Comprehensive TAFL Revision

From finite automata to computational complexity

Definitions • Constructions • Proofs • Comparative analysis

Revision roadmap

Regular languages

DFA, NFA, ε-NFA, minimization, transducers, regular expressions and closure.

Context-free languages

CFGs, derivations, ambiguity, normal forms, PDA and pumping arguments.

Computability and complexity

Turing machines, hierarchy, decidability, reductions, P and NP.

Language classes and machines

Class	Grammar / representation	Machine	Example
Regular	Right-linear / regular expression	FA	(ab)*
Context-free	$A \to \alpha$	NPDA	$\{a^n b^n : n \geq 0\}$
Context-sensitive	Noncontracting*	Nondeterministic LBA	$\{a^n b^n c^n : n \geq 1\}$
Turing-recognizable	Unrestricted	TM	A_{TM}

* Protected start-symbol ε exception. Each class strictly contains the preceding class.

DFA, NFA and ε-NFA

DFA

$\delta : Q \times \Sigma \to Q$. Exactly one next state for every state and input symbol.

NFA / ε-NFA

δ returns a set of states; ε-NFA also allows moves without consuming input. Acceptance requires at least one accepting path.

Subset construction

Start with ε-closure($\{q_0\}$); for subset T and symbol a, compute ε-closure(move(T,a)). Accept if T intersects F.

Worked example: NFA to DFA

Subset	Input 0	Input 1	Accept?
→ {s}	{s,t}	{s}	No
{s,t}	{s,t}	{s,f}	No
{s,f}	{s,t}	{s}	Yes

NFA: s loops on 0 and 1; s —0→ t; t —1→ f. Final state f has no outgoing moves. Language: strings ending in 01.

DFA minimization and equivalence

Preparation

Remove unreachable states. Complete the DFA with a sink if needed.

Table filling

Mark final/nonfinal pairs. Repeatedly mark pairs whose same-symbol successors form a marked pair.

Merge and verify

Unmarked pairs are equivalent; merge them into classes. For equivalence of two DFAs, test emptiness of their symmetric difference.

Moore vs. Mealy machines

Feature	Moore	Mealy
Output function	$\lambda : Q \to \Gamma$	$\lambda : Q \times \Sigma \to \Gamma$
Output label	On states	On transitions
Output length for n inputs	n + 1 (initial output included)	n
Conversion	To Mealy: destination output	To Moore: split by incoming output

Example: parity tracker toggles on 1 and stays on 0. Moore output is 1 in even state; Mealy emits the destination parity.

Regular expressions and conversions

Operators

Union (|), concatenation, Kleene star (*). Precedence: star, concatenation, union.

RE → FA

Thompson construction: combine ε-NFAs for union, concatenation and star; then apply subset construction if a DFA is needed.

FA → RE

State elimination: $R_{ij} \leftarrow R_{ij} \mid R_{ik}(R_{kk})^*R_{kj}$. Arden: $X = AX \mid B$ gives $X = A^*B$ uniquely if $\varepsilon \notin A$.

Worked proof: nonregularity

Assume regularity

$L = \{a^n b^n : n \geq 0\}$. Let p be the pumping length and choose $w = a^p b^p$.

Consider every valid split

$w = xyz$, $|xy| \leq p$ and $|y| > 0$. Thus $y = a^k$ for some $1 \leq k \leq p$.

Pump down

$xy^0z = a^{p-k}b^p \notin L$. Contradiction; L is not regular. A pumping condition is necessary, not sufficient, for regularity.

CFGs, derivations and ambiguity

Grammar

G = (V, Σ, P, S); productions A → α. Example: S → aSb | ε generates equal blocks.

Derivations

Leftmost expands the leftmost variable; rightmost expands the rightmost. S ⇒ aSb ⇒ aaSbb ⇒ aabb.

Ambiguity

A grammar is ambiguous if some word has two distinct parse trees. E → E+E | E*E | id gives two parses for id+id*id.

Grammar simplification and normal forms

Simplify

Introduce a fresh start if needed; remove ε-productions (preserving ε when required), unit productions, then non-generating and unreachable symbols.

