Theory of Computation

Short notes for B.Tech CSE / IT · MCA · GATE CS

Uncodemy Handwritten Notes · free for students

1. Basics

- **Alphabet (Σ)** = finite set of symbols, e.g. {0, 1}.
- **String** = finite sequence of symbols. **Empty string** = ε (length 0).
- **Language** = a set of strings over Σ. Σ^* = all strings including ε. $\Sigma+ = \Sigma^* - \{\varepsilon\}$.
- Number of strings of length n over Σ with |Σ| = k is k^n.

➤ Chomsky hierarchy

Type	Grammar	Language	Machine
3	Regular	Regular	Finite automaton (DFA / NFA)
2	Context-free	CFL	Pushdown automaton (PDA)
1	Context-sensitive	CSL	Linear bounded automaton (LBA)
0	Unrestricted	Recursively enumerable	Turing machine

- Regular ⊂ CFL ⊂ CSL ⊂ Recursive ⊂ RE.

2. Finite Automata

- **DFA** = (Q, Σ, δ, q0, F). δ: Q x Σ → Q. Exactly one move for each state and symbol.
- **NFA**: δ: Q x Σ → set of states. Can have 0, 1 or many moves, and ε-moves (ε-NFA).
- DFA, NFA and ε-NFA accept the same class (regular languages).
- NFA to DFA by **subset construction**: an n-state NFA can need up to 2^n DFA states.

Example DFA: strings over {0, 1} ending with "01"

```
        0          1
-> q0 ---> q1 ---> (q2)
   q0 on 1 -> q0      q1 on 0 -> q1
   q2 on 0 -> q1      q2 on 1 -> q0
```

- q0 = start, q2 = final. 3 states, which is the minimum.

> **TIP** Minimum DFA for "strings whose length is divisible by n" has n states. For "binary number divisible by n" the remainder DFA has n states (0 to n − 1); it is minimal when n is odd (n = 4 needs only 3).

> **EXAMPLE** Strings over {a, b} containing substring "aba": minimum DFA has 4 states (3 progress states + 1 accept state).

DFA minimization

- Find equivalent states using partitions: start with {final} and {non-final}, split groups that go to different groups on some symbol, repeat.
- Remove unreachable states first.

Mealy and Moore

- Finite automata with output. Moore: output on state. Mealy: output on transition. For the same input, a Moore machine gives an output string one symbol longer.

3. Regular Expressions and Regular Languages

RE	Language
a^*	ε, a, aa, aaa, ...
a+	a, aa, ... (same as aa^*)
$(a + b)^*$	All strings over {a, b}
$(a + b)^*abb$	Ends with abb
$(0 + 1)^*1(0 + 1)$	Second last symbol is 1
$(aa)^*$	Even number of a's

Useful identities

- $\varepsilon + RR^* = R^*$, $(R^*)^* = R^*$, $(a + b)^* = (a^*b^*)^*$, $R(SR)^* = (RS)^*R$.

- **Arden's theorem**: if $R = Q + RP$ and P has no ε, then $R = QP^*$.

➤ Closure properties of regular languages

- Closed under: union, intersection, complement, concatenation, Kleene star, reversal, difference, homomorphism.

➤ Pumping lemma (regular)

- If L is regular, there is p such that any string w in L with $|w| >= p$ splits as $w = xyz$ with $|xy| <= p$, $|y| >= 1$, and $xy^i z$ in L for all $i >= 0$.
- Used only to prove a language is NOT regular.

EXAMPLE $L = \{a^n b^n : n >= 0\}$ is not regular. Take $w = a^p b^p$. y has only a's. Pump $i = 2$: more a's than b's, not in L. Contradiction.

Regular	Not regular
$a^n b^m$ ($n, m >= 0$)	$a^n b^n$
Even number of a's	Equal number of a's and b's
a^n with $n \bmod 3 = 0$	a^p, p prime
Strings ending in ab	ww (w over $\{a, b\}$)

4. Context-Free Grammars

- **CFG** = (V, T, P, S). Productions of the form $A \to \alpha$ (one variable on the left).
- **Derivation**: leftmost / rightmost. **Parse tree** shows the derivation.
- **Ambiguous grammar**: some string has two or more parse trees (or two leftmost derivations).
- Example: $E \to E + E \mid E * E \mid id$ is ambiguous for $id + id * id$.
- Some CFLs are **inherently ambiguous**: every grammar for them is ambiguous.

