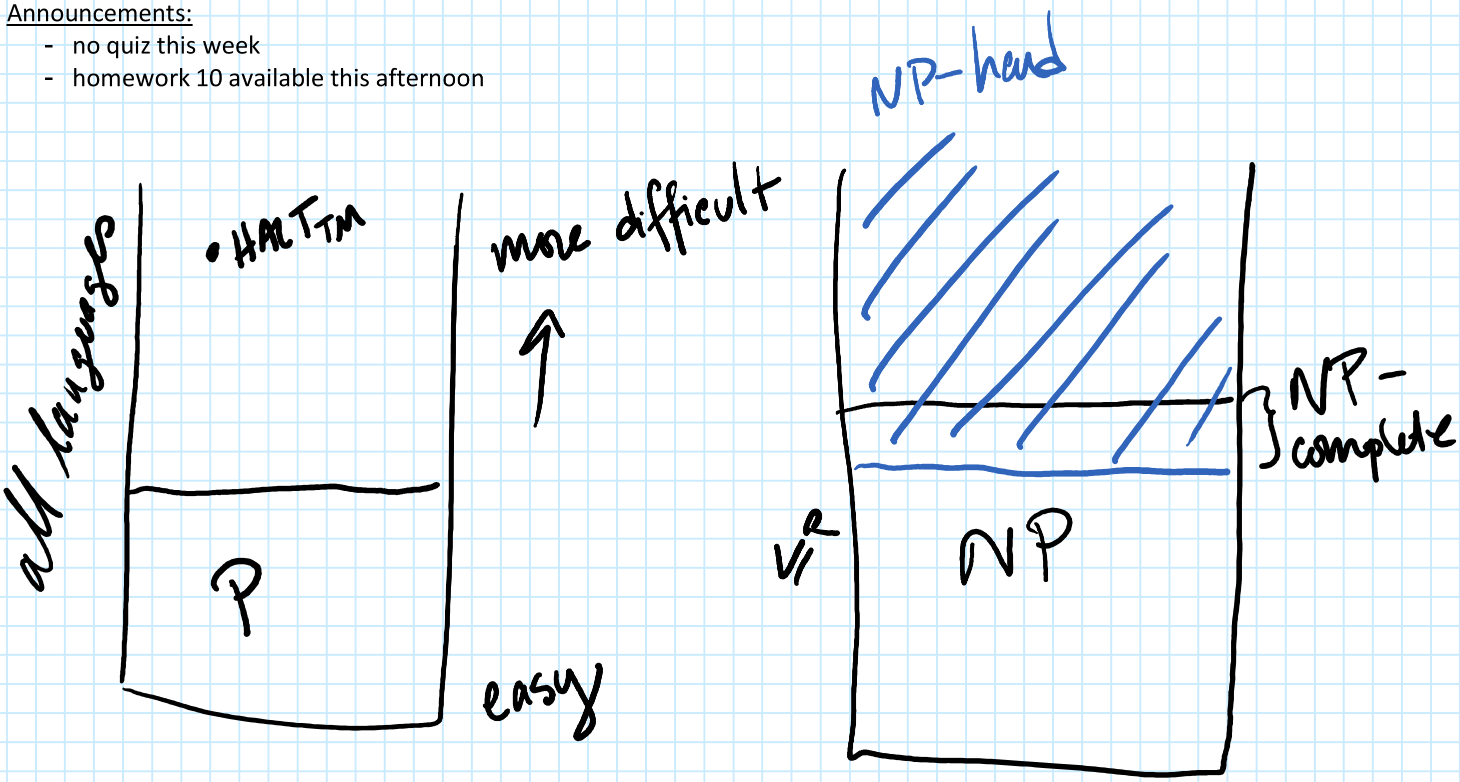
W13 L1 NP-completeness and Cook-Levin Theorem

Monday, April 27, 2020 9:06

<u>Announcements:</u>

- no quiz this week
- homework 10 available this afternoon



We want to combine diagrams with P & NP.

$A \le_p B$

How do you prove a new problem B is NP-complete?

<u>Need to show</u> that $A \le_p B$ for every $A \in NP$.

This seems challenging BUT

<u>Thm</u>: If $C \le_p D$ and $D \le_p E$ then $C \le_p E$.

($\le_p$ are transitive)

<u>Alternately</u>, need to show: $A \le_p B$ for some NP-complete A.

(Of course need $B \in NP$ as well.)

(Of course need $B \in NP$ as well.)

We need one NP-complete problem to get started.

