LECTURE 8 – SOLVABILITY AND UNSOLVABILITY

Overview

- The Halting Problem
- Decision Problems

The Halting Problem

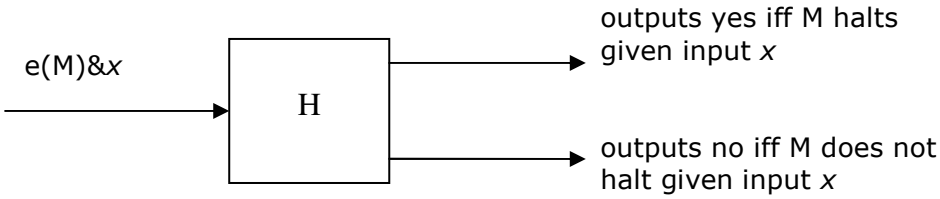
An algorithm can be defined as a procedure that always terminates and gives an answer. Formally, an algorithm can be defined as a Turing machine that always halts (whether succeeds or fails).

A problem is said to be solvable if there is an algorithm that solves the given problem (every instance).

A problem is said to be unsolvable is no algorithm exists that solves the given problem.

The Halting Problem for Turing machines is defined as follows:
Given a TM $M=(Q,\Sigma,\Gamma,\delta,q_0,F)$ and an input string $x \in \Gamma^*$, will M eventually halt?

Then, the Halting problem would be solvable if a Turing machine H that behaves like illustrated below, can be constructed:



In the above, the symbol & is a separator and e(M) is an encoding of M, i.e. e(M) is for example the set of 5-tuples (q,x_1,p,r,R) that describe the Turing machine.

The Halting problem is then:

"Does there exist an effective procedure (computable function) for deciding, for every pair (e(M),x); does M halt for x?"

Theorem. The Halting problem for Turing machine is unsolvable.

Proof. The proof is by contradiction, that is, assume that the Halting problem is solvable and then find a contradiction.

If the Halting problem is solvable, then there must be a Turing machine to decide the Halting problem (Church thesis), that is, the Turing machine H exists.

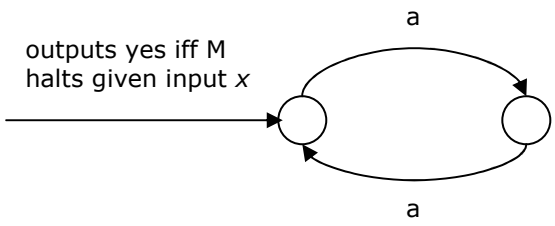
Then, if H can solve the Halting problem for input e(M)&x it should be able to solve the Halting problem for input e(M)&e(M).

Then, it should be possible to construct another Turing machine H' that accepts as input e(M) but behaves like H.

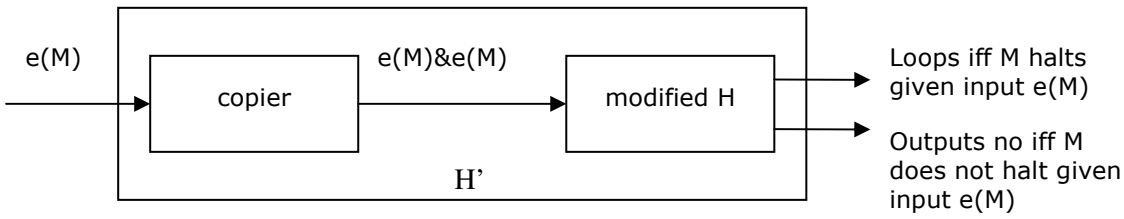
The construction is as follows:

H' takes the input: e(M) and makes a copy to obtain the input: e(M)&e(M)

Then H' executes H with the input e(M)&e(M) but in the case that H outputs yes, H' loops forever. This modification is easily done by adding a couple of states that for a non-terminating right-left loop.



Then, the Turing machine H' is illustrated as follows:



H' is also a Turing machine, then assume that H' receives e(H') as an input. What is the behaviour of H'?

- If H' halts for e(H') then H answers yes and H' loops
- If H' does not halt for e(H') then H answers no and H' halts

Both of the above represent a contradiction, hence H nor H' exist so the Halting problem is effectively unsolvable.

