3. P versus NP

Possible outcomes of the $P \overset{?}{=} NP$ challenge are:

(1) $P = NP$ - find a polynomial algorithm for SAT!

(2) $P \neq NP$ - prove a superpolynomial lower bound for SAT!

(3) $P \overset{?}{=} NP$ is independent of (certain systems of) set theory

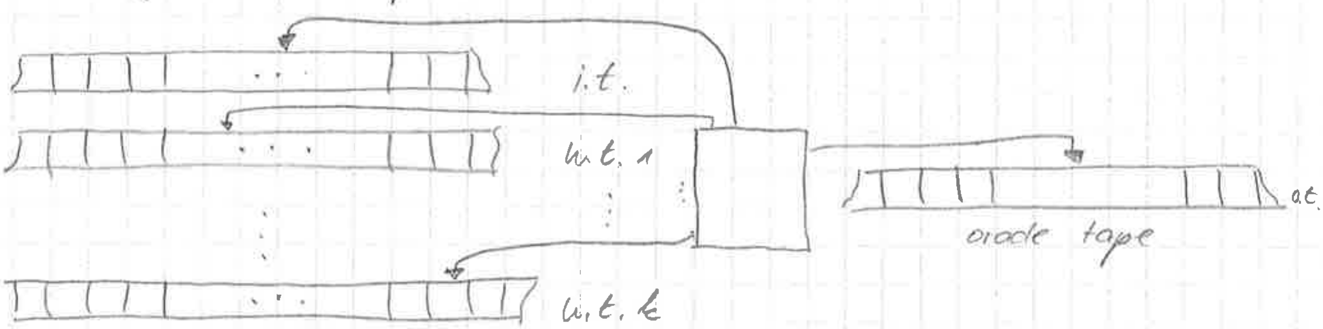
Most complexity theorists believe that $P \neq NP$.

Why is it hard to prove $P \neq NP$?

- counting: combinatorially involved, e.g., only $4n$ lower bound for SAT
- diagonalization: relativizable results

3.1 Oracle Turing machines

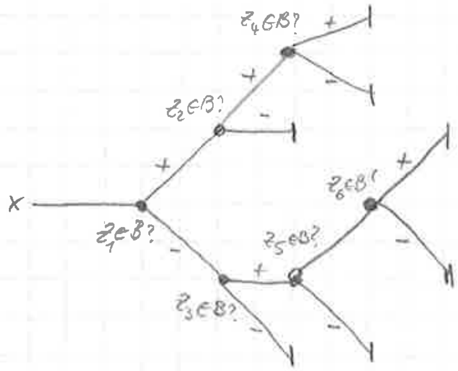
An oracle TM is a TM equipped with an additional (one-way) oracle tape:



Additional states:

- $s_?$ oracle query state (i.e., M asks z to the oracle)
- s_+ positive answer (returning) state (oracle tape is cleared)
- s_- negative answer (returning) state (oracle tape is cleared)

Work of a DOTM on input x for a general oracle B:



decision tree

<u>Example:</u> (Maximum knapsack)

Determine maximum value given total weight B:

$$\max\{\sum_{i\in I} a_i \mid I \subseteq \{1,\dots,m\},\ \sum_{i\in I} b_i \le B\}$$

Use binary search!

For complexity measures, the oracle is neglected.

<u>Relativized complexity classes</u> (relative to an oracle B):

- $DSPACE^B(s)$, $NSPACE^B(s)$, $DTIME^B(\mathrm{Pol}\, t)$, $NTIME^B(\mathrm{Pol}\, t)$
- L^B, NL^B, P^B, NP^B, $PSPACE^B$
- $XSPACE^{\mathcal{K}}(s) =_{df} \bigcup_{B\in\mathcal{K}} XSPACE^B(s)$, $XTIME^{\mathcal{K}}(t) =_{df} \bigcup_{B\in\mathcal{K}} XTIME^B(t)$

All theorems relating complexity classes hold relative to oracle B:

- $DSPACE^B(s) \subseteq NSPACE^B(s) \subseteq DTIME^B(2^{O(s)})$
- $DTIME^B(\mathrm{Pol}\, t) \subseteq NTIME^B(\mathrm{Pol}\, t) \subseteq DSPACE^B(\mathrm{Pol}\, t)$
- $NSPACE^B(s) \subseteq DSPACE^B(s^2)$
- $coNSPACE^B(s) = NSPACE^B(s)$

($s(n) \ge \log n$, $t(n) \ge n$ space-constructible)

Hierarchy theorems also hold relative to any oracle.