Chomsky normal form

$A \to BC$ or $A \to a$; protected start $\to \varepsilon$ if needed. Replace terminals in mixed RHSs and binarize long RHSs.

Greibach normal form

$A \to aA_1 \ldots A_k$, $k \geq 0$; protected ε exception. Every derivation of a nonempty word of length n takes n steps.

PDA and CFG equivalence

PDA model

Finite control + stack; transition reads an input symbol or ε and replaces the top stack symbol.

CFG → NPDA

Push S; nondeterministically replace a variable by a production RHS, then match terminal stack symbols against input. Accept by empty stack after consuming input.

NPDA → CFG

Use variables [pAq] for strings that take state p with A on top to state q after removing A, leaving the lower stack unchanged.

Worked example: PDA stack trace

Remaining input	Stack (top at left)	Action
aabb	Z	Start
abb	AZ	Push A
bb	AAZ	Push A
b	AZ	Pop A
ε	Z	Pop A
ε	ε	Pop Z and accept

Language $a^n b^n$: push during a-phase; pop during b-phase; reject any a after b.

Closure and decision properties

Property	Regular	Context-free
Union / concatenation / star	Yes	Yes
Reversal / homomorphism	Yes	Yes
Intersection / complement	Yes	No (in general)
Intersection with regular	Yes	Yes
Membership / emptiness	Decidable	Decidable
Equivalence / inclusion	Decidable	Undecidable

Worked proof: non-context-freeness

Choose witness

Assume $L = \{a^n b^n c^n : n \geq 0\}$ is CFL. For pumping length p, take $w = a^p b^p c^p$.

Analyze all splits

$w = uvxyz$; $|vxy| \leq p$, $|vy| > 0$. The window touches at most two adjacent blocks, so at least one block is untouched.

Pump down

uv^0xy^0z reduces at least one count but leaves an untouched count p. Equal counts fail; contradiction.

Turing machines and computation

Model

A TM reads/writes a tape and moves its head. $\delta(q,a) = (r,b,D)$, where D is L or R (some models allow stay).

Recognizer vs. decider

A recognizer accepts members and may loop on nonmembers. A decider halts on every input.

Equivalent variants

Multi-tape and nondeterministic TMs recognize the same language class as standard TMs; simulation may change efficiency.

Decidability and reductions

Undecidable problems

$A_{TM} = \{\langle M,w\rangle : M \text{ accepts } w\}$; $HALT_{TM} = \{\langle M,w\rangle : M \text{ halts on } w\}$. Both are recognizable but undecidable.

Mapping reduction

$A \leq_m B$ means a total computable f with $x \in A$ iff $f(x) \in B$. If A is undecidable and $A \leq_m B$, then B is undecidable.

Rice and PCP

Every nontrivial semantic property of TM-recognized languages is undecidable. PCP asks for a nonempty sequence of tiles with matching concatenations.

Worked reduction: acceptance to halting

Source and target

Reduce A_TM to HALT_TM. Given ⟨M,w⟩, construct N that ignores its input and simulates M on w.

Construction

If M accepts w, N halts. If M rejects w, N loops; if M loops, N also loops.

Equivalence

⟨M,w⟩ ∈ A_TM iff ⟨N,ε⟩ ∈ HALT_TM. A halting decider would decide A_TM, a contradiction.

Computational complexity recap

Class	Definition	Example
P	Decision problems solvable in polynomial time	Reachability
NP	Polynomial-size certificates verifiable in polynomial time	SAT
NP-hard	Every NP problem reduces in polynomial time	TSP optimization*
NP-complete	In NP and NP-hard	3-SAT, CLIQUE

P ⊆ NP; whether P = NP remains open. *Optimization hardness is understood via its decision version / suitable reductions.

Exam strategy and final self-check

Construction questions

State the language, choose the model, draw all required transitions, mark start/final states, and test positive and negative inputs.

Proof questions

Use the correct quantifiers; for pumping choose a witness and cover all valid decompositions. For reductions start from a known hard problem.

Self-check

Can you classify $a^n b^n$, convert an ε-NFA, distinguish a recognizer from a decider, and explain NP-completeness? Lecture 45 provides solved exam questions.