Decision Problems

The Halting problem for Turing machines is not the only unsolvable problem.

There are many problems in computing and mathematics that are unsolvable.

The proof of the unsolvability of many problems relies upon the unsolvability of the Halting problem.

Many problems can also be stated as a corresponding problem with a yes/no solution. These are called decision problems.

Examples of decision problems:

INV(w) – the inverse of string w
Given a strings w and v $\in \Sigma^*$
Does INV(w) = v ?

Palindromes recognition
Given a string w $\in \Sigma^*$
Is w a palindrome ?

Unit addition
Given two integers x and y
Does y = x + 1 ?

Addition of two integers
Given three integers x, y and z
Does z = x + y ?

Halting problem for Turing machines
Given a Turing machine M and an input string x
Does M halt for x as input ?

Travelling salesman problem
Given the set of cities, distances between them, and number B
Is there a tour of all cities with length at most B ?

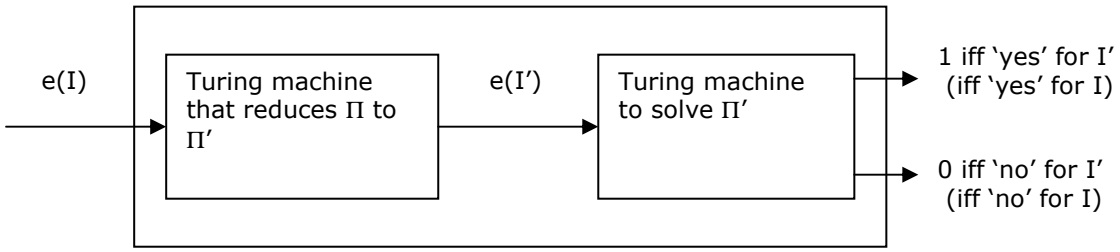
Then, a decision problem Π is solvable if there exists a Turing machine M_Π that always halts successfully and with an output 'yes' or 'no' for every instance of the decision problem.

If no such Turing machine exists then the decision problem Π is unsolvable.

Given two decision problems Π and Π',

We say that Π reduces to Π' if an algorithm to solve Π' can be directly used to solve Π.

That is, Π reduces to Π' if there is a Turing machine that takes as input an encoding of any instance I of Π and produces as output some encoding of an instance I' of Π' such that the answer for I is 'yes' if and only if (iff) the answer for I' is 'yes'.



Theorem. If Π reduces to Π' and Π' is solvable then Π is also solvable.
If Π reduces to Π' and Π is unsolvable then Π' is also unsolvable.

Then, since the Halting problem for Turing machines is unsolvable, if there is a decision problem Π that reduces to the Halting problem, then Π must be also unsolvable.

The proof that a problem Π is unsolvable establishes that: if there was an algorithm for solving problem Π, then there would be an algorithm for the Halting problem, which is absurd (it was already shown by contradiction).

Example. Proof that the Empty word halting problem is unsolvable.

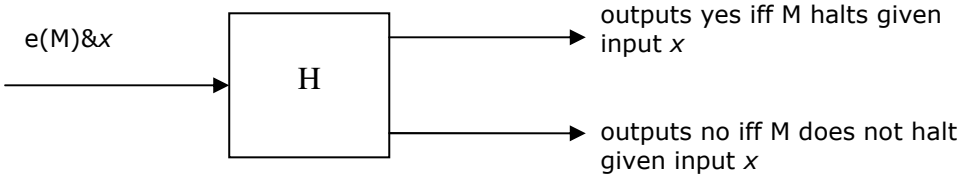
Empty word halting problem:

Given a Turing machine M

Does M halt given the input ε ?

Proof. The technique is to show that the Halting problem (HP) reduces to the Empty word halting problem (εHP)

Assume an instance I for HP, that is, a Turing machine M and the input string x.



To construct an instance I' for εHP, that is, a Turing machine M' and the input string ε, M' first writes x onto its tape to then simulate the actions of M.

Clearly M halts for input x iff M' halts for input ε

Then I gives 'yes' iff I' gives 'yes'.

Then εHP is unsolvable or undecidable.

Reading: (Papadimitriou, chapters 2,3), (Boolos et al., chapter 4), (Lewis&Papadimitriou, chapter 4)