We say that these theorems relativize

($\mathcal{K}_1 \subseteq \mathcal{K}_2$ relativizes $\Leftrightarrow_{df}$ $\mathcal{K}_1^B \subseteq \mathcal{K}_2^B$ for all oracles B)

That is, diagonalization is a relativizable proof technique.

However, $P \stackrel{?}{=} NP$ cannot be solved using a rel. proof technique

3.2 P = NP relative to some oracle

Theorem 1.

There is an oracle B such that $P^B = NP^B$.

Proof: Consider any set B $\leq_m^{log}$-complete for PSPACE.
Then, we have $PSPACE \subseteq P^B \subseteq NP^B$.
It remains to show $NP^B \subseteq PSPACE$: Let $A \in NP^B$ via NPOTM $M^{(\cdot)}$, polynomial p. Consider the same machine M' as in Thm 1.26 with an additional oracle tape (used as an input tape) and an additional working. M' iterates over all comp. paths of $M^{(\cdot)}$, simulates $M^{(\cdot)}$ on x, whenever $M^{(\cdot)}(x)$ asks a query z, M' simulates the 2-T-TM for B on z on the addl. working tape. So, M' accepts x iff $x \in A$ and runs space $2 \cdot p(|x|) + q(p(|x|))$. Hence, $L \in PSPACE$. ∎

3.3 $P \neq NP$ relative to some oracle

Theorem 2.

There is an oracle B such that $P^B \neq NP^B$.

Proof: Define, for any set B, the language

$$L^B =_{def} \{ 0^n \mid (\exists x)[\,|x|=n \wedge x \in B\,] \}$$

Clearly, $L^B \in NP^B$ (guess an x and check if $x \in B$)

We have to show that $L^B \notin P^B$ for an appropriate B:

Let $M_1^{(\cdot)}, M_2^{(\cdot)}, M_3^{(\cdot)}, \ldots$ be an enumeration of all POTM, i.e., $M_i^{(\cdot)}$ runs in time p_i for all oracles.

We construct B in stages B_i, $i \in \mathbb{N}$, i.e., $B_i \subseteq B_{i+1}$ and $B = \bigcup_{i=0}^{\infty} B_i$. In each stage i, we guarantee that there exists an x_i s.t.

$$x_i \in L^B \iff M_i^B \text{ does not accept } x_i$$

Stage 0: Set $n_0 =_{def} 0$, $B_0 =_{def} \emptyset$.

Stage i: We assume that there already exists n_{i-1} and $B_{i-1} \subseteq \{ x \in \{0,1\}^* \mid |x| \leq n_{i-1} \}$. Choose least integer n satisfying:

(i) $2^n > p_i(n)$ (where p_i polynomial bounding $M_i^{(\cdot)}$)

(ii) $n > 2^{n_{i-1}}$

Set $n_i = n$ and $x_i = 0^{n_i}$. Simulate $M_i^{B_{i-1}}$ on x_i and consider two cases:

(i) If $M_i^{B_{i-1}}$ accepts x_i then $B_i =_{def} B_{i-1}$

(i.e., $B_i \cap \{ y \in \{0,1\}^* \mid |y| = n_i \} = \emptyset$)

(ii) If $M_i^{B_{i-1}}$ rejects x_i then find y of length $|y| = n_i$ not queried during comp. $M_i^{B_{i-1}}$ on x_i and set $B_i =_{def} B_{i-1} \cup \{y\}$.
(note that y ex. since $M_i^{B_{i-1}}(x_i)$ can only ask $p_i(n_i) < 2^{n_i}$ queries)

It follows that: M_i^B accepts x_i $\iff$ $M_i^{B_{i-1}}$ accepts x_i
(since $n_{i+1} > 2^{n_i} > p_i(n_i)$).

We obtain for all $i \in \mathbb{N}_+$:

$$x_i \in L^B \iff M_i^B \text{ rejects } x_i. \qquad \blacksquare$$

Remarks:

1. $P \neq NP$ relative to a random oracle (i.e., with probability 1)
2. $IP^B \neq PSPACE^B$ for some oracle, but $IP = PSPACE$.