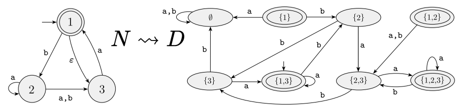
Reg / DFA / NFA (1)

	$\overline{\text{REG}}$	REG	CFL	Turing DECID.	Turing RECOG.	P	NP	NPC
$L1 \cup L2$	**no**	✓	✓	✓	✓	✓	✓	**no**
$L1 \cap L2$	**no**	✓	**no**	✓	✓	✓	✓	**no**
$\overline{L}$	✓	✓	**no**	✓	**no**	✓	?	?
$L1 \cdot L2$	**no**	✓	✓	✓	✓	✓	✓	**no**
L^*	**no**	✓	✓	✓	✓	✓	✓	**no**
$L^{\mathcal{R}}$		✓	✓	✓	✓	✓		
$L \cap R$		✓	✓	✓	✓	✓		
$L1 \setminus L2$		✓	**no**	✓	**no**	✓	?	

- (**DFA**) $M = (Q, \Sigma, \delta, q_0, F)$, $\delta : Q \times \Sigma \to Q$
- (**NFA**) $M = (Q, \Sigma, \delta, q_0, F)$, $\delta : Q \times \Sigma_\varepsilon \to \mathcal{P}(Q)$
- (**GNFA**) $(Q, \Sigma, \delta, q_0, q_a)$, $\delta : (Q \setminus \{q_a\}) \times (Q \setminus \{q_{\text{start}}\} \longrightarrow \mathcal{R}$ (where $\mathcal{R} = \{\text{all regex over } \Sigma\}$)
- GNFA accepts $w \in \Sigma^*$ if $w = w_1 \cdots w_k$, where $w_i \in \Sigma^*$ and there exists a sequence of states $q_0, q_1, \dots, q_k$ s.t. $q_0 = q_{\text{start}}$, $q_k = q_a$ and for each i, we have $w_i \in L(R_i)$, where $R_i = \delta(q_{i-1}, q_i)$.
- (**DFA-to-GNFA**) $G = (Q', \Sigma, \delta', s, a)$, $Q' = Q \cup \{s, a\}$, $\delta'(s, \varepsilon) = q_0$, For each $q \in F$, $\delta'(q, \varepsilon) = a$, ((TODO...))
- (**P.L.**) If A is a regular lang., then $\exists p$ s.t. every string $s \in A$, $|s| \geq p$, can be written as $s = xyz$, satisfying: (**i**) $\forall i \geq 0, xy^iz \in A$, (**ii**) $|y| > 0$ and (**iii**) $|xy| \leq p$.
- Every NFA can be converted to an equivalent one that has a single accept state.
- (**reg. grammar**) $G = (V, \Sigma, R, S)$. Rules: $A \to aB$, $A \to a$ or $S \to \varepsilon$. $(A, B, S \in V; a \in \Sigma)$.
- (**NFA $\rightsquigarrow$ DFA**)



  - $N = (Q, \Sigma, \delta, q_0, F)$
  - $D = (Q' = \mathcal{P}(Q), \Sigma, \delta', q_0' = E(\{q_0\}), F')$
  - $F' = \{q \in Q' \mid \exists p \in F : p \in q\}$
  - $E(\{q\}) := \{q\} \cup \{\text{states reachable from } q \text{ via } \varepsilon\text{-}$
  - $\forall R \subseteq Q, \forall a \in \Sigma, \delta'(R, a) = E\left(\bigcup_{r \in R} \delta(r, a)\right)$

CFL / CFG / PDA (2)

