFA.

ε-NFA = $\{Q = \{q_0 \ldots q_5\}, \Sigma = \{+,-,\ldots,0,\ldots,9\}, \delta, q_0, F\{q_5\}\}$

contoh:

1. tanda (+) atau (−)
2. sebuah string dari digit
3. (.) titik desimal



uji: 4.5 → diterima

tabel transisi:

δ	ε	+, −	.	0, ..., 9.
$\rightarrow q_0$	$\{q_1\}$	$\{q_1\}$	$\emptyset$	$\emptyset$
q_1	$\emptyset$	$\emptyset$	$\{q_2\}$	$\{q_1, q_4\}$
q_2	$\emptyset$	$\emptyset$	$\emptyset$	$\{q_3\}$
q_3	$\{q_5\}$	$\emptyset$	$\emptyset$	$\{q_3\}$
q_4	$\emptyset$	$\emptyset$	$\{q_3\}$	$\emptyset$
$*q_5$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

- Eclose $(q_0) = \{q_0, q_1\}$ → yg mengandung ε
- Eclose $(q_1) = \{q_1\}$
- Eclose $(q_3) = \{q_3, q_5\}$

Fungsi Transisi yg Diperluas.

Basis: $\hat{\delta}(q, \varepsilon) = \text{Eclose}(q)$

Induksi: $w = xa$

misal $\{p_1, p_2, \dots p_k\}$ adalah $\hat{\delta}(q, x)$

misal $\bigcup_{i=1}^{k} \delta(p_i, a) = \{r_1, r_2, \dots r_m\}$

$$\hat{\delta}(q, w) = \bigcup_{j=1}^{m} \text{Eclose}(r_j)$$

contoh:

① $w = 5.6$

Basis: $\hat{\delta}(q_0, \varepsilon) = \text{Eclose}(q_0) = \{q_0, q_1\}$

Induksi: $w = \varepsilon 5$

- $\{p_1, p_2\} = \hat{\delta}(q_0, \varepsilon) = \{q_0, q_1\}$
- $\bigcup_{i=1}^{2} \delta(p_i, 5) = \delta(q_0, 5) \cup \delta(q_1, 5)$
  $= \emptyset \cup \{q_1, q_4\}$
  $= \{q_1, q_4\}$
- $\hat{\delta}(q_0, \varepsilon 5) = \bigcup_{r_j \in \{q_1, q_4\}} \text{Eclose}(r_j)$
  $= \{q_1\} \cup \{q_4\}$
  $= \{q_1, q_4\}$

$w = \varepsilon 5.$ → $x = \varepsilon 5$, $a = .$

- $\bigcup_{p_i \in \hat{\delta}(q_0, \varepsilon 5),\ p_i \in \{q_1, q_4\}} = \delta(q_1, .) \cup \delta(q_4, .)$
  $= \{q_2\} \cup \{q_3\}$
  $= \{q_2, q_3\}$
- $\hat{\delta}(q_0, \varepsilon 5.) = \bigcup_{r_j \in \{q_2, q_3\}} \text{eclose}(r_j)$
  $= \{q_2\} \cup \{q_3, q_5\}$
  $= \{q_2, q_3, q_5\}$

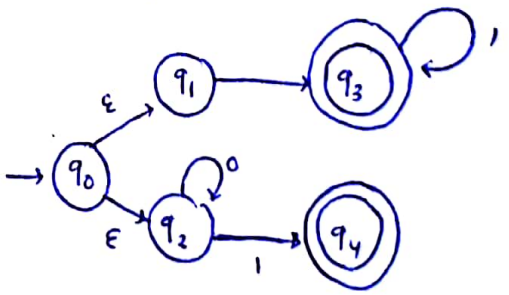
$w = \varepsilon 5.6$ → $x = \varepsilon 5.$, $a = 6.$

- $\bigcup_{p_i \in \{q_2, q_3, q_5\}} \delta(p_i, 6) = \{q_3\} \cup \{q_3\} \cup \emptyset$
  $= \{q_3\}$
- $\hat{\delta}(q_0, \varepsilon 5.6) = \bigcup_{r_j \in \{q_3\}} \text{eclose}(r_j)$
  $= \{q_3, q_5\}$

∴ $\hat{\delta}(q_0, 5.6) = \{q_3, q_5\} \cap \{q_4\}$
$\neq \emptyset$

diterima

latihan



a. tabel transisi:

	δ	ε	0	1
→	q_0	$\{q_1, q_2\}$	$\emptyset$	$\emptyset$
	q_1	$\emptyset$	$\{q_3\}$	$\emptyset$
	q_2	$\emptyset$	$\{q_2\}$	$\{q_4\}$
*	q_3	$\emptyset$	$\emptyset$	$\{q_3\}$
*	q_4	$\emptyset$	$\emptyset$	$\emptyset$

K-11

strukdis.

b. $w = 011$

Basis : $\hat{\delta}(q_0, \varepsilon) = E\text{close}(q_0) = \{q_0, q_1, q_2\}$

Induksi : $w = \varepsilon 0$

- $\{p_1, p_2\} = \hat{\delta}(q_0, \varepsilon) = \{q_0, q_1, q_2\}$
- $\bigcup_{i=1}^{3} \delta(p_i, 0) = \delta(q_0, 0) \cup \delta(q_1, 0) \cup \delta(q_2, 0)$

  $= \emptyset \cup \{q_3\} \cup \{q_2\}$

  $= \{q_2, q_3\}$
- $\hat{\delta}(q_0, \varepsilon 0) = \bigcup_{r_j \in \{q_2, q_3\}} \text{eclose}(r_j)$

