O-notation: $f(n) = O(g(n))$, f grows at most as fast as g, if there exists $c > 0$ and $n_0 > 0$ such that for all $n \geq n_0, f(n) \leq c \cdot g(n)$

Turing machine: $M = (K, \Sigma, \delta, s)$, states K, symbols Σ, transition function δ, starting state s, starting symbol $\triangleright$, blank symbol $\sqcup$, halting states $\{h, yes, no\}$

Transition function: $\delta(q, \sigma) = (p, \rho, D)$, current state q, current symbol σ, new state p, replacing symbol ρ, cursor direction $D \in \{\rightarrow, \leftarrow, -\}$

Configuration: (q, w, u), current state q, string left of cursor including scanned symbol w, string right of cursor u, relations $\rightarrow^M, \rightarrow^{M^t}, \rightarrow^{M^*}$

---

k-tape Turing machine: $M = (K, \Sigma, \delta, s)$

Transition function: $\delta(q, \sigma_1, ..., \sigma_k) = (p, \rho_1, D_1, ..., \rho_k, D_k)$

Configuration: $(q, w_1, u_1, ..., w_k, u_k)$

With input and output: First tape is read-only and last tape is write only.

Runtime is t if $(s, \triangleright, x, \triangleright, \epsilon, ..., \triangleright, \epsilon) \rightarrow^{M^t} (H, w_1, u_1, ..., w_k, u_k)$, where x is the input and $H \in \{h, yes, no\}$. Output string $w_k u_k$ is read from kth tape.

Space usage: $\sum_{i=2}^{k-1} |w_i u_i|$ where $|w_i u_i|$ is the length of the concatenation of strings w_i and u_i.

Language: $L \subseteq (\Sigma - \{\sqcup\})^*$, complement $\bar{L} = \Sigma^* - L$

Decidable/Recursive $\forall x \in (\Sigma - \{\sqcup\})^*$: If $x \in L$, then $M(x) = yes$ and if $x \notin L$, then $M(x) = no$

Semidecidable/Recursively enumerable $\forall x \in (\Sigma - \{\sqcup\})^*$: If $x \in L$, then $M(x) = yes$ and if $x \notin L$, then $M(x) = \nearrow$

Universal Turing machine: $U(M; x) = M(x)$, simulates machine M on input x

Halting problem: Does M halt on x? Language $H = \{M; x \mid M(x) \neq \nearrow\}$ is semidecidable and undecidable (All $M; x$ where M halts on x). Proof:

1) Assume TM M_H decides H
2) $D(M)$: **if** $M_H(M; M) = yes$ **then** $\nearrow$ **else** *yes*
3) $D(D)$ has no satisfactory result. We have condadictions:
   - $D(D) \neq \nearrow$, then $M_H(D; D) = yes$ and $D(D) = \nearrow$
   - $D(D) = \nearrow$, then $M_H(D; D) = no$ and $D(D) \neq \nearrow$

Undecidability: Reduce deciding H to deciding A with reduction t.

1) $M_H(M; x)$: $y \leftarrow M_t(M; x)$; **return** $M_A(y)$

Example: Reduction of H to $T = \{M | M \text{ halts on all inputs}\}$

1) $M_x(y)$: **if** $y = x$ **then** $M(x)$ **else** halt
2) Define reduction mapping $t(M; x) = M_x$
3) $M; x \in H$ iff M halts on x iff M_x halts on all inputs iff $M_x \in T$

Complexity class: C is set of all languages decided by Turing machine M operating in appropriate mode such that M expends at most $f(|x|)$ units of specified resource for any input x

Complexity classes: **TIME**(f), **SPACE**(f), **NTIME**(f), **NSPACE**(f), f is proper complexity function

Closure under complement: Deterministic time and space classes are closed under complement, that is, if $L \in C$, then $\bar{L} \in C$

Class inclusion: **L** $\subseteq$ **NL** $\subseteq$ **P** $\subseteq$ **NP** $\subseteq$ **PSPACE** $\subseteq$ **EXP**

Reduction $A \leq B$: Language A is reducible to language B. A is at most as hard as B. Algorithm R that transforms any instance x of A to *equivalent* instance $R(x)$ of B.

Logspace reduction $A \leq_{\mathrm{L}} B$: Algorithm R is computable by deterministic Turing machine in $O(\log n)$ space.

C-hard: For every $L' \in C$ is we have $L' \leq_{\mathrm{L}} L$.

C-complete: If $L \in C$ then for every $L' \in C$ we have $L' \leq_{\mathrm{L}} L$.

C is **closed under reductions** if $L' \leq_{\mathrm{L}} L$ and $L \in C$, then $L' \in C$

Computation table method: Used to establish the first complete problems for complexity class.

Circuit value is **P**-complete

Circuit sat is **NP**-complete

Boolean expression: variables $X = \{x_1, x_2, ..., x_n\}$, connectives $\wedge, \vee, \neg$, literals $x_i, \neg x_i$

Conjunctive normal form (CNF):

Disjunctive normal form (DNF):

n-ary Boolean function $f : \{0,1\}^* \rightarrow \{0,1\}$, 2^{2^n} functions

Boolean function f to boolean expression ϕ: Compute truth table from f. Convert truth table to CNF formula ϕ. Worst case size of ϕ is $O(n2^n)$.