MODUL 2:
Bahasa Regular dan Ekspresi Regular

DEFINISI BAHASA REGULAR

Bahasa Regular *L* dari alfabet Σ adalah bahasa yang dapat dihasilkan dari bahasa-bahasa paling sederhana dari Σ dengan melakukan operasi-operasi

- gabungan,
- konkatenasi, dan/atau
- Kleene-*.

Contoh:

$L = \{x \in \{0,1\} \mid x$ konkatenasi berulang 110 dan diakhiri oleh satu simbol apa saja (0 atau 1)$\}$.

$L = \{110\}^*\{0, 1\}$ atau

$L = \{\{1\}\{1\}\{0\}\}^*\{\{0\} \cup \{1\}\}$.

- konkatenasi $\{1\}\{1\}\{0\}$ menjadi $\{110\}$
- Kleen $\{110\}$ menjadi $\{110\}^*$
- penggabungan $\{0\} \cup \{1\}$ menjadi $\{0,1\}$
- konkatenasi hasil Kleen $\{110\}^*$ dengan $\{0,1\}$

Jadi merupakan bahasa regular.

Ekspresi Regular

- Ekspresi bisa dibuat lebih sederhana lagi dengan ekspresi yang mirip dengan ekspresi aritmatis yang disebut ekspresi regular.
- Dengan aturan:
  - ganti setiap { dengan (
  - ganti setiap } dengan)
  - ganti $\cup$ dengan +

Contoh:

Bahasa	**Ekspresi Regular**
$\{\Lambda\}$	Λ
$\{0\}$	0
$\{001\}$	001
$\{0, 1\}$	$0+1$
$\{0, 10\}$	$0+10$
$\{1, \Lambda\}\{001\}$	$(1+\Lambda)001$
$\{110\}^*\{0,1\}$	$(110)^*(0+1)$
$\{1\}^*\{10\}$	1^*10
$\{10, 111, 11010\}^*$	$(10+111+11010)^*$
$\{0, 10\}^*(\{11\}^*\cup(001, \Lambda\})$	$(0+10)^*((11)^* + 001 + \Lambda)$

Ekspresi regular memperlihatkan pola string yang paling tipikal dari bahasa ybs.

Contoh: $1^{*}10$ merupakan string-string berisikan akhiran 10 dan diawali sejumlah 1.

DEFINISI FORMAL EKSPRESI REGULAR

Kelas R dari bahasa regular pada Σ, dan ekspresi regularnya terdefinisi sebagai berikut:

1. Bila $\varnothing$ merupakan anggota dari R, ekspresi regularnya adalah $\varnothing$.
2. Bila $\{\Lambda\}$ adalah anggota dari R, ekspresi regularnya adalah Λ.
3. Untuk setiap $a \in \Sigma$, $\{a\}$ adalah anggota dari R, ekspresi regularnya adalah a.

4. Bila L_1 dan L_2 adalah anggota R, sementara r_1 dan r_2 adalah ekspresi regularnya maka
   (a) $L_1 \cup L_2$ anggota R, dan ekspresi regularnya adalah $(r_1 + r_2)$
   (b) L_1L_2 anggota R, maka ekspresi regularnya adalah (r_1r_2)
   (c) L_1^* anggota R, maka ekspresi regularnya adalah $(r_1)^*$.
5. Hanya dengan keempat pernyataan di atas bahasa regular dapat diperoleh.

Adalnya aturan pertama mengenai bahasa kosong $\emptyset$ diberikan untuk menjaga konsistensi dan kesederhanaan ekspresi. Sesuai dengan definisi himpunannya maka untuk setiap ekspresi regular r_1 dan r_2 yang mana tepat salah satu $\emptyset$, misalnya $r_2 = \emptyset$, maka $r_1+r_2 = r_2$ dan $r_1r_2 = \emptyset$. Kadang-kadang kita gunakan ekspresi regular r^2 menggantikan rr, serta r^+ menggantikan r^*r.

Hirarki antar Operator

Seperti halnya dalam ekspresi aritmetis, untuk menghilangkan ambiguitas saat menggunakan notasi +, *, dan konkatenasi, maka diperlukan suatu hirarki penulisan notasi.

- Hirarki tertinggi adalah pangkat (termasuk kleene-* dan kleene-$^+$), kemudian konkatenasi, dan terakhir +.
- Tanda kurung untuk mengelompokkan suatu subekspresi sebagai satu entitas ekspresi seperti halnya penulisan formula aritmetis.

