MODUL 1:
PENGANTAR TEORI BAHASA

- **Pengantar Automata dan Bahasa**
- **Teori Pendukung**
- **Konsep Bahasa**

PENGANTAR AUTOMATA DAN BAHASA

Obyektif

membahas model-model komputasi sebagai mesin abstraks yang dapat didefinisikan secara matematis, mulai dari yang paling sederhana hingga yang paling powerful.

- formalisasi matematis disusun secara bertahap
- hubungannya dengan masalah dunia nyata

Model Masalah Keputusan

model-model tsb digambarkan sebagai mesin untuk menjawab masalah keputusan.

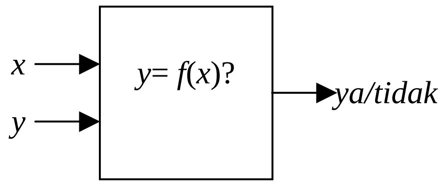
masukan string *x*, keluaran berupa: **ya** atau **tidak**
misalnya:

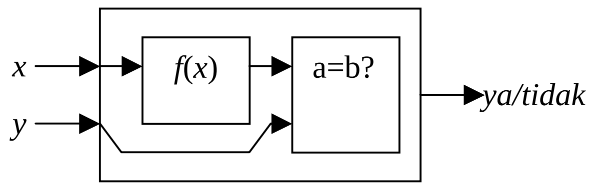
- apakah *x* adalah bilangan prima?
- apakah suatu string s anggota dari bahasa B?

Masalah Komputasi

Masalah komputasi lebih umum daripada masalah keputusan, namun masalah keputusan memerlukan komponen komputasi

Contoh: untuk (x, y), "apakah $y = f(x)$?" hanya dapat dijawab jika $f(x)$ dapat dikomputasi.





Model-model mesin yang akan dibahas

- Finite Automata (FA)
- Pushdown Automata (PDA), dan
- Mesin Turing (TM).

Bahasa-bahasa dari model-model tsb

- bahasa regular,
- bahasa context-free, dan
  bahasa rekursif & *recursively enumerable*.

Finite Automaton

- model memiliki sejumlah berhingga status
- setiap saat berada dl suatu status & berpindah ke status lain dengan aturan transisi yang terdefinisi sesuai simbol masukan
- digunakan untuk mengenal bahasa-bahasa regular
- keterbatasan: tidak memiliki memori/storage (status-status dirancang sebagai “memori”)

Pushdown Automaton

- Mrpk FA yang dilengkapi suatu memory berstruktur stack & kapasitas tak berhingga
- Dapat mengenali bahasa context-free yaitu suatu kelas bahasa yang lebih tinggi dari bahasa regular
- Dapat digunakan untuk melakukan parsing struktur suatu string.

Mesin Turing

- FA dengan sequential storage (tape) sbg memory yang dapat di baca-tulis.
- Dapat mengenali kelas bahasa yang lebih tinggi dari bahasa context-free
- Selain menjawab “ya” dan “tidak”, tetapi tape menjadi medium keluaran proses
- menjadi alat ukur untuk membandingkan kompleksitas masalah-masalah komputasi

Kelas-kelas Bahasa

Kelas-kelas bahasa tsb merepresentasikan kelas-kelas masalah dan model-model mesin abstraks tsb merepresentasikan kelas-kelas metoda pemecahan masalah.

Hirarki

- Kelas bhs reguler $\subset$ kelas bhs CF
- Kelas bhs CF $\subset$ dari kelas bhs Rekursif
- Kelas bhs Rekursif $\subset$ dari kelas bhs *Recursively Enumerable*.

TEORI PENDUKUNG

Notasi-notasi dan konsep-konsep penting yang perlu diingat kembali karena digunakan di dalam kuliah ini adalah sbb.