Definitions:

- a **variable** is something like x, y, z (these are logical variables, so they can take on the values true/false)
- a **literal** is a variable or its negation, for example $x, \overline{x}$
- a **disjunction** is an "or" and we write it $\vee$, for example "x or y" is written $x \vee y$
- a **conjunction** is an "and" and we write it $\wedge$, for example "x and y" is written $x \wedge y$
- a **clause** is a disjunction of several literals, for example $(x \vee y \vee z)$, and $(x \vee \overline{x} \vee y \vee w)$ and (y)
- a formula is in **conjunctive normal form** if it is the conjunction of some clauses for example $(x \vee y) \wedge (z \vee \overline{x}) \wedge (w \vee y \vee \overline{z}) \wedge (y)$
- a **satisfying assignment** is a way to give each variable a Boolean value to make the overall expression evaluate to TRUE

Define $SAT = \{ \langle \varphi \rangle \mid \varphi$ is a conjunctive normal form formula and also φ has a satisfying assignment$\}$.

For example $(x) \wedge (\overline{x})$ is not in SAT, but $(x \vee y) \wedge (\overline{z})$ is in SAT.

Theorem: (Cook and Levin) SAT is NP-complete.
Proof:
Need to show that $SAT \in NP$ and also that for any $A \in NP, A \leq_p SAT$.

$SAT \in NP$: The certificate is the truth assignment, and we just check that the input is a properly-formatted formulae φ in conjunctive normal form and that the given truth assignment evaluates to TRUE.

Take any language $A \in NP$. We need to come up with a polynomial-time reduction $A \leq_p SAT$.

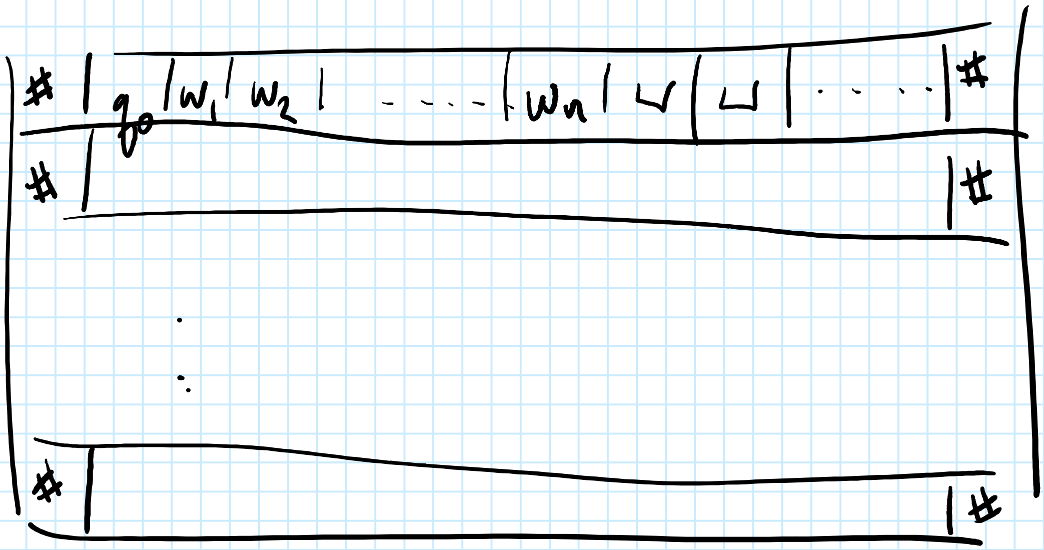
From the assumption that $A \in NP$, we know there is some nondeterministic polynomial-time TM N which decides A. Let's say that N runs in time n^k for some $k \in \mathbb{N}$. We know that on input string w, some branch of N's computation will accept string w.

On that branch, we will go through configurations of N:
first one is the starting configuration: $q_0 w$
next one will be some configuration we get to by following a transition of N based on the last configuration
next one will be some configuration we get to by following a transition of N based on the last configuration
next one will be some configuration we get to by following a transition of N based on the last configuration
next one will be some configuration we get to by following a transition of N based on the last configuration
...
next one is an accepting configuration where N is in state q_{accept}.

at most n^k

longest configuration has length at most n^k

Let's think about this accepting computation as a table of $n^k \times n^k$ many squares:



Now we know that $w \in A$ if and only if there is some table like the above, which has the following properties:

- it is $n^k \times n^k$
- the first row is $\#q_0w$ (blanks) #
- the i^{th} row is one step of computation after the $i-1^{th}$ row
- there is a row where we reached q_{accept}

Our goal is now to write a formula φ which is satisfiable if and only if this table can exist.
(Want that $w \in A$ if and only if $\varphi \in SAT$.)