Reducing Problems

We have seen that polynomial time reduction between problems is a very useful concept for studying relative complexity of problems. It allowed us to distinguish a class of problems, **NP**, which includes many important problems and is viewed as the class of hard problems

We are going to do the same for space complexity classes: **NL** and **PSPACE**

There is a problem:

Polynomial time reduction is too powerful

Log-Space Reduction

A transducer is a 3-tape Turing Machine such that

- the first tape is an input tape, it is never overwritten
- the second tape is a working tape
- the third tape is an output tape, no instruction of the transition function uses the content of this tape

The space complexity of such a machine is the number of cells on the working tape visited during a computation

A function $f : \Sigma^* \to \Sigma^*$ is said to be log-space computable if there is a transducer computing f in $O(\log n)$

Definition A language A is log-space reducible to a language B, denoted $A \leq_L B$, if a log-space computable function f exists such that for all $x \in \Sigma^*$

$$x \in A \Leftrightarrow f(x) \in B$$

Note that a function computable in log-space is computable in polynomial time, so

$$A \leq_L B \Rightarrow A \leq B$$

Completeness

Definition

A language L is said to be **NL**-complete if **L** $\in$ **NL** and, for any A $\in$ **NL**,

$$A \leq_L L$$

Definition

A language L is said to be **P**-complete if L $\in$ **P** and, for any A $\in$ **P**,

$$A \leq_L L$$