- Teori Himpunan
- Fungsi
- Logika
- Relasi

Mengenai Teori Himpunan

- Penulisan himpunan A = {00, 01, 10. 11} = $\{ab \mid a, b$ berharga 0 atau 1}
- $\in$: anggota himpunan
- $\notin$: bukan anggota himpunan
- U: himpunan universal
- $\emptyset$: himpunan kosong
- Himpunan komplemen $A' = \{x \in U \mid x \notin A\}$
- Diagram Venn
- Opr. gabungan: $A \cup B$

- Opr. irisan: $A \cap B$
- Opr. perbedaan: $A - B$
- Opr. perbedaan simetris: $A \oplus B$
- "A dan B disjoin": jika $A \cap B = \emptyset$,
- himpunan bagian: "A $\subseteq$ B"
- himpunan bagian: "A $\subset$ B"
- power set (2^A)
- pasangan terurut (a, b)
- Cartesian product $(A \times B)$
- partisi himpunan A menjadi $A_1, A_2, \ldots A_n$

- Sifat-sifat: komutatif, asosiatif, distributif, idempoten, absorptif
- hukum De Morgan: $A \cup B = (A' \cap B')'$

Mengenai Fungsi (f: $A \rightarrow B$):

- fungsi parsial
- fungsi total
- fungsi injeksi (satu-ke-satu)
- fungsi surjeksi (onto)
- fungsi bijeksi

- Komposisi fungsi-fungsi ($g \circ f$)
- invers dari fungsi ($x = f^{-1}(y)$)
- Operasi uner ($u(x)$)
- operasi biner ($f(x, y)$)

Mengenai Relasi (*a R b*)

- Sifat-sifat relasi: refleksif, simetri, transitivitas
- relasi ekivalen
- Kelas ekivalen berisikan *a* ($[a]_R$)

KONSEP MENGENAI BAHASA

Simbol

Elemen unik terkecil dari bahasa. Dari suatu bahasa terdapat sejumlah berhingga simbol yang digunakan.

Alfabet Simbol

Himpunan berhingga simbol suatu bahasa sebagai alfabet simbol (atau disingkat alfabet) dari bahasa tersebut.

String

Simbol-simbol menyusun string/deretan simbol sebagai suatu entitas (atau disingkat string). Dalam suatu string bisa ada simbol yang muncul lebih satu kali dan ada simbol yang tidak digunakan.

Bahasa

- Bahasa merupakan himpunan dari sejumlah string-string dari alfabet bahasa tersebut.
- Bahasa merupakan subset dari himpunan seluruh kemungkinan string yang dapat dibentuk dari alfabet bahasa tersebut.
- Himpunan bisa berhingga atau tidak berhingga.

Grammar

Susunan simbol-simbol dalam string-string suatu bahasa mengikuti aturan-aturan grammar (pembentukan) tertentu. Kompleksitas aturan-aturan pembentukan ini yang akan membedakan kelas-kelas bahasa yang akan dibahas nanti.

Dalam Konteks Bahasa Sehari-hari

- Secara gramatikal bahasa manusia mrpk himpunan **kalimat** di mana setiap kalimat tersusun atas **kata-kata** dengan aturan gramatikal yang tertentu. Alfabet adalah kata-kata (kosakata).
- Setiap kalimat bisa dipandang tersusun **huruf-huruf** dengan alfabet dalam arti yang sebenarnya di tambah sejumlah simbol (spasi, dlsb).

- Bisa juga dalam representasi biner ASCII alfabetnya = {0, 1} (kalimat = string karakter ASCII dan setiap simbol ASCII sebagai 8 bit biner).

Dalam Konteks Bahasa Pemrograman

- Sebuah program lengkap adalah suatu string.
- Keyword-keyword, nama-nama variabel, nama-nama fungsi, tanda-tanda operasi dan notasi adalah simbol-simbol di dalam bahasa ini.

Notasi Alfabet

- Alfabet kita tulis dengan lambang Σ
- Jika lebih dari satu alfabet kita nomori sebagai Σ_1, Σ_2, Σ_3, …
- Untuk penyederhanaan pembahasan alfabet hanya berisi dua bahkan mungkin satu simbol saja. Namun secara umum berlaku untuk alfabet yang lebih besar.