  $= \{q_3\} \cup \{q_2\}$

  $= \{q_2, q_3\}$

$w = \varepsilon 01$ $\quad x = \varepsilon 0, \ a = 1$

$\bigcup_{p_i \in \hat{\delta}\{q_2, q_3\}} \delta(p_i, 1) = \{q_4\} \cup \{q_3\}$

$= \{q_3, q_4\}$

- $\hat{\delta}(q_0, \varepsilon 01) = \bigcup_{r_j \in \{q_3, q_4\}} E\text{close}(r_j)$

  $= \{q_3\} \cup \{q_4\}$

  $= \{q_3, q_4\}$

$w = \varepsilon 011$

$\bigcup_{p_i \in \{q_3, q_4\}} \delta(p_i, 1) = \{q_3\} \cup \emptyset$

$= \{q_3\}$

$\hat{\delta}(q_0, \varepsilon 011) = \bigcup_{r_j \in \{q_3\}} E\text{close}(r_j)$

$= \{q_3\} \cap \{q_3, q_4\}$

$\therefore$ diterima $\neq \emptyset$

v Ekuivalensi ε-NFA ke DFA.

misalkan $E = (Q_{\varepsilon\text{-NFA}}, \Sigma, \delta_{\varepsilon\text{-NFA}}, q_0, F_{\varepsilon\text{-NFA}})$

ekuival : $\Downarrow$

$D = (Q_D, \Sigma, \delta_D, q_D, F_D)$ — eclose (q_0).

Dengan syarat :

1. Q_D adalah himpunan dan subset Q_E, untuk $S \subseteq Q_E$, sedemikian sehingga $S = \text{eclose}(S)$.
2. q_D Eclose (q_0) ; (Basis)
3. $S_D(S, a)$ untuk $a \in \Sigma, S \in Q_D$

   misal $S = \{p_1, p_2, \dots, p_k\}$

   hitung $\bigcup_{i=1}^{k} S_E(p_i, a)$: misal

   himpunan $\{r_1, r_2, \dots r_m\}$

   maka $S_D(S, a) = \bigcup_{i=1}^{m} \text{eclose}(r_j)$.

Contoh :

① soal tadi.

$q_D = \text{eclose}(q_0) = \{q_0, q_1, q_2\}$ // start state

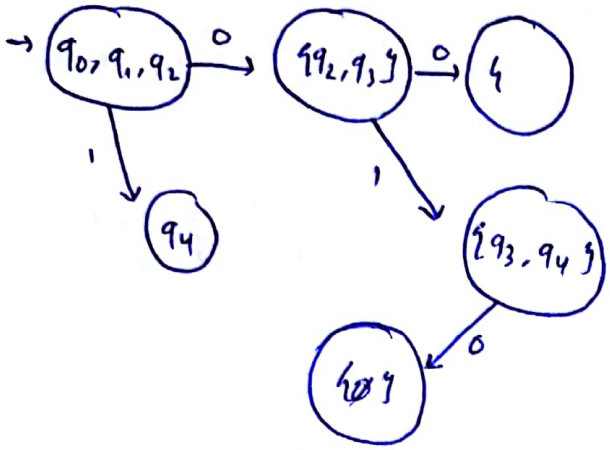
$S_D = (\{q_0, q_1\}, 1)$

$\bigcup_{p_i \in S} \delta_E(p_i, 1) = \emptyset$

$S_D(\{q_2, q_3\}, 1) = \bigcup_{r_j \in \{\emptyset\}} \text{eclose}(r_j)$

$= \emptyset$

DFAD.



K-12 strukdis.

Ekspresi Regular

- notasi lain dari NFA.
- menyatakan bahasa, berisi semua string yg diawali 0 dan diikuti oleh para 1, atau sebaliknya.

  contoh: $01^* + 10^*$

- operasi dalam bahasa.

1. Union.

   himpunan dari string yg ada dalam L atau M.

   contoh:

   $L = \{001, 10, 111\}$

   $M = \{\varepsilon, 001\}$

   Union = $\{\varepsilon, 10, 001, 111\}$

2. Concatenation.

   himpunan dari string yg dibentuk dg mengambil string dalam L dan dirangkai dg M

   notasi dg "dot" / tdk ada operator

   contoh:

   $L = \{001, 10, 111\}$

   $M = \{\varepsilon, 001\}$

   Maka: $LM = \{001, 10, 111, 001001, 10001, 111001\}$

3. Closure (atau star / kleene closure)

   notasi: L^*

   $$L^* = L^0 \cup L^1 \cup L^2 \ldots$$

   contoh:

   $L = \{0, 11\}$

   L^*

   $L^0 = \{\varepsilon\}$

   $L^1 = \{0, 11\}$

   $L^2 = \{00, 011, 110, 1111\}$

   ⋮

   dst.

   catatan:

   $$\phi^* = \{\varepsilon\} \text{ dan } \phi^0 = \{\varepsilon\}$$

- Hukum - hukum Aljabar

1. Asosiatif & Komutatif

   a. $L+M = M+L$

   b. $(L+M)+N = L+(M+N)$

   c. $(LM)N = L(MN)$

strukdis.

2 Identitas & Annihilator

a. $\phi + L = L + \phi = L$, ϕ = identitas unron.

b. $\varepsilon L = L\varepsilon = L$, ε identitas perangkaian.

c. $\phi L = L\phi = \phi$, ϕ annihilator u/ perangkaian.