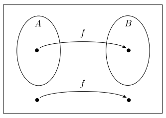
- (**CFG**) $G = (\underset{\text{n.t.}}{V}, \underset{\text{ter.}}{\Sigma}, R, S)$. Rules: $A \to w$. (where $A \in V$ and $w \in (V \cup \Sigma)^*$).
- A derivation of w is a **leftmost derivation** if at every step the leftmost remaining variable is the one replaced.
- w is derived **ambiguously** in G if it has at least two different l.m. derivations.
- G is **ambiguous** if it generates at least one string ambiguously.
- A CFG is ambiguous iff it generates some string with two different parse trees.
- (**P.L.**) If L is a CFL, then $\exists p$ s.t. any string $s \in L$ with $|s| \geq p$ can be written as $s = uvxyz$, satisfying: (**i**) $\forall i \geq 0, uv^ixy^iz \in L$, (**ii**) $|vxy| \leq p$, and (**iii**) $|vy| > 0$.
- (**CNF**) $A \to BC$, $A \to a$, or $S \to \varepsilon$, (where $A, B, C \in V$, $a \in \Sigma$, and $B, C \neq S$).
- If $G \in$ CNF, and $w \in L(G)$, then $|w| \leq 2^{|h|} - 1$, where h is the height of the parse tree for w.
- $L \in \textbf{CFL} \Leftrightarrow \exists \underset{\text{CFG}}{G} : L = L(G) \Leftrightarrow \exists \underset{\text{PDA}}{M} : L = L(M)$
- A CFL is **inherently ambiguous** if all CFGs that generate it are ambiguous.
- $\forall L \in$ CFL, $\exists G \in$ CNF : $L = L(G)$.
- REG $\subseteq$ CFL.
- $\{w \in \{a, b\}^* \mid w = w^{\mathcal{R}}\}$, $\{ww^{\mathcal{R}} \mid w \in \{a, b\}^*\}$, $\{a^nb^n \mid n \in \mathbb{N}\}$, $\{w \in \{\mathtt{a}, \mathtt{b}\}^* \mid \#_{\mathtt{a}}(w) = \#_{\mathtt{b}}(w)\} \in$ CFL but $\notin$ REG.
- $\{a^ib^jc^k \mid 0 \leq i \leq j \leq k\}$, $\{a^nb^nc^n \mid n \in \mathbb{N}\}$, $\{ww \mid w \in \{a, b\}^*\}$, $\{\mathtt{a}^{j^2} \mid j \geq 0\}$, $\{w \in \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}^* \mid \#_{\mathtt{a}}(w) = \#_{\mathtt{b}}(w) = \#_{\mathtt{c}}(w)\} \notin$ CFL
- (**derivation**) $S \Rightarrow u_1 \Rightarrow u_2 \Rightarrow \cdots \Rightarrow u_n = w$, where each u_i is in $(V \cup \Sigma)^*$. (in this case, G **generates** w (or S **derives** w), $S \overset{*}{\Rightarrow} w$)
- (**PDA**) $M = (Q, \underset{\text{input}}{\Sigma}, \underset{\text{stack}}{\Gamma}, \delta, q_0 \in Q, \underset{\text{accepts}}{F} \subseteq Q)$. (where Q, Σ, Γ, F finite). $\delta : Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \longrightarrow \mathcal{P}(Q \times \Gamma_\varepsilon)$.
- M **accepts** $w \in \Sigma^*$ if there is a seq. $r_0, r_1, \dots, r_m \in Q$ and $s_0, , s_1, \dots, s_m \in \Gamma^*$ s.t.:
  - $r_0 = q_0$ and $s_0 = \varepsilon$
  - For $i = 0, 1, \dots, m-1$, we have $(r_i, b) \in \delta(r_i, w_{i+1}, a)$, where $s_i = at$ and $s_{i+1} = bt$ for some $a, b \in \Gamma_\varepsilon$ and $t \in \Gamma^*$.
  - $r_m \in F$
- A PDA can be represented by a state diagram, where each transition is labeled by the notation " $a, b \to c$" to denote that the PDA: **Reads** a from the input (or read nothing if $a = \varepsilon$). **Pops** b from the stack (or pops nothing if $b = \varepsilon$). **Pushes** c onto the stack (or pushes nothing if $c = \varepsilon$)
- (**CSG**) $G = (V, \Sigma, R, S)$. Rules: $S \to \varepsilon$ or $\alpha A \beta \to \alpha \gamma \beta$ where: $\alpha, \beta \in (V \cup \Sigma \setminus \{S\})^*$; $\gamma \in (V \cup \Sigma \setminus \{S\})^+$; $A \in V$.