Contoh:

$$1\,1^{*}(\,0 + 01^{*}\,)^{*}$$

Aljabar Ekspresi Regular

Keuntungan ekspresi regular lain: kita dapat menerapkan suatu aljabar ekspresi regular untuk penulisan ekpresi yang lebih ringkas/ sederhana.

Contoh:

$$1^*(1+\Lambda) = 1^*1 + 1^* = 1^+ + 1^*$$

$$1^*1^* = 1^*$$

$$(0^*1^*)^* = (0+1)^*$$

$$(0+1)^*01(0+1)^* + 1^*0^* = (0+1)^*$$

Untuk pers. Ketiga:

$0^*1^* = \Lambda + 0 + 1 + 0^+1^+ = (0 + 1) + (\Lambda + 0^+1^+)$

Maka

$(0^*1^*)^* = ((0 + 1) + (\Lambda + 0^+1^+))^* = (0 + 1)^* + (0 + 1)^*(\Lambda + 0^+1^+)^*$.

Karena $(0+1)^*$ merupakan himp. semesta dari string-string yang terbentuk dari {0, 1} maka terbukti $(0^*1^*)^* = (0+1)^*$.

Untuk pers. Keempat:
$(0+1)^*01(0+1)^*$ mrpk re dari bahasa yang selalu memiliki substring 01.
Komplemennya berisikan 0^* (hanya berisi simbol 0 saja), 1^* (hanya berisi simbol 1 saja), atau jika berisi 0 dan 1 simbol-simbol 0 akan berada setelah simbol-simbol 1, yaitu 1^*0^*.

Gabungan keduanya adalah himpunan seluruh string yang terbentuk dari {0,1} dengan ekspresi regular $(0+1)^*$.

Berikut ini sejumlah contoh penulisan bahasa regular dari deskripsi bahasa tsb.

Contoh 1: $L \subseteq \{0,1\}^*$ merupakan bahasa dari semua string yang panjangnya genap ($L = \{x \in \{0,1\}^* \mid |x| \in$ bil. genap$\}$, dalam hal ini $\Lambda \in L$ karena 0 adalah genap).

Setiap string yang panjangnya genap mrpk konkatenasi dari substring-substring yang panjangnya dua, jadi : $L = \{00, 01, 10, 11\}^*$. Ekspesi regularnya $(00+01+10+11)^*$ atau $(0(0+1)+1(0+1))^*$ atau $((0+1)^2)^*$

Contoh 2: Bila $L \subseteq \{0,1\}^*$ semua string yang berisi 1 dalam jumlah yang ganjil.

Setiap string memiliki minimal satu simbol 1. Maka selalu terdapat prefiks berbentuk 0^*10^* serta sejumlah (atau nol) substring berisi pasangan 1. Ekspresi regularnya: $0^*10^*(10^*10^*)^* = 0^*1(0^*10^*1)^*0^* = (0^*10^*1)^*0^*10^* = 0^*(10^*10^*)^*1(0^*10^*1)^*0^*$.

Contoh 3: $L \subseteq \{0,1\}^*$ string-string yang panjangnya 6 atau kurang atau $L = \{x \in \{0,1\}^* \mid |x| \leq 6\}$.

$L = \{\Lambda, 0, 1, 00, 01, 10, 11, 000, \ldots, 111110, 111111\}$ atau ekspresi regularnya adalah

$\Lambda+0+1+00+01+10+11+000+\ldots+111110+111111 = (\Lambda+0+1)^6$

Contoh 4: $L \subseteq \{0,1\}^*$ string-string yang diakhiri oleh 1 dan tidak memiliki substring 00. Pernyataan bahwa tidak ada substring 00 dan 0 tidak ada diakhir berimplikasi setelah setiap kemunculan 0 akan diikuti 1. Jadi ekspresi regular L adalah $(1+01)^+$. Kleene-$^+$ disini

Karena Λ tidak termasuk L.

Contoh 5: Bila *l* adalah ekspresi regular (`a+b+…+z+A+B+…+Z`) dan d adalah ekspresi regular (`0+1+…+9`) dan _ adalah karakter 'underscore'. maka setiap variabel dalam bahasa pemrograman C (setiap string dari huruf, angka dan dash tapi diawali oleh huruf atau dash) memikili eksrepsi regular:

$$(l+\_)(l+d+\_)^*$$

Contoh 6: Jika p adalah karakter titik dan s adalah ekspresi regular $(\Lambda + a + m)$ dengan a=karakter plus, m=karakter minus, maka bilangan real dalam bahasa pemrograman Pascal memiliki ekspresi regular:

$$sd^+ \, (pd^+ + pd^+Esd^+ + Esd^+)$$