3. Distributif.

a. $L(M+N) = LM + LN$

b. $(M+N)L = ML + NL$

4. Idempotent u/ unron.

$L+L = L$

5. a. $(L^*)^* = L^*$

b. $\phi^* = \varepsilon$

c $\varepsilon^* = \varepsilon$

d. $L^+ = LL^* = L^*L$

$L^+ = L + LL + LLL + \dots$

$L^* = \varepsilon + L + LL + \dots$

Jadi, $LL^* = L\varepsilon + LL + LLL + \dots$

e $L^* = L^+ + \varepsilon$

f. $L? = \varepsilon + L$, definisi dari operator?

contoh:

① $0 + 01^* = 0\varepsilon + 01^*$, 2b

$= 0(\varepsilon + 1^*)$, 3a

$= 01^*$, $\varepsilon + R = R$

✓ Membuat Ekspresi Regular

L dinyatakan secara rekursif terdiri dari basis dan induksi.

✓ Basis:

1. ε dan ϕ , Ekspresi Rugular.
2. Jika a simbol , a ER
3. sebuah variabel, dicetak tebal & miring , merepresentasikan bahasa.

✓ Induksi

1. Jika E, F ER .
   maka E+F ER
2. Jika E, F ER ,
   maka EF ER , boleh pake dot. (E.F)
3. Jika E ER ,
   maka E^* ER
4. Jika E ER , maka (E) ER.

contoh:

① L = 01, 0101, 010101, ... ⇒ $(01)^*$

$\Sigma = \{0, 1\}$

basis 2

a. 0, 1 simbol , $0 \in \Sigma$, $1 \in \Sigma$

0 ER , 1 ER

b. Induksi no.2.

0 ER , 1 ER ,

maka 0.1 ER

c Induksi no.3

01 ER , maka $(01)^*$ ER //

misal : 010 ✓

- induksi no.2 0ER, 1 ER → 10 ER.
- induksi no.3 : 10 ER → $(10)^*$ ER
- induksi 2 : $0(10)^*$ ER . //

(2.) $(0+\varepsilon)\,01^*$; ER

buktikan !.

Pernyataan	Keterangan
1. ε : ER	Basis 1
2. 0 : ER, 1 : ER	Basis 2
3. $0+\varepsilon$: ER	Induksi 1, P1, P2
4. $(0+\varepsilon)$: ER	Induksi 4, P3.
5. $(0+\varepsilon)0$: ER	Induksi 2, P3 P2
6. 1^* : ER	Induksi 3, P2
6. $(0+\varepsilon)01^*$: ER	Induksi 2, P4 P5.

urutan prioritas operator:

1. star
2. dot
   - bersifat asosiatif, misal $012 = (01)2$.
3. operator + (union) yg bersifat asosiatif.

contoh: $01^*+1 \rightarrow (0(1^*))+1$

Dari DFA ke Ekspresi Regular

Definisi u/ mengkontruksi $R_{ij}^{(k)}$.

Basis :

$k=0$.

2 macam path :

a. arc dari node (state) i ke j.

b. path dg panjang 0. hanya terdiri dari node i.

Jika $i \neq j$, hanya 1 kasus.

misal: input = 0.

transisi dari i ke j.

- tidak ada input (a), maka $R_{ij}^{(0)} = \emptyset$
- Jika terdapat 1 simbol a, maka $R_{ij}^{(0)} = a$.
- Jika terdapat $a_1, a_2, \dots a_k$ maka, $R_{ij}^{(0)} = a_1 + a_2 + \dots + a_k$

catatan :

Jika $i=j$ → panjang 0 dan semua loop dari i ke dirinya sendiri.

direpresentasikan oleh ε. ekspresi regular.

$$R_{ij}^{(k)} = R_{ij}^{(k-1)} + R_{ik}^{(k-1)} \left(R_{kk}^{(k-1)}\right)^* R_{kj}^{(k-1)}$$

$k=n$. → $R_{ij}^{(n)}$

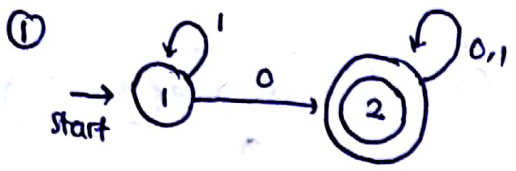
i = start state

j = accepting state.

Penyederhanaan.

- $(\varepsilon + R)^* = R^*$
- $(\varepsilon + 1)^* = 1^*$
- $\emptyset R = R\emptyset = \emptyset$
- $\emptyset + R = R + \emptyset = R$, $\emptyset$ identitas union

contoh:



$R_{ij}^{(0)}$	ER
$R_{11}^{(0)}$	$\varepsilon+1$
$R_{12}^{(0)}$	0
$R_{21}^{(0)}$	$\emptyset$
$R_{22}^{(0)}$	$\varepsilon+0+1$

$$R_{ij}^{(1)} = R_{ij}^{(0)} + R_{i1}^{(0)} (R_{11}^{(0)})^* R_{1j}^{(0)}$$

$R_{ij}^{(1)}$	ER
$R_{11}^{(1)}$	$= R_{11}^{(0)} + R_{11}^{(0)} (R_{11}^{(0)})^* R_{11}^{(0)}$
	$= (\varepsilon+1) + (\varepsilon+1)(\varepsilon+1)^* (\varepsilon+1)$
	$= 1^*$
$R_{12}^{(1)}$	$= R_{12}^{(0)} + R_{11}^{(0)} (R_{11}^{(0)})^* R_{12}^{(0)}$
	$= 0 + (\varepsilon+1)(\varepsilon+1)^* 0$
	$= 1^* 0$
$R_{21}^{(1)}$	$= \emptyset$
$R_{22}^{(1)}$	$= \varepsilon + 0 + 1$

$$R_{ij}^{(2)} = R_{ij}^{(1)} + R_{12}^{(1)} (R_{22}^{(1)})^* R_{21}^{(1)}$$

$R_{ij}^{(2)}$	ER.
$R_{11}^{(2)}$	$R_{11}^{(1)} + R_{12}^{(1)} (R_{22}^{(1)})^* R_{21}^{(1)}$
	$= 1^* + 1^* 0 (\varepsilon+0+1)^* \emptyset$
	$= 1^* + \emptyset$
	$= 1^*$

