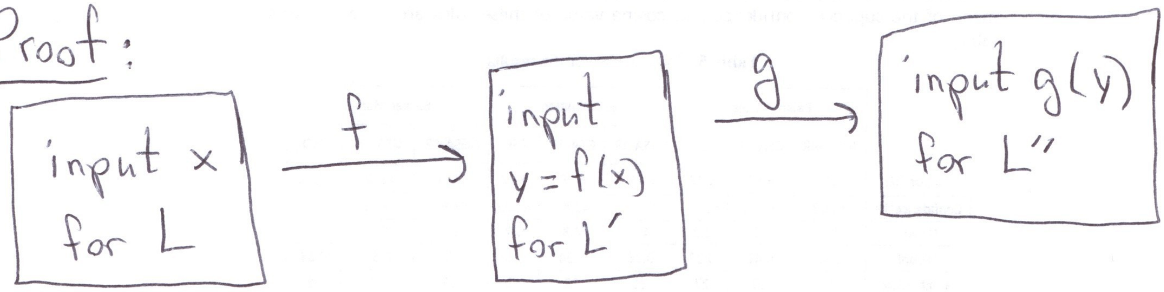
Theorem: The relation $\leq_P$ is transitive:

$$\left.\begin{array}{l} L \leq_P L' \\ L' \leq_P L'' \end{array}\right\} \Rightarrow L \leq_P L''.$$

Proof:



$$x \in L \iff y = f(x) \in L' \iff g(y) \in L''$$

$$\therefore\ x \in L \iff g(f(x)) \in L''.$$

Reduction from L to L'' is given by the function $g \circ f$.

Given x, $(g \circ f)(x) = g(f(x))$ can be computed in time that is polynomial in the length of x. (Why?)

$\square$

Definition:

① Language L is NP-hard if

for every L' in NP : $L' \leq_P L$.

② Language L is NP-complete if

* $L \in NP$ and
* for every L' in NP : $L' \leq_P L$.

In English: L is NP-complete means

* L is in NP,
* L is at least as difficult as every problem in NP.

$\therefore$ L belongs to the most difficult problems in NP.

[this is what we wanted on page 182]

Theorem: Assume L is NP-complete. Then:

$$L \in \mathcal{P} \iff \mathcal{P} = N\mathcal{P}.$$

Proof:

informally:

if $L \in \mathcal{P}$: L is easy.

L is NP-complete: L belongs to the most difficult problems in NP.

$\therefore$ the most difficult problem in NP is easy

$\therefore$ all problems in NP are easy

$\therefore P = NP$.

formally:

$\Leftarrow$ Assume $\mathcal{P} = N\mathcal{P}$.

Since L is NP-complete: $L \in NP$.

$\therefore L \in \mathcal{P}$.

$\Rightarrow$ Assume $L \in \mathcal{P}$.

We have to show that $P = NP$.

We know that $P \subseteq NP$.

To show that $NP \subseteq P$:

Let $L' \in NP$.

Since L is NP-complete: $L' \leq_P L$.

Since $L \in P$: $L' \in P$ (see page 184).

$\square$

<u>Theorem</u>:

$$\left.\begin{array}{l} L \ \text{ NP-complete} \\ L \leq_P L' \\ L' \in NP \end{array}\right\} \Rightarrow L' \ \text{ NP-complete.}$$

<u>Proof</u>:

<u>informally</u>:

$$\left.\begin{array}{l} L \text{ is at least as} \\ \quad \text{difficult as every} \\ \quad \text{problem in } NP \\ \text{and} \\ L' \text{ at least as difficult} \\ \quad \text{as } L \end{array}\right\} \therefore \begin{array}{l} L' \text{ is at least as} \\ \text{difficult as every} \\ \text{problem in } NP. \end{array}$$

<u>formally</u>: To show that L' is NP-complete, we have to show:

* $L' \in NP$: this is given.
* for each $L'' \in NP$: $L'' \leq_P L'$.

  why is this true:

  * since L is NP-complete: $L'' \leq_P L$.
  * we are given: $L \leq_P L'$.
  * by transitivity (page 203): $L'' \leq_P L'$.

$\square$

$\boxed{\text{How to use this}}$

How to use this: To show that L' is NP-complete:

1. show that $L' \in NP$.
2. look for a problem L that is "similar" to L' and that is <u>known to be</u> NP-complete.
3. show that $L \leq_P L'$.

In order to apply this, we need a first NP-complete problem:

We need <u>one</u> language L in NP such that

$\text{HAMCYCLE} \le_P L$,

$\text{TSP} \le_P L$,

$\text{SUBSETSUM} \le_P L$,

$\text{CLIQUE} \le_P L$,

$\text{INDEP-SET} \le_P L$,

$\text{VERTEX-COVER} \le_P L$,

$\text{3SAT} \le_P L$,

$\text{3COLOR} \le_P L$,

$\vdots$

for <u>every</u> L' in NP: $L' \le_P L$.

Not obvious that NP-complete problems exist!

1971: Stephen Cook proved that SAT is
NP-complete.

independently in Russia:

1972: Leonid Levin proved that a certain tiling
problem is NP-complete.