Contoh

Bahasa-bahasa yang terbentuk dari $\Sigma = \{a, b\}$:

$\{\Lambda, a, aa, ab\}$

$\{x \in \{a, b\}^* \mid |x| \leq 8\}$

$\{x \in \{a, b\}^* \mid |x| \in$ bilangan ganjil$\}$

$\{x \in \{a, b\}^* \mid n_a(x) = n_b(x)\}$

Notasi Variabel Simbol/string

- variabel string tertentu sering direpresentasikan sebagai *r*, *s*, *t*, .., *x*, *y*, atau *z*. (huruf-huruf kecil di urutan belakang dalam abjad latin)
- variabel simbol sering direpresentasikan sebagai *a*, *b*, *c*,… yaitu huruf-huruf kecil pada urutan terdepan di dalam abjad.

Notasi variabel himpunan string

Variabel untuk himpunan string dinotasikan dengan huruf kapital pada urutan awal abjad.

Contoh: L = {001, 1110, 01, 000001}.

Panjang String

Suatu string tersusun atas sejumlah n simbol, $n \geq 0$. Panjang string x ditulis $|x|$

Contoh: $|aabba| = 5$

String Kosong

Jika $|x| = 0$ maka string x tersebut disebut string kosong**.** Sebagai konvensi untuk kuliah ini string kosong dituliskan dengan notasi Λ.

Note:

- Buku teks lain menggunakan notasi ε
- Himpunan yang berisi hanya string kosong bukanlah himpunan kosong ($\{\Lambda\} \neq \emptyset$, tetapi $\{\} = \emptyset$)

Contoh

Dari $\Sigma = \{a, b\}$, beberapa string yang bisa dibentuk dari Σ adalah Λ, *a*, *abaa*, dan *aabbaa*.

Note: *a* dalam hal ini bisa berarti simbol anggota $\Sigma = \{a, b\}$, atau string dengan panjang 1. Serta,

$|aaa| = |aba| = 3$

$|a| = 1$

$|\Lambda| = 0$

Operasi konkatenasi pada string-string

String x dan y apabila dikonkatenasi akan menjadi string xy.

Contoh: Jika $x = aaba$ dan $y = bba$ maka $xy = aababba$. Jika $x = aaba$ dan $y = \Lambda$, maka $xy = x$.

Operasi konkatenasi berulang pada string

Konkatenasi dapat dilakukan sekian kali pada string yang sama. Penulisannya dapat menggunakan notasi pangkat. (**Note**: notasi pangkat disini berbeda arti dari notasi pangkat dalam operasi aritmetika!)

Contoh: Untuk x = *abb*, x^4 = *abbabbabbabb*.

Secara umum konkatenasi berulang adalah x^n dengan $n \geq 0$, yang mana jika $n = 0$, $x^n = \Lambda$.

Operasi konkatenasi pada himpunan string

Jika L_1 dan L_2 himp.-himp. string maka L_1L_2 adalah $\{xy \mid x \in L_1 \text{ dan } y \in L_2\}$

Contoh: Jika $L_1 = \{aa, bab\}$ dan $L_2 = \{bb, a\}$, maka $L_1L_2 = \{aabb, aaa, babbb, baba\}$.

Note: bahwa $L_1L_2 \neq L_2L_1$. Untuk $L_1 = \{\Lambda\}$ maka $L_1L_2 = L_2L_1 = L_2$.

Operasi konkatenasi berulang pada alfabet

konkatenasi dapat dilakukan sekian kali pada alfabet Σ. Juga, penulisannya dapat menggunakan notasi pangkat.

Contoh: untuk $\Sigma = \{a, b\}$, $\Sigma^3 = \Sigma\Sigma\Sigma = \{aaa, aab, aba, abb, baa, bab, bba, bbb\}$.

Secara umum konkatenasi adalah Σ^n dengan $n \geq 0$, yang mana jika $\Sigma^0 = \{\Lambda\}$.

Konkatenasi berulang pada himpunan string

Konkatenasi dapat dilakukan sekian kali pada himpunan string yang sama. Juga, penulisannya dapat menggunakan notasi pangkat.

