

Rangkuman Materi Bahasa Formal dan Otomata

Halo! Berikut rangkuman materi Bahasa Formal dan Otomata untuk persiapan ujian tengah semester Anda:

1. Pengenalan Bahasa Formal

Konsep Dasar:

- **Alfabet (Σ):** Kumpulan simbol dasar (misal: $\Sigma = \{0, 1\}$)
- **String:** Rangkaian simbol dari alfabet (misal: "010", "11")
- **Bahasa:** Himpunan string yang dibentuk dari alfabet
- **String kosong (ϵ):** String tanpa simbol

2. Hierarki Chomsky

Membagi bahasa formal menjadi 4 tipe:

- **Tipe 0:** Bahasa Rekursif yang Dapat Dihitung (Turing Machine)
- **Tipe 1:** Bahasa Context-Sensitive (Linear Bounded Automata)
- **Tipe 2:** Bahasa Context-Free (Pushdown Automata)
- **Tipe 3:** Bahasa Regular (Finite Automata)

3. Regular Language dan Regular Expression

Regular Expression (Regex):

- Cara untuk mendeskripsikan regular language
- Operator dasar:
 - Concatenation (ab)
 - Alternation/Union ($a|b$)
 - Kleene Star (a^*)
 - Plus (a^+): satu atau lebih kemunculan
 - Optional ($a^?$): nol atau satu kemunculan

Contoh:

- $a(b|c)^*$ menggambarkan string yang dimulai dengan 'a' diikuti 0 atau lebih 'b' atau 'c'

4. Finite Automata (FA)

Deterministic Finite Automata (DFA)

- **Komponen:** $(Q, \Sigma, \delta, q_0, F)$
 - Q : Himpunan state
 - Σ : Alfabet input
 - δ : Fungsi transisi ($Q \times \Sigma \rightarrow Q$)
 - q_0 : State awal
 - F : Himpunan state penerima
- **Ciri khas:** Untuk setiap state dan input, tepat satu transisi yang mungkin

Nondeterministic Finite Automata (NFA)

- Menggunakan komponen yang sama seperti DFA
- **Perbedaan:** Satu state dapat memiliki beberapa transisi untuk input yang sama
- Dapat memiliki transisi- ϵ (berpindah state tanpa membaca input)

5. Konversi antar Model

NFA ke DFA

- Menggunakan algoritma subset construction
- State DFA adalah kombinasi (subset) dari state-state NFA
- Melacak semua kemungkinan state NFA yang aktif pada suatu saat

Regex ke FA

- Menggunakan metode Thompson's Construction
- Membangun NFA untuk setiap operator regex
- Menggabungkannya sesuai struktur regex

FA ke Regex

- Menggunakan metode state elimination
- Mengeliminasi state satu per satu
- Menyesuaikan transisi dengan ekspresi regex

6. Pumping Lemma untuk Regular Language

- Alat untuk membuktikan bahwa suatu bahasa BUKAN regular
- Prinsip: Jika L adalah regular, ada konstanta p sehingga setiap string $s \in L$ dengan $|s| \geq p$ dapat dipecah menjadi $s = xyz$ dengan:
 1. $|y| > 0$
 2. $|xy| \leq p$
 3. $xy^iz \in L$ untuk semua $i \geq 0$

Contoh penerapan: Membuktikan bahasa $\{a^n b^n \mid n \geq 0\}$ bukan bahasa regular

7. Context-Free Grammar (CFG)

- **Komponen:** (V, Σ, R, S)
 - V : Variabel non-terminal
 - Σ : Simbol terminal
 - R : Aturan produksi
 - S : Simbol awal
- **Aturan produksi:** $A \rightarrow \alpha$, dimana $A \in V$ dan $\alpha \in (V \cup \Sigma)^*$

Contoh CFG: $S \rightarrow aSb \mid \epsilon$ menghasilkan bahasa $\{a^n b^n \mid n \geq 0\}$

Bentuk Normal

- **Bentuk Normal Chomsky (CNF):** Semua produksi berbentuk $A \rightarrow BC$ atau $A \rightarrow a$
- **Bentuk Normal Greibach (GNF):** Semua produksi berbentuk $A \rightarrow a\alpha$

8. Pushdown Automata (PDA)

- Otomata dengan tambahan memory stack
- **Komponen:** $(Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$
 - Q : Himpunan state
 - Σ : Alfabet input
 - Γ : Alfabet stack
 - δ : Fungsi transisi
 - q_0 : State awal
 - Z_0 : Simbol awal stack
 - F : Himpunan state penerima
- Dapat mengenali bahasa context-free
- **Contoh:** PDA untuk $\{a^n b^n \mid n \geq 0\}$:
 - Push ke stack untuk setiap 'a'
 - Pop dari stack untuk setiap 'b'

- Terima jika stack kosong di akhir

Tips untuk Ujian

1. Pahami definisi formal dan notasi matematika
2. Latih menggambar dan memahami DFA, NFA, dan PDA
3. Berlatihlah mengonversi antara berbagai representasi bahasa
4. Fokus pada pembuktian menggunakan Pumping Lemma

Semoga rangkuman ini membantu persiapan UTS Anda! Selamat belajar dan sukses untuk ujiannya.