$R_{ij}^{(2)}$	ER
$R_{12}^{(2)}$	$1^* 0 (0+1)^*$ → hasil akhir
$R_{21}^{(2)}$	$\emptyset$
$R_{22}^{(2)}$	$(0+1)^*$

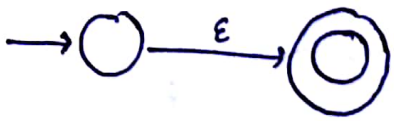
Konversi Ekspresi Regular ke Automata

Bukti:

1. tepat 1 accepting state
2. tdk ada arc ke dalam start state
3. tdk ada arc yg keluar dari accepting state

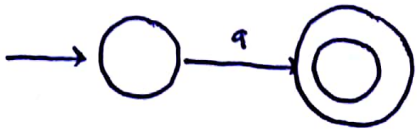
Basis:

a. konstruksi automaton yg menangani ekspresi ε



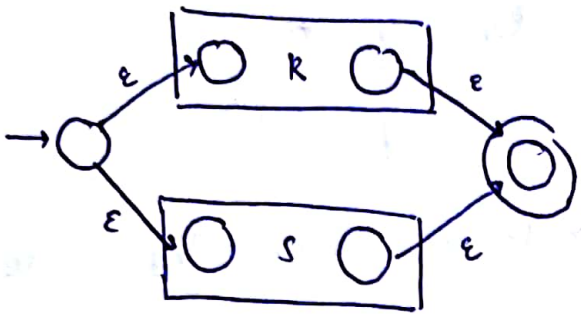
b. konstruksi untuk ϕ

c. konstruksi automaton untuk ekspresi regular a.



Induksi :

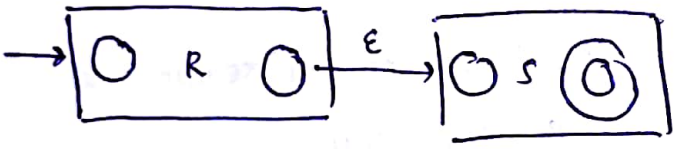
1. Ekspresi adalah R+S, untuk ekspresi yg lebih kecil R & S.



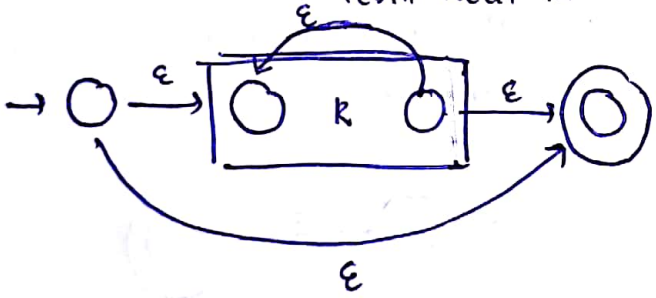
bahasa dari automaton adalah

$L(R) \cup L(S)$.

2. Ekspresi RS, untuk R & S yg lebih kecil.

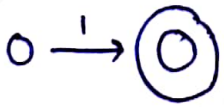


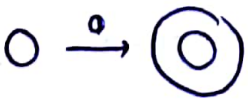
3. Ekspresi R^*, untuk ekspresi yg lebih kecil R.



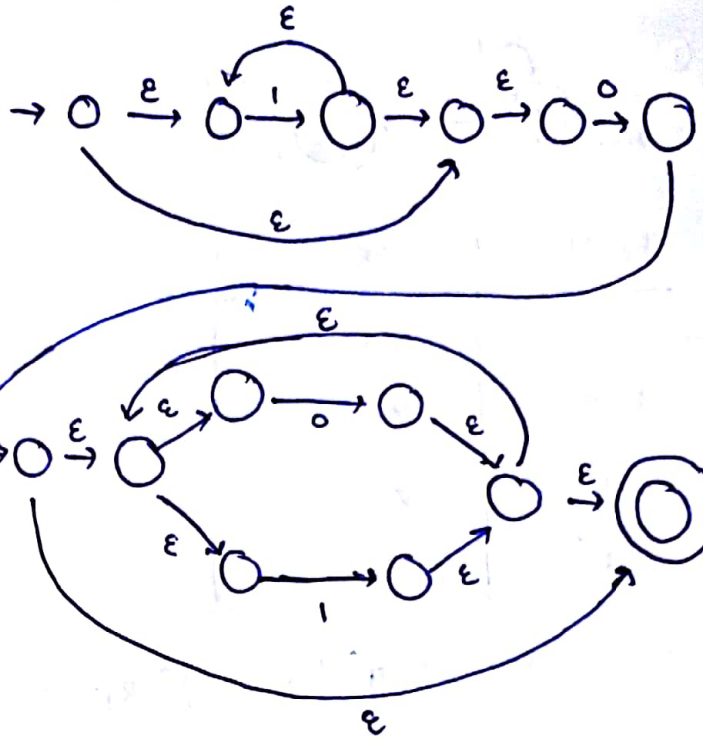
contoh:

① $1^*0(0+1)^*$

$1 \in R$ 

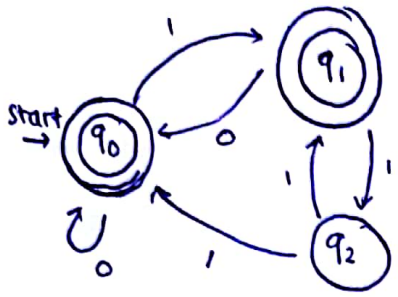
$0 \in R$ 

$1^+ \in R$



K.13. strukdrs.

