Reg/DFA/NFA (1)

- (**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) $(Q, \Sigma, \delta, q_0, q_{\mathrm{a}})$, $\delta : (Q \setminus \{q_{\mathrm{a}}\}) \times (Q \setminus \{q_{\mathrm{start}}\} \longrightarrow \mathcal{R}$ (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 $q_0, q_1, \ldots, q_k$ s.t. $q_0 = q_{\mathrm{start}}$, $q_k = q_{\mathrm{a}}$ 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 $q \in F$, $\delta'(q, \varepsilon) = a$,
- (**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.
- (**regular grammar**) $G = (V, \Sigma, R, S)$. Rules: $A \to aB$, $A \to a$ or $S \to \varepsilon$. ($A, B, S \in V$ and $a \in \Sigma$).

	N-Reg	Reg	CFL	TD	TR	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**	✓	?	

CFG (2)

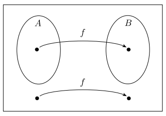
- (**CFG**) $G = (\underset{\text{n.t.}}{V}, \underset{\text{ter.}}{\Sigma}, R, S)$. Rules: $A \to w$. (where $A \in V$ and $w \in (V \cup \Sigma)^*$).
  - A derivation of w is a **leftmost derivation** if at every step the leftmost remaining variable is the one replaced.
  - w is derived **ambiguously** 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 is a CFG in CNF, and $w \in L(G)$, then $|w| \leq 2^{|h|} - 1$, where h is the height of the parse tree for w.
  - Every CFL is generated by a CFG in CNF.
- L is **CFL** if it is generated by some CFG.
  - A CFL is **inherently ambiguous** if all CFGs that generate it are ambiguous.
  - Every CFL is generated by a CFG in CNF.
  - Every regular lang. is 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. $r_0, r_1, \ldots, r_m \in Q$ and $s_0, , s_1, \ldots, s_m \in \Gamma^*$ s.t.:
  - $r_0 = q_0$ and $s_0 = \varepsilon$
  - For $i = 0, 1, \ldots, m-1$, 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: $\alpha, \beta \in (V \cup \Sigma \setminus \{S\})^*$; $\gamma \in (V \cup \Sigma \setminus \{S\})^+$; $A \in V$.

TM (3), Decidability (4)

- (**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 $\delta : Q \times \Gamma \longrightarrow Q \times \Gamma \times \{\text{L}, \text{R}\}$
- (**unrec.**) $\overline{A_{TM}}$, $\overline{EQ_{\mathsf{TM}}}$, EQ_{CFG}, $\overline{HALT_{\mathsf{TM}}}$, $\text{REGULAR}_{\mathsf{TM}} = \{M \text{ is a TM and } L(M) \text{ is regular}\}$, E_{TM}, $EQ_{\mathsf{TM}} = \{M_1, M_2 \text{ are TMs and } L(M_1) = L(M_2)\}$
- (**rec.**) accepts if $w \in L$, rejects/loops if $w \notin L$.
  - There exists some languages that are not rec.
- (**co-TR**) if its complement is rec.
  - Every inf. rec. lang. has an inf. dec. subset.
  - (**rec. but not undec**)A_{TM}, $HALT_{TM} = \{\langle M, w\rangle \mid M \text{ is a TM halts on } w\}$, $D = \{p \mid p \text{ is an int. poly. with an int. root}\}$, $\overline{EQ_{\mathsf{CFG}}}$, $\overline{E_{\mathsf{TM}}}$
- (**dec.**) accepts if $w \in L$, rejects if $w \notin L$.
  - A_{DFA}, A_{NFA}, A_{REX}, E_{DFA}, EQ_{DFA}, A_{CFG}, E_{CFG}, every CFL, every finite lang., A_{LBA}, ALL_{DFA}, $A\varepsilon_{\mathsf{CFG}}$,
- L is dec. $\iff$ L is rec. $\wedge$ L is co-TR $\iff$ $\exists$TM de .
- (**decider**) TM that halts on all inputs.
- (**Rice**) Let P be a lang. of TM desc., such that (*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.

Reducibility (5)

- $f : \Sigma^* \to \Sigma^*$ 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 $A \leq_{\mathrm{m}} B$), 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.)
- (5.22) If $A \leq_{\mathrm{m}} B$ and B is dec., then A is dec.
- (5.23) If $A \leq_{\mathrm{m}} B$ and A is undec., then B is undec.
- (5.28) If $A \leq_{\mathrm{m}} B$ and B is rec., then A is rec.
- (5.29) If $A \leq_{\mathrm{m}} B$ and A is not rec., then B is not rec.
- (e5.6) If $A \leq_{\mathrm{m}} B$ and $B \leq_{\mathrm{m}} C$, then $A \leq_{\mathrm{m}} C$.

Complexity (7)

- ((**Run. 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. $V(\langle w, c\rangle) = \mathsf{accept}$.
- $\mathbf{NP} = \bigcup_{k\in\mathbb{N}} \mathsf{NTIME}(n^k)$
- $\mathbf{NP} = \{L \mid L \text{ is decidable by a PT verifier}\}$.
- $\text{P} \subseteq \text{NP}$.
- $\text{CLIQUE} = \{\langle G, k\rangle \mid G \text{ is an undir. g. with a } k\text{-clique}\}$.
- $\text{SUBSET-SUM} = \{\langle S, k\rangle \mid S \text{ is a m. set of int.} \wedge \exists T \subseteq S : \sum_{x\in T} x = k\}$.
- $f : \Sigma^* \to \Sigma^*$ 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 $A \leq_P B$, 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 $A \leq_P B$ and $B \in \text{P}$, then $A \in \text{P}$.
  - If $A \leq_P B$ and $B \leq_P A$, 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 $\equiv_P$.
- $\textbf{NP-complete} = \{B \mid B \in \text{NP}, \forall A \in \text{NP}, A \leq_P B\}$.
- $\text{CLIQUE, SUBSET-SUM, SAT, 3SAT, VERTEX-COVER, HAMPATH, UHAMATH,}$ $3COLOR \in \text{NP-complete.}$
- $\emptyset, \Sigma^* \notin \text{NP-complete.}$
- If $B \in \text{NP-complete}$ and $B \in \text{P}$, then $\text{P} = \text{NP}$.
- If $B \in \text{NP-complete}$ and $C \in \text{NP}$ s.t. $B \leq_P C$, then $C \in \text{NP-comp.}$
- If $\text{P} = \text{NP}$, then $\text{NP-complete} = \text{P} = \text{NP}$.