Contoh: $L = \{aa, b\}$, $L^3 = LLL = \{aaaaaa, aaaab, aabaa, aabb, baaaa, baab, bbaa, bbb\}$.

Secara umum ditulis L^n dengan $n \geq 0$, yang mana jika $L^0 = \{\Lambda\}$.

Jadi, untuk $a \in \Sigma$, $x \in L$, dan $L \subseteq \Sigma^*$, maka:
$a^k = aa\ldots a$, $x^k = xx\ldots x$, $L^k = LL\ldots L$
$\Sigma^k = \Sigma\Sigma\ldots\Sigma = \{x \in \Sigma^* \mid |x| = k\}$
$a^0 = \Lambda$, $x^0 = \Lambda$, $L^0 = \{\Lambda\}$, $\Sigma^0 = \{\Lambda\}$

Contoh:
$\Sigma = \{0, 1\}$, maka $\Sigma\Sigma = \{00, 01, 10, 11\}$
$L = \{0, 111\}$, maka $LLL = \{000, 00111, 01110, 0111111, 11100, 1110111, 1111110, 111111\}$

Operasi Kleene-* pada Alfabet

Untuk alfabet Σ maka himpunan seluruh string yang mungkin terbentuk dari Σ adalah Σ^* (dibaca: Kleene-* dari Σ).
Jadi, setiap bahasa L yang menggunakan alfabet Σ adalah subset dari Σ^*.

Contoh: Jika $\Sigma = \{a, b\}$, maka $\Sigma^* = \{\Lambda, a, b, aa, ab, ba, bb, aaa, aab, aba, baa, abb, bab, bba, bbb, \ldots\}$.

Operasi Kleene-* pada himpunan string

Bila L^k adalah himpunan semua string yang terbentuk dengan konkatenasi k elemen dari L maka L^* adalah himpunan semua string dari berbagai konkatenasi yang bisa dilakukan pada setiap elemen dari L. Ditulis

$$L^* = \bigcup_{i=0}^{\infty} L^i$$

Operasi ini disebut Kleene-* (mengambil nama S.C. Kleene). L^* termasuk Λ karena $L^0 = \{\Lambda\}$.

Operasi Kleene-$^+$ pada himpunan string

Untuk meniadakan L^0 maka digunakan notasi L^+ yang didefinisikan sebagai operasi Kleene-$^+$ sbb:

$$L^+ = \bigcup_{i=1}^{\infty} L^i$$

Dengan demikian $L^+ = LL^* = L^*L$.

Bukti: menurut definisinya $L^+ = L^1 \cup L^2 \cup L^3 \cup L^4 \cup \ldots = LL^0 \cup LL^1 \cup LL^2 \cup LL^3 \cup \ldots = L(L^0 \cup L^1 \cup L^2 \cup L^3 \cup \ldots) = LL^*$

Operasi-operasi Himpunan Umum

Sebagai himpunan-himpunan string, bahasa baru dapat dibentuk sebagai hasil operasi himpunan: gabungan, irisan, perbedaan, kompelemen.

Operasi-operasi ini dpt dilakukan baik pada bahasa-bahasa dari alfabet yang sama maupun yang berbeda.

Contoh: $L_1 \subseteq \Sigma_1^*$ dan $L_2 \subseteq \Sigma_2^*$ maka $(L_1 \cup L_2)$ adalah subset dari $(\Sigma_1 \cup \Sigma_2)^*$.

Operasi-operasi string lain

- x substring dari string y jika terdapat string lain w dan z sehingga $y = wxz$.
- x prefiks dari string y jika terdapat string lain z sehingga $y = xz$.
- x sufiks dari string y jika terdapat string lain w sehingga $y = wx$.

Bahasa Paling Sederhana dari Σ

Himpunan string subset dari Σ^* yang berisi string tunggal dengan panjang 0 atau 1 disebut bahasa paling sederhana dari Σ.

Contoh: $\Sigma = \{a, b, c\}$, maka bahasa-bahasa paling sederhana dari Σ adalah $\{\Lambda\}$, $\{a\}$, $\{b\}$, dan $\{c\}$.