Review.



$$R_{ij}^{(k)} = R_{ij}^{(k-1)} + R_{ik}^{(k-1)} \left(R_{kk}^{(k-1)}\right)^* R_{kj}^{(k-1)}$$

$R_{11}^{(0)}$	ER
$R_{00}^{(0)}$	$\varepsilon + 0$
$R_{01}^{(0)}$	1
$R_{02}^{(0)}$	$\emptyset$
$R_{10}^{(0)}$	0
$R_{11}^{(0)}$	ε
$R_{12}^{(0)}$	1
$R_{20}^{(0)}$	0
$R_{21}^{(0)}$	1
$R_{22}^{(0)}$	ε

$R_{ij}^{(1)}$	ER
$R_{00}^{(1)}$	$R_{00}^{(0)} + R_{01}^{(0)} \left(R_{11}^{(0)}\right)^* R_{10}^{(0)}$
	$= \varepsilon + 0 + 1 (\varepsilon)^* 0$
	$= \varepsilon + 0 + 10$
$R_{01}^{(1)}$	$R_{01}^{(0)} + R_{01}^{(0)} \left(R_{11}^{(0)}\right)^* R_{11}^{(0)}$
	$= 1 + 1 (\varepsilon)^* \varepsilon \quad = 1 + 1 = 1$
$R_{02}^{(1)}$	$R_{02}^{(0)} + R_{01}^{(0)} \left(R_{11}^{(0)}\right)^* R_{12}^{(0)}$
	$= \emptyset + 1 (\varepsilon)^* 1 = 11$

⋮ Lanjutin sendiri

CFG → context free Grammar.

Bahasa

- Regular : direpresentasikan oleh RE / DFA.
- non-regular : direpresentasikan oleh Statement logika,
  ↓
  dikenal dg CFG.

contoh: $\{0^n 1^n \mid n$: bil. bulat positif $\}$

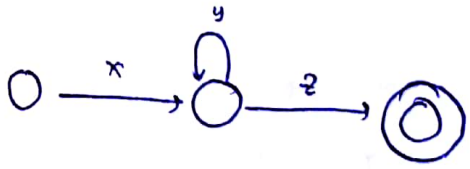
Pumping Lemma → u/ bahasa regular.

↳ u/ membuktikan suatu bahasa merupakan bahasa regular / bukan

$w = xyz$, panjang $\geq n$.

memenuhi :

1. $|xy| \leq n$
2. $|y| > 0$
3. untuk semua $i \geq 0$, $xy^i z$ di dalam L.



CFG: notasi formal untuk menyatakan definisi rekursif dari bahasa.

CFG terdiri atas :

- 1 / lebih variabel.
- bahasa
- rule yg menyatakan bagaimana string dikonstruksi
- pengkonstruksian dapat menggunakan simbol / string.

v Contoh Bahasa non-reguler.

• Palindrom.

suatu kata yg dibaca dari depan, dan belakang sama.

contoh : wadidaw , madam, 010 , dll.

contoh:

Rekursif dari Lpal = { ε, 0, 1, 00, 11, 000, 101, 010, 111 }

Basis: ε, 0, 1 ∈ Lpal.

Induksi:

① Jika x ∈ Lpal maka 0x0 ∈ Lpal
1x1 ∈ Lpal.

② Bahwa tidak ada string lain yang diambil selain yg ada dalam Lpal.

CFG

↳ 4 komponen notasi

G = (V, T, P, S)

V = himpunan variabel

T = terminal

P = himpunan produksi / rule

S = start symbol.

dari contoh diatas:

G = (V = {P}, Σ = {0,1}, A, S = P)

A = rule.

① P → ε

② P → 0

③ P → 1

④ P → 0P0

⑤ P → 1P1

misal : 1001 palindrom ?

P → 1P1 → 10P01 → 10ε01
↓
1001 ✔.

contoh :

v sebuah CFG menyatakan ekspresi dalam bahasa pemrograman, dengan ketentuan:

① Ekspresi dibatasi hanya mengandung operator penjumlahan & perkalian.

② Argumen juga dapat berupa Identifier yg dibatasi hanya untuk huruf a dan b, dan digit 0 dan 1.

③ Setiap identifier haruslah diawali dg a atau b dan dapat diikuti oleh string dalam {a, b, 0, 1}*.

contoh : → lanjutan.

① E → I
E → E+E
E → E*E
E → (E)
} A.

I → a | b | Ia | Ib | I0 | I1

G = (V = { E, I }, T = { +, *, 0, 1, a, b }, A, S = E)

A = yg diatas.
misal : (a1 + a0)

E → (E)
→ (E + E)
→ (I + E)
→ (I1 + E)
→ (a1 + E)
→ (a1 + I)
→ (a1 + I0)
→ (a1 + a0) //

karena kala S=I salah.

I → I1
→ a1 x

② $L_1 = \{ 0^n 1^n \mid n \geq 0 \}$
$L_1 = \{ \varepsilon, 01, 0011, 000111, \dots \}$

Definisi Rekursif.

1. Basis : $\varepsilon \in L_1$
2. Induksi
Jika $x \in L_1$ maka $0x1 \in L_1$
$x = \varepsilon \in L_1 \rightarrow 0\varepsilon1 \in L_1$

$G_1 = (V = \{Q\}, T = \{0,1\}, A, S = Q)$

A : Q → ε
Q → 0Q1

③ $L_2 = \{ a, (a), a+a, a*a, (a)+a, (a*a), a+a*a, \dots \}$

$G_2 = (V = \{E\}, T = \{+, *, a, (\}, A, S = E)$

A : E → a
E → (E)
E → E+E
E → E*E

misal : (a) * a
E → E * E
→ (E) * E
→ (a) * E
→ (a) * a //

Penurunan Menggunakan Grammar

2 pendekatan :

① inferensi rekursif
↳ menggunakan aturan body ke head

② Derivation
↳ menggunakan aturan head ke body.