(3) TM, (4) Decidability

- (**TM**) $M = (Q, \underset{\text{input}}{\Sigma} \subseteq \Gamma, \underset{\text{tape}}{\Gamma}, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$, where $\sqcup \in \Gamma$ (**blank**), $\sqcup \notin \Sigma$, $q_{\text{reject}} \neq q_{\text{accept}}$, and $\delta : Q \times \Gamma \longrightarrow Q \times \Gamma \times \{\text{L}, \text{R}\}$
- (**unrecognizable**) $\overline{A_{TM}}$, $\overline{EQ_{\mathsf{TM}}}$, EQ_{CFG}, $\overline{HALT_{\mathsf{TM}}}$, $\text{REGULAR}_{\mathsf{TM}} = \{M \text{ is a TM and } L(M) \text{ is regular}\}$, E_{TM}, $EQ_{\mathsf{TM}} = \{M_1, M_2 \text{ are TMs and } L(M_1) = L(M_2)\}$
- (**recognizable**) accepts if $w \in L$, rejects/loops if $w \notin L$.
  - L is recognizable $\iff L \leq_{\text{m}} A_{\mathsf{TM}}$.
  - There exists some lang. that are unrecognizable.
  - A is **co-recognizable** if $\overline{A}$ is recognizable.
  - Every inf. rec. lang. has an inf. dec. subset.
  - (**rec. but undec.**)A_{TM}, $HALT_{\mathsf{TM}} = \{\langle M, w\rangle \mid M \text{ is a TM halts on } w\}$, $D = \{p \mid p \text{ is an int. poly. with an int. root}\}$, $\overline{EQ_{\mathsf{CFG}}}$, $\overline{E_{\mathsf{TM}}}$
- (**decidable**) accepts if $w \in L$, rejects if $w \notin L$.
- L is decidable $\iff L \leq_{\text{m}} 0^*1^*$.
  - A_{DFA}, A_{NFA}, A_{REX}, E_{DFA}, EQ_{DFA}, A_{CFG}, E_{CFG}, every CFL, every finite lang., A_{LBA}, $ALL_{\mathsf{DFA}} = \{\langle M\rangle \mid M \text{ is a DFA}, L(A) = \Sigma^*\}$, $A\varepsilon_{\mathsf{CFG}} = \{\langle G\rangle \mid G \text{ is a CFG that generates } \varepsilon\}$,
- L is dec. $\iff$ (L is rec. $\wedge$ L is co-rec.) $\iff$ $\exists$ TM
.
- (**decider**) TM that halts on all inputs.
- (**Rice**) Let P be a lang. of TM descriptions, s.t. (**i**) P is nontrivial (not empty and not all TM desc.) and (**ii**) for each two TM M_1 and M_2, we have $L(M_1) = L(M_2) \implies (\langle M_1\rangle \in P \iff \langle M_2\rangle \in P)$. Then P is undecidable.

(5) Mapping Reduction $\leq_{\text{m}}$

- $f : \Sigma^* \to \Sigma^*$ is **computable** if there exists a TM M s.t. for every $w \in \Sigma^*$, M halts on w and outputs $f(w)$ on its tape.



- A is **m. reducible** B (denoted by $A \leq_{\text{m}} B$), if there is a comp. func. $f : \Sigma^* \to \Sigma^*$ s.t. for every w, we have $w \in A \iff f(w) \in B$. (Such f is called the **m. reduction** from A to B.)
- If $A \leq_{\text{m}} B$ and B is decidable, then A is dec.
- If $A \leq_{\text{m}} B$ and A is undecidable, then B is undec.
- If $A \leq_{\text{m}} B$ and B is recognizable, then A is rec.
- If $A \leq_{\text{m}} B$ and A is unrecognizable, then B is unrec.
- (transitivity) If $A \leq_{\text{m}} B$ and $B \leq_{\text{m}} C$, then $A \leq_{\text{m}} C$.
- If A is recognizable and $A \leq_{\text{m}} \overline{A}$, then A is decidable.
- $A \leq_{\text{m}} B \iff \overline{A} \leq_{\text{m}} \overline{B}$

