

► **k-Turing Machine** $M = (Q, \Sigma, \Gamma, \Sigma', \delta, \square, q_0, q_+, q_-)$

Q states, $q_0, q_+, q_- \in Q$

Σ input symbols

Γ working symbols, $\square \in \Gamma$

Σ' output symbols

$\delta : (Q \setminus \{q_+, q_-\}) \times (\Sigma \cup \{\vdash, \dashv\}) \times \Gamma^k \rightarrow \mathcal{B}$, no moving out of the input bounds

$\mathcal{B} = \mathcal{P}_{\neq \emptyset}(Q \times \Gamma^k \times \{-1, 0, 1\}^{k+1} \times (\Sigma' \cup \{\epsilon\}))$

Configuration $(q, B_1, \dots, B_k, i_0, \dots, i_k) \in Q \times (\Gamma^+)^k \times \mathbb{N}^{k+1}$

► **Number of configurations** There exists at most $a \cdot |w| \cdot b^n$ ($a, b \in \mathbb{N}$) configurations that have at most length $n \in \mathbb{N}$. For f -space-bound TM: $a \cdot |w| \cdot b^{f(|w|+2)}$.

► **f is time-constructible** if $f(n) \geq n \ \forall n \in \mathbb{N}$ and \exists det. TM that stops on a^n after exactly $f(n+2)$ steps.

► **f is space-constructible** if $f(n) \geq \log n \ \forall n \in \mathbb{N}$ and \exists f -space-bounded det. TM that computes $a^n \mapsto a^{f(n+2)}$.

► **Linear speed-up** Let $L \in \text{DTIME}(f)$ with $f(n) \geq n \ \forall n \in \mathbb{N}$ and $\epsilon > 0$. Then $L \in \text{DTIME}(g)$ with $g(n) = \lceil \epsilon f(n) \rceil \ \forall n \in \mathbb{N}$.

► **Linear tape compression** $\text{DSPACE}(f) \subseteq \text{DSPACE}(g) \ \forall \epsilon > 0$ and $g(n) = \lceil \epsilon f(n) \rceil \ \forall n \in \mathbb{N}$

► **Space hierarchy** f space-constructible. $f(n) \geq \log n$.

Then $\text{DSPACE}(o(f)) \neq \text{DSPACE}(\mathcal{O}(f))$.

► **Time hierarchy** $\text{DTIME}(o(f)) \subsetneq \text{DTIME}(\mathcal{O}(f; g))$ with $g(n) = n \log n \ \forall n \in \mathbb{N}$

► **Klein-O** $g \in o(f)$ iff $\lim_{x \rightarrow \infty} \frac{g(x)}{f(x)} = 0$. L'Hospital applicable if both converge to 0 or ∞ .

► **Savitch** $\text{NSPACE}(f) \subseteq \text{DSPACE}(f; g)$ with $g(n) = n^2 \ \forall n \in \mathbb{N}$

► **Immerman & Szelepcsényi** $\text{NSPACE}(f) = \text{co-NSPACE}(f)$

► **Padding** $L \subseteq \Sigma^*$, $f(n) \geq n \ \forall n \in \mathbb{N}$, $\# \notin \Sigma$. $\text{Pad}_f^\#(L) = \{w \#^{f(|w|+2)-|w|} \mid w \in L\}$

► **Time - Translation** $f(n) \geq n \leq g(n) \ \forall n \in \mathbb{N}$, g time-constructible, output $\#^{f(n+2)}$ computable in time $g(f(n+2)+2)$ on input $\#^n$, $L \subseteq \Sigma^*$, $\# \notin \Sigma$.

▪ $\text{Pad}_f^\#(L) \in \text{N/DTIME}(\mathcal{O}(g)) \Leftrightarrow L \in \text{N/DTIME}(\mathcal{O}(f; g))$

► **Space - Translation** $f(n) \geq n$, $g(n) \geq \log n \ \forall n \in \mathbb{N}$, g space-constructible, binary repr. of $f(n+2)$ computable in space $g(f(n+2)+2)$ on input $\#^n$, $L \subseteq \Sigma^*$, $\# \notin \Sigma$.