✓ langkah - langkah inferensi rekursif

① ambil string* yg ada dalam bahasa dari setiap variabel dari body.

② Rangkai string** tsb.

③ simpulkan bahwa string yg dihasilkan adalah bahasa dari variabel pada head.

contoh:

① CFG : ① E → I
② E → E+E
3. E → E*E
4. E → (E)
5. I → a
6. I → b
7. I → Ia
8. I → Ib
9. I → I0
10 I → I1

misal: E → a* (a + boo) ✓

Baris	string hasil	Bahasa	produksi	string dipakai
1	a	I	5	-
2	b	I	6	-
3	b0	I	9	(2)
4	b00	I	9	(3)
5	a	E	1	(1)
6	b00	E	1	(4)
7	a+b00	E	2	(5) & (6)
8	(a+b00)	E	4	(7)
9	a*(a+b00)	E	3.	(8) & (5)

② dari soal tadi:

G_1 = S → ε
S → 0S1

misal : 0011

(I)	(II)	(III)	(IV)	(V)
1	ε	S	1	-
2	0ε1	S	2	(1)
3	0011	S	2	(2)

③ G_2 : E → a
E → (E)
E → E+E
E → E*E

misal : (a)* a.

(I)	(II)	(III)	(IV)	(V)
1	a	E	1	-
2	(a)	E	2	(1)
3	(a)* a	E	4	(2) & (1)

✓ nomor 1, cara lain:

$E \xrightarrow{3} E^*E \xrightarrow{1} I^*E \xrightarrow{5} a^*E \xrightarrow{4} a^*(E) \xrightarrow{2} a^*(E+E) \xrightarrow{1} a^*(I+E) \xrightarrow{5} a^*(a+E) \xrightarrow{1} a^*(a+I) \xrightarrow{9} a^*(a+I0) \xrightarrow{9} a^*(a+I00) \xrightarrow{6} a^*(a+b00)$

$E \xrightarrow{*} a^*(a+b00)$ GILM

Leftmost Derivation → dari kiri dulu.

contoh:

① $L_1 = \{ \tau \in \{0,1\}^* \}$

Basis:

$G = (V = \{S\}, T = \{0,1\}, A, S)$

$L = \{ \varepsilon, 01, 101, \dots \}$

A: $S \to \varepsilon$

$S \to 0S1$

$S \to 1S0$

$S \to SS$

② A: <Sentence> → <noun-phrase><Predicate>

<noun-phrase> → <article> <noun>

<Predicate> → <verb>

<article> → The | a

<noun> → Cat | Man | Dog

<verb> → eats | sleeps | walks | runs.

Penurunan dari " the dog walks " ?

Jawab:

$G = (V = \{S, NP, P, A, N, V\}, T = \{9\}, A, S)$

$S \xrightarrow{1} \langle NP \rangle \langle P \rangle \xrightarrow{2} \langle A \rangle \langle N \rangle \langle P \rangle \xrightarrow{4}$ the $\langle N \rangle \langle P \rangle$

$\xrightarrow{5}$ the dog $\langle P \rangle \xrightarrow{3}$ the dog $\langle verb \rangle \xrightarrow{6}$

the dog walks // → benar.

Parse Tree → hasilnya berupa terminal.

Root : start symbol

Child : Body aturan CFG.

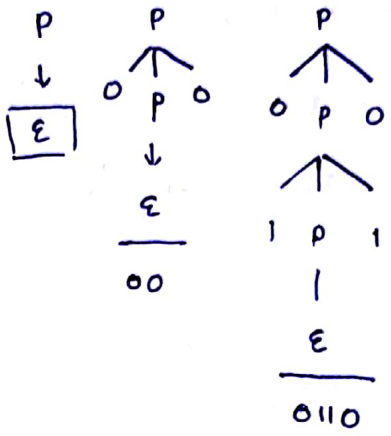
leaf : Hasil

contoh:

① Gpal: $P \to \varepsilon \mid 0 \mid 1$

$P \to 0P0$

$P \to 1P1$



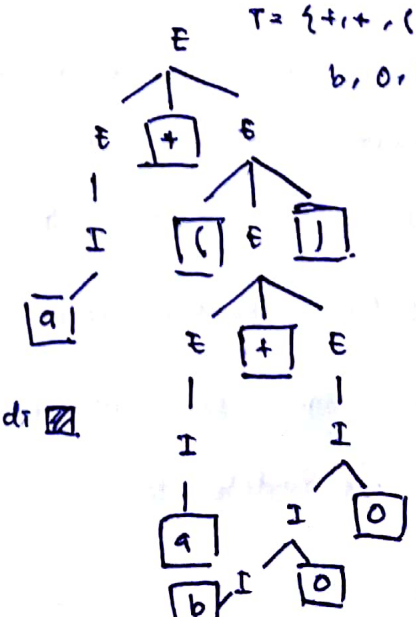
$\{ \varepsilon, 00, 0110, \dots \}$

② CFG:

1. $E \to I$
2. $E \to E+E$
3. $E \to E*E$
4. $E \to (E)$
5. $I \to a \mid b$
6. $I \to Ia \mid Ib$
7. $I \to I_0 \mid I_1$

misal: $a * (a + b00)$ ← string.