EXAMPLE Grammar for $a^n b^n$ ($n >= 0$): $S \to aSb \mid \varepsilon$. Palindromes over $\{a, b\}$: $S \to aSa \mid bSb \mid a \mid b \mid \varepsilon$.

➤ Simplification and normal forms

- Remove ε-productions, unit productions, then useless symbols (in this order).
- **CNF (Chomsky Normal Form)**: $A \to BC$ or $A \to a$.
- **GNF (Greibach Normal Form)**: $A \to a\alpha$ (terminal followed by variables).

FORMULA In CNF, a string of length n has a derivation of exactly 2n − 1 steps.

5. Pushdown Automata

- **PDA** = finite automaton + a stack. Accepts by final state or by empty stack (both give the same class for NPDA).
- **NPDA** accepts all CFLs. **DPDA** accepts only deterministic CFLs (a smaller class).
- Example: a^n b^n: push A for each a, pop A for each b, accept when the stack is back to the bottom symbol.
- ww^R (even palindromes) needs an NPDA (must guess the middle). wcw^R works with a DPDA.

➤ Closure properties of CFLs

Operation	CFL	DCFL
Union	Yes	No
Intersection	No	No
Intersection with regular	Yes	Yes
Complement	No	Yes
Concatenation, Star	Yes	No
Reversal, Homomorphism	Yes	No

EXAM POINT a^n b^n c^n is not context-free. Proof: CFL pumping lemma (w = uvxyz, |vxy| <= p, |vy| >= 1).

CFL	Not CFL
a^n b^n	a^n b^n c^n
a^n b^m c^m	a^n b^n c^n d^n
ww^R	ww
a^i b^j with i <= j	a^(n^2)

6. Turing Machines

- **TM** = finite control + infinite tape + read/write head that moves left or right.
- δ: Q x Γ → Q x Γ x {L, R}.

- Variants (multi-tape, non-deterministic, multi-head) have the same power as a basic TM.
- **Church–Turing thesis**: anything computable by an algorithm can be computed by a TM.

Term	Meaning
Recursive (decidable)	A TM always halts, says yes or no
Recursively enumerable	A TM halts and says yes for strings in L; may loop for others
Universal TM	A TM that can simulate any other TM

- Recursive languages are closed under complement. RE languages are not.
- If L and its complement are both RE, then L is recursive.

7. Decidability

Problem	Regular	CFL	CSL	Recursive	RE
Membership	D	D	D	D	UD
Emptiness	D	D	UD	UD	UD
Finiteness	D	D	UD	UD	UD
Equivalence	D	UD	UD	UD	UD
Is $L = \Sigma^*$	D	UD	UD	UD	UD

- D = decidable, UD = undecidable.

➤ Famous undecidable problems

- **Halting problem**: will TM M halt on input w? Undecidable (but RE).
- **Post Correspondence Problem (PCP)**: undecidable.
- Is a CFG ambiguous? Undecidable. Is the intersection of two CFLs empty? Undecidable.

REMEMBER Rice's theorem: every non-trivial property of the language of a TM is undecidable.

8. Most Asked Questions (practice)

1. Design a DFA for given conditions (ends with, contains, divisible by).

2. Convert an NFA to a DFA using subset construction.
3. Minimise a DFA using the partition method.
4. Write regular expressions for given languages. Use Arden's theorem.
5. Prove a language is not regular using the pumping lemma.
6. Show a grammar is ambiguous. Convert a CFG to CNF.
7. Design a PDA for a^n b^n and for ww^R.
8. Closure properties of regular and context-free languages.
9. Design a Turing machine for a^n b^n c^n or for adding unary numbers.
10. Explain the halting problem and Rice's theorem.

TIP GATE TOC is mostly "identify the class of this language", minimal DFA states, regular expression equivalence and decidability tables. Learn the closure and decidability tables by heart.