(7) Complexity, Polytime Reduction $\leq_{\text{P}}$

- ((**Running time**) decider M is a $f(n)$**-time TM**.) $f : \mathbb{N} \to \mathbb{N}$, where $f(n)$ is the max. num. of steps that DTM (or NTM) M takes on any n-length input (and any branch of any n-length input. resp.).
- $\mathsf{TIME}(t(n)) = \{L \mid L \text{ is dec. by } O(t(n)) \text{ DTM}\}$.
- $\mathsf{NTIME}(t(n)) = \{L \mid L \text{ is dec. by } O(t(n)) \text{ NTM}\}$.
- $\mathbf{P} = \bigcup_{k \in \mathbb{N}} \mathsf{TIME}(n^k)$
- (**verifier** for L) TM V s.t. $L = \{w \mid \exists c : V(\langle w, c\rangle) = \mathsf{accept}\}$.
  - (**certificate** for $w \in L$) str. c s.t. $V(\langle w, c\rangle) = \mathsf{accept}$.
- $\mathbf{NP} = \bigcup_{k \in \mathbb{N}} \mathsf{NTIME}(n^k)$
- $\mathbf{NP} = \{L \mid L \text{ is decidable by a PT verifier}\}$.
- $\text{P} \subseteq \text{NP}$.
- $f : \Sigma^* \to \Sigma^*$ is **PT computable** if there exists a PT TM M s.t. for every $w \in \Sigma^*$, M halts with $f(w)$ on its tape.
- A is **PT (mapping) reducible** to B, denoted $A \leq_{\text{P}} B$, if there exists a PT computable func. $f : \Sigma^* \to \Sigma^*$ s.t. for every $w \in \Sigma^*$, $w \in A \iff f(w) \in B$. (in such case f is called the **PT reduction** of A to B).
  - If $A \leq_{\text{P}} B$ and $B \in \text{P}$, then $A \in \text{P}$.
  - If $A \leq_{\text{P}} B$ and $B \leq_{\text{P}} A$, then A and B are **PT equivalent**, denoted $A \equiv_P B$. $\equiv_P$ is an equivalence relation on NP. $\text{P} \setminus \{\emptyset, \Sigma^*\}$ is an equivalence class of $\equiv_P$.
- $\textbf{NP-complete} = \{B \mid B \in \text{NP}, \forall A \in \text{NP}, A \leq_{\text{P}} B\}$.
- CLIQUE, SUBSET-SUM, SAT, 3SAT, VERTEX-COVER, HAMPATH, UHAMATH, $3COLOR \in$ NP-complete.
- $\emptyset, \Sigma^* \notin$ NP-complete.
- If $B \in$ NP-complete and $B \in \text{P}$, then $\text{P} = \text{NP}$.
- If $B \in$ NP-complete and $C \in \text{NP}$ s.t. $B \leq_{\text{P}} C$, then $C \in$ NP-complete.
- If $\text{P} = \text{NP}$, then $\forall A \in \text{P} \setminus \{\emptyset, \Sigma^*\}$, $A \in$ NP-complete.

Examples

- Counterexamples:
  - $A \leq_{\text{m}} B$ and $B \in$ REG, but, $A \notin$ REG: $A = \{0^n 1^n \mid n \geq 0\}$, $B = \{1\}$, $f : A \to B$, $f(w) = \begin{cases} 1 & \text{if } w \in A \\ 0 & \text{if } w \notin A \end{cases}$.
  - $L \in$ CFL but $\overline{L} \notin$ CFL: $L = \{x \mid \forall w \in \Sigma^*, x \neq ww\}$, $\overline{L} = \{ww \mid w \in \Sigma^*\}$.
  - $L_1, L_2 \in$ CFL but $L_1 \cap L_2 \notin$ CFL: $L_1 = \{a^n b^n c^m\}$, $L_2 = \{a^m b^n c^n\}$, $L_1 \cap L_2 = \{a^n b^n c^n\}$.
- $A \leq_{\text{P}} B$ and $f : A \to B$ s.t. $w \in A \iff f(w) \in B$ and f is poly-time comp.
  - SAT $\leq_{\text{P}}$ DOUBLE-SAT
    - $f(\phi) = \phi \wedge (x \vee \neg x)$
  - SUBSET-SUM $\leq_{\text{P}}$ SET-PARTITION
    - $f(\langle x_1, \ldots, x_m, t\rangle) = \langle x_1, \ldots, x_m, S - 2t\rangle$, where S sum of $x_1, \ldots, x_m$, and t is the target subset-sum.
  - $3COLOR \leq_{\text{P}} 3COLOR_{almost}$
    - $f(\langle G\rangle) = \langle G'\rangle$, where $G' = G \cup K_4$
  - VERTEX-COVER $\leq_{\text{P}}$ WVC
    - $f(\langle G, k\rangle) = (G, w, k)$, $\forall v \in V, w(v) = 1$.
  - $\underset{\text{length } \geq k}{\text{SimplePATH}} \leq_{\text{P}}$ UHAMATH
  - $\underset{\text{undir. } G \text{ has } k\text{-clique}}{\text{CLIQUE}} \leq_{\text{P}} \underset{\text{undir. } G \text{ has } |V|/2\text{-clique}}{\text{HALF-CLIQUE}}$
    - $f(\langle G = (V, E), k\rangle) = \langle G' = (V', E')\rangle$, if $k = \frac{|V|}{2}$, $E = E'$, $V' = V$. if $k > \frac{|V|}{2}$, $V' = V \cup \{j = 2k - |V| \text{ new nodes}\}$. if $k < \frac{|V|}{2}$, $V' = V \cup \{j = |V| - 2k \text{ new nodes}\}$ and $E' = E \cup \{\text{edges for new nodes}\}$
  - CLIQUE $\leq_{\text{P}}$ INDEPENDENT-SET
  - SET-COVER $\leq_{\text{P}}$ VERTEX-COVER
  - 3SAT $\leq_{\text{P}}$ SET-SPLITTING
  - INDEPENDENT-SET $\leq_{\text{P}}$ VERTEX-COVER
  - VERTEX-COVER $\leq_{\text{p}}$ CLIQUE