$T = \{+, *, (,), a, b, 0, 1\}$



leaf yg di ▨

③ diberikan CFG $G = (\{S\}, \{a, b, c\}, A, S)$

A : $S \to a$

$S \to SbS \mid ScS$

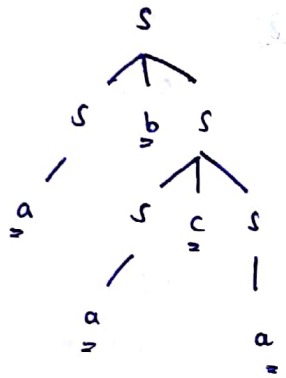
(a.) tentukan LMD dan string abaca, dan parse tree nya.

leftmost:

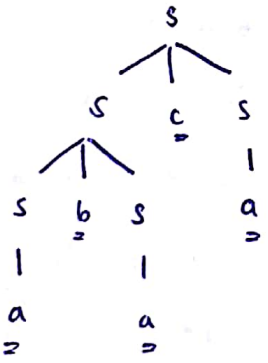
$S \xrightarrow{2} SbS \xrightarrow{1} abS \xrightarrow{3} abScS \xrightarrow{1}$

$abacS \xrightarrow{1} abaca \to S \overset{*}{\underset{lm}{\Rightarrow}} abaca.$

parse tree:



(b.)



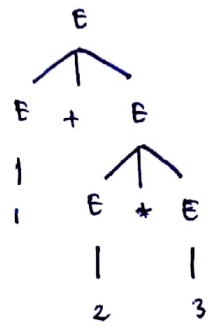
← memiliki sifat ambiguitas.

④ Buatlah grammar untuk ekspresi artimatika sederhana dari bilangan bulat 0 s.d 9. contoh string: $(1+2*3$, $1+2*(3))$. Buatlah parse tree dari kedua string tsb berdasarkan dari CFG yg sudah dibuat.

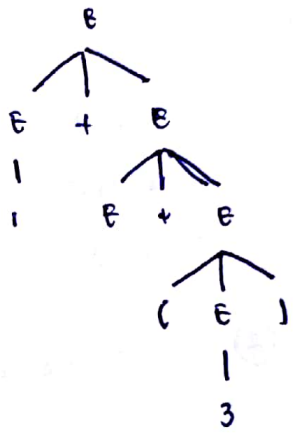
CFG:

1. $E \to 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$
2. $E \to (E)$
3. $E \to E + E$
4. $E \to E * E$

$E = 1 + 2 * 3$



$E = 1 + 2 * (3)$



leftmost :

$E \xrightarrow{3} E+E \xrightarrow{1} 1+E \xrightarrow{4} 1+E*E \xrightarrow{1}$

$1+2*E \xrightarrow{2} 1+2*(E) \xrightarrow{1} 1+2*(3)$

⑤ CFG :

$S \to AB \mid C$

$A \to aAB \mid ab$

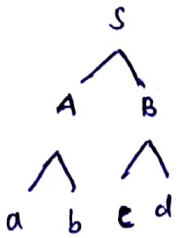
$B \to cBd \mid cd$

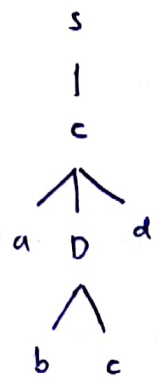
$C \to aCd \mid aDd$

$D \to bCd \mid bc$

$w = abcd$

$S \xrightarrow{1} AB \xrightarrow{2} abB \xrightarrow{3} abcd$





∴ ambigu.

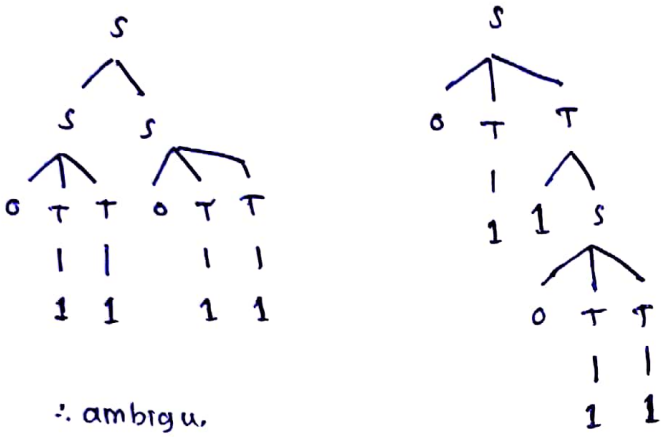
$S \xrightarrow{1} C \xrightarrow{4} aDd \xrightarrow{5} abcd$

⑥ CFG :

$S \rightarrow SS \mid 0TT \mid TOT \mid TT0$

$T \rightarrow TS \mid STS \mid S1 \mid 1.$

Jawab : $w = 0110$



∴ ambigu.

aplikasi CFG

⑦ If (a<20) { printf ("ilkom"); }

for (a=1 ; a<20; a=a+1)

{ printf ("ilkom"); }

Ekspresi : E

State : ○ IF - state

○ For - state

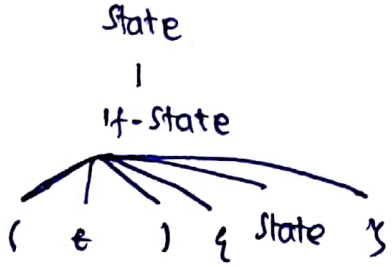
Jawab :

CFG :

State → For-state | If - state | ε

For-state → (E; E ; E) { State };

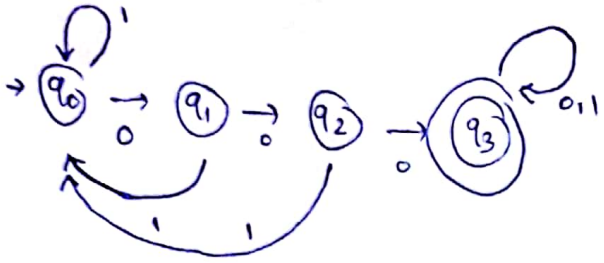
IF-state → (E) { state };



Bella Anggita Safitri . 6664700159.