▪ $\text{Pad}_f^\#(L) \in \text{N/DSPACE}(g) \Leftrightarrow L \in \text{N/DSPACE}(f; g)$

► **Reduction** $L \subseteq \Sigma^*$, $L' \subseteq (\Sigma')^*$. $L \preceq L'$ iff $w \in L \Leftrightarrow f(w) \in L'$ for some *Reduction* f . (log-space: \preceq_L , polynomial-time: \preceq_P) If $L' \in \mathbf{P}$ and $L \preceq_P L'$ then $L \in \mathbf{P}$.

► **Complete Problems** \mathcal{C} a class, L a problem.

▪ \mathcal{C} -hard if $L' \preceq_L L \ \forall L' \in \mathcal{C}$

▪ \mathcal{C} -complete if L \mathcal{C} -hard and $L \in \mathcal{C}$

► **Schöning** $L_1, L_2 \subseteq \Sigma^*$ decidable, $\mathcal{C}_1, \mathcal{C}_2$ (effective) recur. enum. classes of decidable languages over Σ^* and closed under finite variation, $L_1 \notin \mathcal{C}_1$, $L_2 \notin \mathcal{C}_2$. \Rightarrow decidable $L \subseteq \Sigma^*$ with $L \notin \mathcal{C}_1 \cup \mathcal{C}_2$. If $L_1 \in \mathbf{P}$ and $\emptyset \neq L_2 \neq \Sigma^*$, then also $L \preceq_P L_2$.

► **Ladner** If $\mathbf{P} \neq \mathbf{NP}$ then there eff. exists $L \in \mathbf{NP} \setminus \mathbf{P}$ that is not \mathbf{NP} -complete under \preceq_P .

► **Sparse** $L \subseteq \Sigma^*$ is *sparse* if there \exists polynomial p s.t. $|L \cap \Sigma^n| \leq p(n) \ \forall n \in \mathbb{N}$.

► **Mahaney** If $\mathbf{P} \neq \mathbf{NP}$. Then there exists no sparse language that is \mathbf{NP} -hard under \preceq_P (or \preceq_L).

► **Boolean circuit** Finite directed labeled graph $C = (V, E, l)$ for languages $\{0, 1\}^*$

▪ $V = \{1, \dots, o\}$, $o \in \mathbb{N}$, $v < v' \ \forall (v, v') \in E$

▪ $l : V \rightarrow \{\wedge, \vee, \neg, 0, 1\} \cup \{x_1, \dots, x_n\}$ labeling gates

▪ *binary* if input rank is ≤ 2 , *monotone* if no negation, *constant* if no inputs

▪ $I(w)$ is evaluation of C for w

▪ size of C is $|C| = |V|$, *depth* $dp(C)$ of C is length of maximal path in (V, E)

▪ $\mathcal{C} = (C_n)_{n \in \mathbb{N}}$ is *family* of Boolean circuits

▪ There ex. circuit family $\mathcal{C} = (C_n)_{n \in \mathbb{N}}$ that defines L such that $|C_n| \leq 2^n + 2n + 3$ and $dp(C_n) \leq 3 \ \forall L \subseteq \{0, 1\}^*$

▪ There ex. circuit family $\mathcal{C} = (C_n)_{n \in \mathbb{N}}$ of binary circuits C_n that defines L and $|C_n| \leq n2^n + 2n + 1$ and $dp(C_n) \leq 1 + \lceil \log n^+ \rceil + n$

► **c-polynomial circuit family** $\mathcal{C} = (C_n)_{n \in \mathbb{N}}$, $f : \mathbb{N} \rightarrow \mathbb{N}$. \mathcal{C} is *f-size-bounded* if $|C_n| \leq f(n) \ \forall n \in \mathbb{N}$. *c-polynomial* if there is polynomial p s.t. \mathcal{C} is p -size-bounded.

