Reductions*

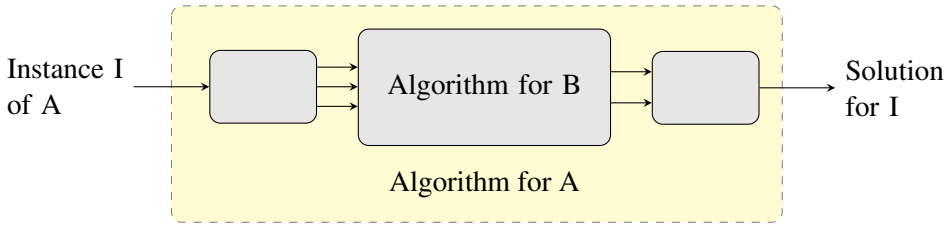
Give me a lever long enough and a fulcrum on which to place it, and I shall move the world. - Archimedes

Given a new problem, we ask ourselves whether we can solve it easily by transforming it to a problem that we already know how to solve. We now give an informal introduction to the notions behind this process.

We say that a problem A *reduces to* a problem B, and write $A \leq B$, if we can use an algorithm to solve B to find an algorithm that solves A, efficiently. Two problems are *equivalent* if they reduce to each other.

"Efficiently" here can mean different things, which leads to different notions of reduction. We adopt here the meaning: in a total amount of time that is, in the worst case, no more than a constant times the worst-case running time of the algorithm that solves B.

Reduction of A to B can be pictured as follows:



Task 1 Show that computing a square of a number reduces to multiplication, and that multiplication of two numbers reduces to computing a square of a number (given that one can subtract and divide by 2 in at most as many steps as the steps needed for computing squares).

Task 2 Show that computing the sum of all numbers between 1 and 100 that are not divisible by 3 reduces to computing the sum of the first n numbers for a given n, $n \leq 100$.

Task 3 You want to plug your computer into a projector. Your computer has a VGA

*The material in this document origins in different sources on the web for most of which we could not track the original authors, but we acknowledge them nevertheless!

output port, but the projector does not have a VGA input port. Instead, it has an input port of type X. You have a box of connectors. Each connector has an input port of one type and an output port of one type (which, in principle, could be the same type). You want to determine whether it is possible to chain some number of connectors together so that you can plug your computer into the projector. To make the problem concrete, consider you are given the following box of connectors, where (I, O) denotes that you have a connector with input I and output O:

(HDMI, USB)	(DB13W3, X)	(VGA, HDMI)
(VGA, DisplayPort)	(DVI, DB13W3)	(DVI, DisplayPort)
(DB13W3, CATV)	(S-Video, DVI)	(USB, S-Video)
(DisplayPort, HDMI)	(Firewire,SDI)	(SDI,X)

Design an algorithm that determines whether it is possible to connect your computer to the projector, by a suitable reduction.

Here is a glimpse of applications of reductions:

(1) positive: If $A \leq B$, then an (efficient) algorithm for solving B gives us an (efficient) algorithm for solving A. Here, "efficient" refers to certain complexity, and enables us to show *upper bounds* for the complexity of A.

(2) negative: If there is no (efficient) algorithm for solving A, then there can be no (efficient) algorithm for solving B. Again, here "efficient" refers to certain complexity and enables us to show *lower bounds*. Having no algorithm for solving A is even more interesting, as it enables us to study the borders of computability and prove problems *undecidable*.

Example for upper bounds: Finding the k-th largest number (known as *the selection problem*) in a list of numbers amounts to sorting the list in descending order and returning the k-th element. As we know sorting algorithms that take at most $n \log(n)$ steps, we see that the selection problem has a complexity of at most $n \log(n)$ (upper bound).

Example for lower bounds:[1] *There is no linear-time algorithm for sorting*; *Sorting reduces to Convex Hull*. Hence, there is no linear algorithm for Convex Hull.

Example for undecidability: *Halting (given a Turing machine M and an input word w, return true iff M halts on input w) is undecidable*; *Halting reduces to emptiness (given a Turing machine M, return true iff M does not accept any word)*; Hence, emptiness is undecidable.

All the best for your further studies !

[1] In the following, we use several claims that we will probably not manage to show in class. They are highlighted in italics, and are just used to make a point.