1. $L = \{0,1\}^* 000 \{0,1\}^*$

Pada alphabet $\{0,1\}$ yg mengan dung 3 buah 0 secara berurutan



Tabel transisi :

Σ	0	1
→ q_0	q_1	q_0
q_1	q_2	q_0
q_2	q_3	q_0
q_3	q_3	q_3

2 Buatlah DFA yg menerima string dan para 0 dan para 1 yg mengandung substring 11. lakukan Pemrosesan string w = 01101 menggunakan DFA yg dihasilkan.

$A = (\{q_0, q_1, q_2\}, \{0,1\}, \delta, q_0, \{q_2\})$

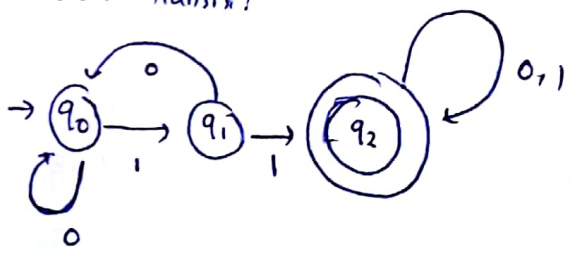
dimana δ:

$\delta(q_0,0) = q_0$ $\quad \delta(q_0,1) = q_1$

$\delta(q_1,1) = q_2$ $\quad \delta(q_1,0) = q_0$

$\delta(q_2,0) = q_2$ $\quad \delta(q_2,1) = q_2$

Diagram Transisi :



Tabel transisi :

Σ	0	1
→ q_0	q_0	q_1
q_1	q_0	q_2
q_2	q_2	q_2

Perluasan:

$w = 01101$

Basis: $\hat{\delta}(q_0, \varepsilon) = q_0$

Induksi:

$w = 0 \rightarrow x = \varepsilon, a = 0$

$\hat{\delta}(q_0, w) = \hat{\delta}(q_0, \varepsilon 0) = \delta(\hat{\delta}(q_0, \varepsilon), 0)$

$= \delta(q_0, 0)$

$= q_0$

$w = 01 \rightarrow x = \varepsilon 0, a = 1$

$\hat{\delta}(q_0, w) = \hat{\delta}(q_1, \varepsilon 01) = \delta(\hat{\delta}(q_0, \varepsilon 0), 1)$

$= \delta(q_0, 1)$

$= q_1$

$w = 011 \rightarrow x = \varepsilon 01, a = 1$

$\hat{\delta}(q_0, w) = \hat{\delta}(q_1, \varepsilon 011) = \delta(\hat{\delta}(q_0, \varepsilon 01), 1)$

$= \delta(q_1, 1)$

$= q_2$

$w = 0110 \rightarrow x = \varepsilon 011, a = 0$

$\hat{\delta}(q_0, w) = \hat{\delta}(q_1, \varepsilon 0110) = \delta(\hat{\delta}(q_0, \varepsilon 011), 0)$

$= \delta(q_2, 0)$

$= q_2$

$w = 01101 \rightarrow x = \varepsilon 0110, a = 1$

$\hat{\delta}(q_0, w) = \hat{\delta}(q_1, \varepsilon 01101) = \delta(\hat{\delta}(q_0, \varepsilon 0110), 1)$

$= \delta(q_2, 1)$

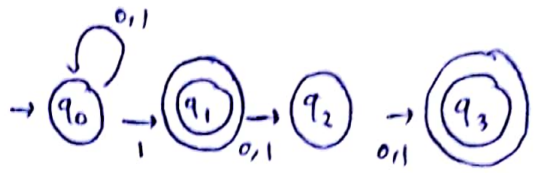
$= q_2 \rightarrow$ Final state

$\therefore \hat{\delta}(q_0, 01101)\ q_2 \in F.$

$\therefore \hat{\delta}(q_0, w = 01101) = q_2 \cap F \neq \emptyset$

Sehingga diterima.

(3.)



NFA $A = (Q, \Sigma, \delta, q_0, F\{q_1, q_3\})$

- Basis = $\hat{\delta}(q_0, \varepsilon) = q_0$ $w = 0100$

Induksi =

- $w = \varepsilon 0 \rightarrow x = \varepsilon, a = 0$

$\hat{\delta}(q_0, \varepsilon 0) = \cup\ \delta(p_i, 0),\ p_i \in \hat{\delta}(q_0, \varepsilon)$

$, p_i \in q_0$

$= \delta(q_0, 0)$

$= q_0$

- $w = \varepsilon 01, x = \varepsilon 0, a = 1$

$\hat{\delta}(q_0, \varepsilon 01) = \cup\ \delta(p_i, 1),\ p_i \in \hat{\delta}(q_0, \varepsilon 0)$

$p_i \in q_0$

$= \delta(q_0, 1)$

$= \{q_0, q_1\}$

- $w = \varepsilon 010, x = \varepsilon 01, a = 0$

$\hat{\delta}(q_0, \varepsilon 010) = \cup \delta(p_i, 0),\ p_i \in \hat{\delta}(q_0, \varepsilon 01)$

$, p_i \in \{q_0, q_1\}$

$= \delta(q_0, 0) \cup \delta(q_1, 0)$

$= q_0 \cup q_2$

$= \{q_0, q_2\}$

- $w = \varepsilon 0100, x = \varepsilon 010, a = 0$

$\hat{\delta}(q_0, \varepsilon 0100) = \cup\ \delta(p_i, 0),\ p_i \in \hat{\delta}(q_0, \varepsilon 010)$

$p_i \in \{q_0, q_2\}$

$= \delta(q_0, 0) \cup \delta(q_2, 0) = \{q_0, q_1\}$

$\therefore$ diterima.