► **Shannon** Fraction of n -ary Boolean functions with binary circuits of size smaller than $\frac{2^n}{3n}$ tends to 0 for $n \rightarrow \infty$

► **Uniform family** $\mathcal{C} = (C_n)_{n \in \mathbb{N}}$ is uniform if there exists log-space computable function f with $f(1^n) = C_n \ \forall n \in \mathbb{N}$

► **Nick's class** $k \in \mathbb{N}$. \mathbf{NC}^k class of languages defined by c-polynomial uniform binary circuit families $\mathcal{C} = (C_n)_{n \in \mathbb{N}}$ with $dp(C_n) \in \mathcal{O}(\log^k n) \ \forall n \in \mathbb{N}$. \mathbf{AC}^k for non-binary circuits.

► **Razbarov's Theorem** Attempt to prove $\mathbf{P} \neq \mathbf{NP}$. He showed that there is a monotone language in \mathbf{NP} that has no c-polynomial monotone bool. circuit. If one could show that all monotone languages in \mathbf{P} have c-pol. mon. bool. circuit, then $\mathbf{P} \neq \mathbf{NP}$.

► **Oracle TM** TM with additional write-only, auto-advance output tape and special states $q_y, q_n, q_?$. For $O \subseteq \Sigma^*$, spontaneous transition from $q_?$ to q_+ or q_- .

▪ $\mathbf{P}^O = \{L(M^O) \mid \text{det. polyn.-time oracle TM } M^O\}$

▪ $\mathbf{NP}^O = \{L(M^O) \mid \text{polyn.-time oracle TM } M^O\}$

▪ $O \in \mathbf{P}^O$

▪ There exists $O \in \mathbf{PSPACE}$ s.t. $\mathbf{P}^O = \mathbf{NP}^O$.

▪ There exists decidable $O \subseteq \{0, 1\}^*$ s.t. $\mathbf{P}^O \neq \mathbf{NP}^O$.

► **Polynomial Turing reduction** A, B problems. $A \preceq_P^T B$ if there ex. det. polyn.-time oracle TM M with $A = L(M^B)$. M uses problem B as Oracle.

► **Probabilistic TM** TM time-bounded by polynomial p is *probabilistic* if all computations for input w have exactly length $p(|w|)$ and exactly 2 nondeterministic alternatives in each step.

▪ Monte-Carlo accepts $L \subseteq \Sigma^*$ if $\forall w \in \Sigma^*$, $w \notin L$ iff all computations reject and $w \in L$ iff at least $\frac{2}{3}$ of all computations accept. \mathbf{RP} set of Monte-carlo accepted languages.

▪ $\text{PIT} \in \text{co-RP}$. Test if two polynomials are identical.

▪ Monte-Carlo algorithms have uncertain correctness, Las Vegas algorithms have uncertain runtime.

▪ „Zero-error Probabilistic Polynomial time“ = $\mathbf{ZPP} = \mathbf{RP} \cap \text{co-RP}$. Result is always correct, and expected runtime is in \mathbf{P} . Run Monte-Carlo TMs for L and L^C in parallel until one accepts and the other rejects.

▪ majority accepts $L_{\text{MAJ}}(M) = \{w \in \Sigma^* \mid \text{more than half of all computations accept } w\}$

▪ \mathbf{PP} set of majority accepted languages

▪ \mathbf{BPP} qualified majority e.g. $w \in L$ iff $> \frac{2}{3}$ runs accept w , $w \notin L$ iff $> \frac{2}{3}$ runs reject.

▪ $\mathbf{P} \subseteq \mathbf{ZPP} = \text{co-ZPP} = \mathbf{RP} \cap \text{co-RP} \subseteq \mathbf{RP} \subseteq \mathbf{NP} \subseteq \mathbf{PP} = \text{co-PP} \subseteq \mathbf{PSPACE}$

▪ $\mathbf{RP} \subseteq \mathbf{BPP} = \text{co-BPP} \subseteq \mathbf{PP}$

- $\text{co-RP} \subseteq \text{BPP}$
- $\text{ZPP} \subseteq \text{co-RP} \subseteq \text{co-NP} \subseteq \text{PP}$

► **Randomized TM** det. TM with input tape Z of infinite random bits. f -time-bounded if classic def. holds for every content of Z . Accepts w with prop. $\frac{\#Z \in \{0,1\}^{f(|w|)} \mid w \text{ accepted}}{2^{f(|w|)}}$

