A deterministic TM is said to be in \( \text{SPACE}(s(n)) \) if it uses space \( O(s(n)) \) on inputs of length \( n \). Similarly, it is in \( \text{TIME}(t(n)) \) if it uses time \( O(t(n)) \) on such inputs.

A language \( L \) is polynomial-time decidable if \( \exists \) \( K \) and a TM \( M \) to decide \( L \) s.t.

\[ M \in \text{TIME}(n^K) \]

Note that \( K \) is independent of \( n \).

E.g., PATH, i.e. does there exist a path between \( s \) and \( t \) in a given graph, has a polynomial-time decider.

Median

Min/Max weight spanning tree.
$P$ is the class of languages with polynomial time TMs.

$P = \bigcup_{k} \text{TIME}(n^k)$

Do all decidable languages belong to $P$?

$\underline{HAM\ PATH}$ \exists path from $s$ to $t$ that visits every vertex in $G$ exactly once?

$\underline{SAT}$ Given a boolean formula, \exists a setting of its variables that makes the formula true?

$f = (x_1 \lor x_2 \lor \overline{x_3}) \land (\overline{x_1} \lor x_3 \lor \overline{x_2}) \land (x_1 \lor \overline{x_3} \lor x_2)$

No polytime algorithms known for these problems. They can be solved (decided) by nonpolytime nondeterministic TMs.

"guess" the path

"guess" the assignment
Recall that a NTM accepts iff any one of its computation paths accepts. The path amounts to a verification of the YES answer.

We have \( \text{NSPACE}(4(n)) \) and \( \text{NTIME}(t(n)) \)

\( \text{NP} \) is the class of languages that can be decided by polynomial-time NTMs.

\[
\text{NP} = \bigcup_k \text{NTIME}(n^k)
\]

Alternatively, \( \text{NP} \) is the class of languages with the property that membership ("YES") can be verified in polynomial-time using a polynomial-sized certificate.

E.g. \( \text{SAT} \): if \( \Phi \) is satisfiable, a valid assignment is the certificate.

\( \text{HAMPATH} \): if \( G \) has a HAM path then the sequence of vertices visited is the certificate.
Clearly $P \subseteq NP$

From Savitch’s theorem,

$$NPSPACE = PSPACE$$

Since the space requirement only squares.

Also $NTIME(t(n)) \subseteq DTIME(2^{O(t(n))})$

EXPTIME := Languages that can be decided in exponential time.

$$P \subseteq NP \subseteq PSPACE \subseteq EXP$$

Amazingly, we do not know if these containments are strict, i.e., if a language $L$

- $L \in EXP$ and $L \notin PSPACE$
- $L \in PSPACE$ and $L \notin NP$
- $L \in NP$ and $L \notin P$

We know that $P \subseteq EXP$ from the hierarchy theorem.
\[ L \in \text{NP} \iff \exists \text{NTM } M \text{ s.t. } \exists x \mid L = \{ x \mid \exists \text{accepting path in } M \text{ on input } x \}\]

The class of languages that are complements of languages in NP is called CoNP.

\[ L \in \text{CoNP} \iff \exists \text{NTM } M \text{ s.t. } \exists x \mid L = \{ x \mid \text{an accepting path of } M \text{ is accepting for } x \}\]

\[ L \in \text{CoNP} \iff \exists L \in \text{NP} \iff \exists x \mid L = \{ x \mid x \notin L \}\]


— L is rejected on every path —

How to verify membership in a CoNP language?

Short (polynomial-size) certificate that \( x \notin L \), e.g. \( x \) does not have a HAM PATH?

\( \exists \) does not satisfy assignment?
SAT: \( \exists F \; / \; \exists x : F(x) = 1 \)

\( \overline{\text{SAT}} : \exists F \; / \; \forall x : F(x) = 0 \)

\( \Sigma_2 \text{SAT} : \exists F \; / \; \exists x \forall y : F(x, y) = 1 \)

\( \Pi_2 \text{SAT} : \exists F \; / \; \forall x \exists y : F(x, y) = 0 \)

\( \Sigma_i \text{SAT} : \exists F \; / \; \exists x, \forall x_2, \ldots \; F(x_1, \ldots) = 1 \)

\( \Pi_i \text{SAT} : \exists F \; / \; \forall x_1, \ldots \; F(\ldots) = 0 \)

Alternating Turing Machines that can at each node of computation accept if any or all paths emanating from the node accepts or if all paths accept.

\( \mathcal{P} \cup \sum_i \cup \Pi_i \text{TIME}(n^k) = \cup_i \Pi_i \text{TIME}(n^k) \)

\( \mathcal{P} \cup \mathcal{P} \subseteq \mathcal{PSPACE} \).
How hard are problems in \( \text{PSPACE} \)? We don't know, but we can define the hardest problems.

A language \( L \) is said to be \( \text{PSPACE-complete} \) if

(a) \( L \in \text{PSPACE} \)

(b) \( \forall B \in \text{PSPACE} \)

\( \exists \) polynomial-time reduction \( B \rightarrow L \)

i.e., using \( L \) as an oracle/procedure and polynomial additional time, \( B \) can be solved in \( \text{PSPACE} \)

"\( L \) is at least as hard as any problem in \( \text{PSPACE} \)."

(If only (b) holds, \( L \) is \( \text{PSPACE-hard} \)).

Do there exist complete languages for \( \text{PSPACE} \)?

TQBF: True Quantified Boolean Formula

\[
F = \forall x_1 x_2 \exists x_3 \forall x_4 \ldots P(x_1, x_2, \ldots, x_n)
\]

TQBF = \( \exists F : F \) is true
TQBF is PSPACE-complete.

**Proof:**

**TQBF ∈ PSPACE**

For this we just give an ATM that matches the quantifiers of a given formula and the depth of the tree is the # variables.

Any problem $B ∈ PSPACE$ has a reduction $B → L$.

Since $B ∈ PSPACE$, ITM $M$ that decides $B$.

Examine the computation tableau of $M$ on input $x$.

We can write a boolean formula $\phi_x$ to check that the computation is valid, the start is valid and the end state is ACCEPT. We $B \iff \phi_x$ is true.

But $\phi_x$ has exponential size!
F_{stat, accept, t} : formula that checks tableau from start to accept using at most t steps.

\[ F_{a, t} = \exists u \left( \phi_{s, u, \left[ \frac{t}{2} \right]} \land \phi_{m, a, \left[ \frac{t}{2} \right]} \right) \]

if \( t = 1 \) or \( 0 \), we can write an explicit formula.

\[ F_{a, 1} \]

Does this work? No! still exponential size.

We can use universal quantifiers.

\[ F_{a, t} = \forall x, y, z \exists u \left( x, y \in E(x, u), (u, a)^y \right) \quad F_{x, y, \left[ \frac{t}{2} \right]} \]

\( \forall x, y, z \exists u \left( x, y \in E(x, u), (u, a)^y \right) \quad F_{x, y, \left[ \frac{t}{2} \right]} \]

\( \forall x, y, z \exists u \left( x, y \in E(x, u), (u, a)^y \right) \quad F_{x, y, \left[ \frac{t}{2} \right]} \]

Every boolean \( F \rightarrow \text{AND/OR/NOT}. \) Now size \( (F) = O(n^{2k}) \).