► **Interactive Proof System** Pair (A, B) , $C = \{0, 1\}$, $A : \bigcup_{i \in \mathbb{N}} (C^*)^{1+2i} \rightarrow C^*$, p, q polyn., p -time-bounded rand. TM B . $q(|w|)$ rounds. A and B share tape W . Round i :

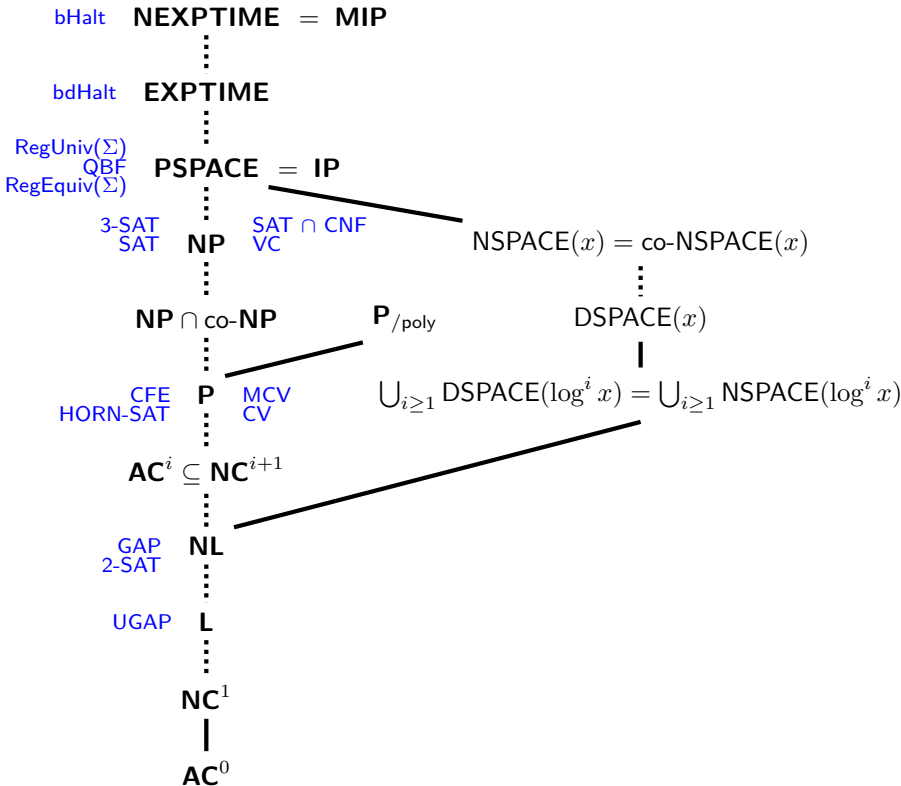
- $a_i = A(w, a_1, b_1, \dots, a_{i-1}, b_{i-1})$
- ← $b_i = B(w, a_1, b_1, \dots, a_{i-1}, b_{i-1}, Z_1, \dots, Z_i)$

In round $q(|w|)$ Bob decides whether to accept w . (A, B) accepts L if for all w :

- $w \in L$, then Bob accepts input w with prop. $\geq 1 - \frac{1}{2^{|w|}}$
- $w \notin L$, then no (A', B) exists s.t. Bob accepts w with probability $\geq \frac{1}{2^{|w|}}$

- **IP** set of languages defined by IPs
- Non-isomorphism of finite graphs is in **IP**
- **IP** is closed under \preceq_P

Note: Alice tries to convince Bob that $w \in L$. Bob can't trust Alice and has to verify. Usually this is done by sending Alice a decision problem, that she can only correctly decide with probability $\frac{1}{2^n}$. Switching equal colored socks n times, she has to decide if there was a switch or not.



$$\text{DTIME}(f) \subseteq \text{NTIME}(f) \subseteq \text{DSPACE}(f) \subseteq \text{NSPACE}(f) \subseteq \text{DTIME}(2^{\mathcal{O}(f)})$$

Hennie & Stearns: We can emulate k -band in 2-band with log overhead

$$\mathbf{L} \subseteq \mathbf{NL} \subseteq \text{DTIME}(2^{\mathcal{O}(\log x)}) = \mathbf{P}$$

$$\mathbf{NL} \subsetneq \mathbf{CSL} = \mathbf{LBA} = \text{NSPACE}(x) \subseteq \text{DTIME}(2^{\mathcal{O}(x)})$$

$$\text{DSPACE}(x^2) \subseteq \text{DTIME}(2^{\mathcal{O}(x^2)})$$

PSPACE = $\bigcup_{i \in \mathbb{N}_+} \text{DSPACE}(x^i) = \bigcup_{i \in \mathbb{N}_+} \text{NSPACE}(x^i)$ since $\text{NSPACE}(x^i) \subseteq \text{DSPACE}(x^{2i})$ by Savitch

$$\mathbf{L} \subsetneq \text{DSPACE}(\log^2 x) \subsetneq \text{DSPACE}(x) \subseteq \text{NSPACE}(x) \subsetneq \mathbf{PSPACE}$$

$$\text{DTIME}(\mathcal{O}(x)) \subsetneq \text{DTIME}(\mathcal{O}(x^2)) \subsetneq \mathbf{P}$$

$$\mathbf{P} \subsetneq \text{DTIME}(\mathcal{O}(2^x)) \subsetneq \text{DTIME}(\mathcal{O}(2 + \epsilon)^x)$$

$$\mathbf{DCSL} = \text{DSPACE}(x) \neq \text{NSPACE}(x) = \mathbf{CSL} \Rightarrow \mathbf{L} \neq \mathbf{NL}$$

$$\mathbf{P} \neq \text{DSPACE}(x) = \mathbf{DCSL}$$

$$L \in \mathbf{L} \Leftrightarrow \exists \phi \in \text{FO}(\text{CTC}): L = \{w \in \Sigma^* \mid w \models \phi\}$$

$$L \in \mathbf{NL} \Leftrightarrow \exists \phi \in \text{FO}(\text{TC}): L = \{w \in \Sigma^* \mid w \models \phi\}$$

$$\mathbf{PRIM} \in \mathbf{NP} \cap \text{co-NP}$$

There are **NP**-complete problems \Rightarrow there are **NP**-complete problems in $\text{NTIME}(x)$

If $\mathbf{P} \neq \mathbf{NP}$, then there eff. exists $L \in \mathbf{NP} \setminus \mathbf{P}$ that is not **NP**-complete under \preceq_L

► **NL-complete** $\text{GAP} \in \text{DSPACE}(\log^2 x) \in \text{NSPACE}(\log x) \in \mathbf{NL}$. Any log-space-bounded TM can be transformed into a dir. graph with polynomial size. Each configuration (limited by $2^{\mathcal{O}(\log x)}$) resembles a vertex. Decide if accepting conf. is reachable from starting conf. with **GAP**.

We can reduce 2-SAT to **GAP** and vice-versa. Each disjunction $\neg x_1 \Rightarrow x_2$ of 2-SAT is edge in graph. If we can reach x_1 from $\neg x_1$ and vice-versa, the formula contains contradiction.

► **P-complete** $\text{CFE} \in \mathbf{P-c}$ because we can construct grammar G_w for TM M with input w that produces ϵ from final configuration in log space. If grammar produces empty language $w \notin L_M$. $\text{CV} \in \mathbf{P-c}$ because we can construct bool. circuit from G_w that evaluates to 1 iff $w \in L$. $\text{HORN-SAT} \in \mathbf{P-c}$ because we can reduce bool. circuit to HORN-SAT formula.

► **NP-complete** $\text{SAT} \in \mathbf{NP-c}$ because we can construct a huge formula from any TM that is satisfiable iff $w \in L(M)$. Create constraints for the configurations at each point in time. We can also create CNF from every prop. formula which is also a 3-SAT formula, hence CNF, 3-SAT $\in \mathbf{NP-c}$. All of these are obviously in **NP** because we can guess an interpretation and verify it.

► **(N)EXPTIME-complete** bdHalt is **EXPTIME**-complete because we can simulate any TM M that is time-bounded by $2^{p(|w|+2)}$ and see if it halts on input w . Input of bdHalt would be binary repr. of M and binary repr. of $2^{p(|w|+2)}$ which can be calculated in log-space. The same applies to $\text{bHalt} \in \mathbf{NEXPTIME-c}$.