Department of Computer & Information Sciences
Pakistan Institute of Engineering and Applied Sciences

Reducibility

Based on slides by Costas Busch [http://csc.lsu.edu/~busch]

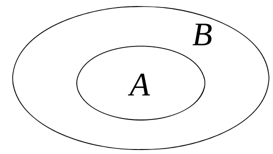
Umar Faiz

http://www.pieas.edu.pk/umarfaiz/cis317

Reducibility

Problem A is reduced to problem B

If we can solve problem B then we can solve problem A



Reducibility

Problem A is reduced to problem B

If B is decidable then A is decidable

If A is undecidable then B is undecidable

Reducibility

Example:

The halting problem is reduced to the state-entry problem

Reducibility

The state-entry problem:

Inputs:

Turing Machine M
State q
String w

Question:

Does M enter state q on input w ?

Reducibility

Theorem:

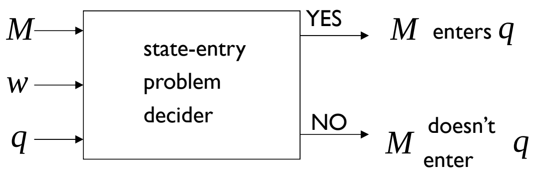
The state-entry problem is undecidable

Proof:

Reduce the halting problem to the state-entry problem

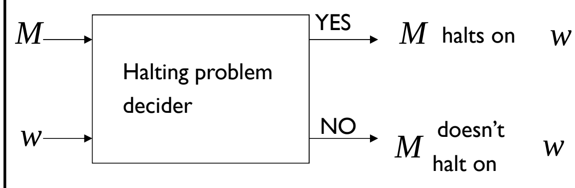
Reducibility

Suppose we have a Decider for the state-entry algorithm



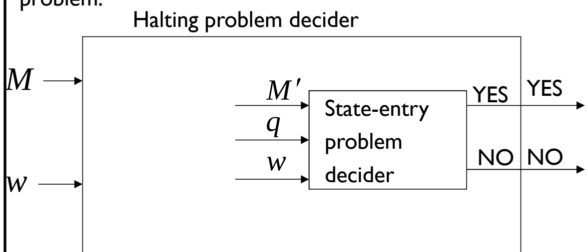
Reducibility

We want to build a decider for the halting problem:



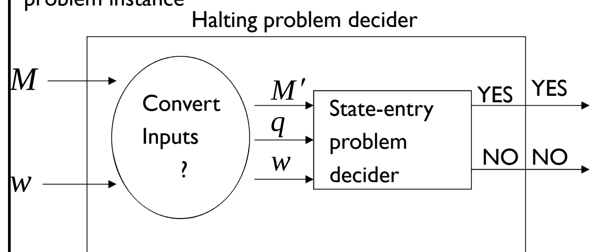
Reducibility

We want to reduce the halting problem to the state-entry problem:



Reducibility

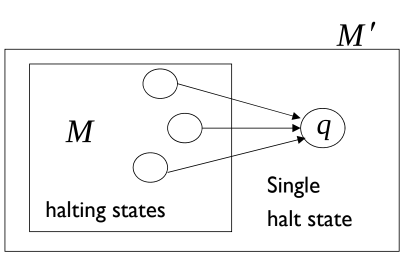
We need to convert one problem instance to the other problem instance



Reducibility

Convert M to M'

Add new state q. From any halting state of M add transitions to q



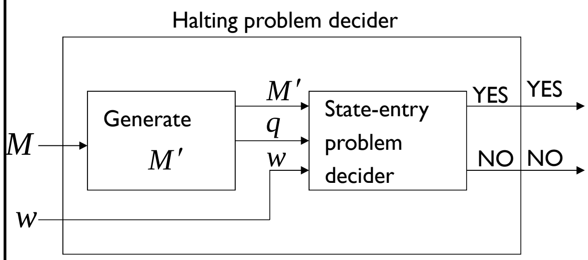
Reducibility

M halts on input w

if and only if

M' halts on state q on input w

Reducibility



Reducibility

We reduced the halting problem to the state-entry problem. Since the halting problem is undecidable, the state-entry problem is undecidable

END OF PROOF

Reducibility

Another example:

The halting problem is reduced to the blank-tape halting problem

Reducibility

The blank-tape halting problem

Input: Turing Machine M

Question:

Does M halt when started with a blank tape?

Reducibility

Theorem:

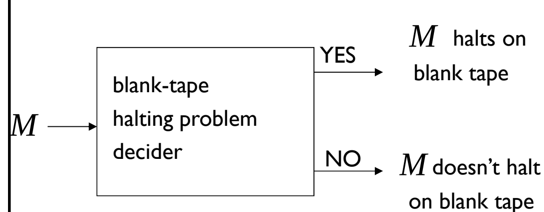
The blank-tape halting problem is undecidable

Proof:

Reduce the halting problem to the blank-tape halting problem

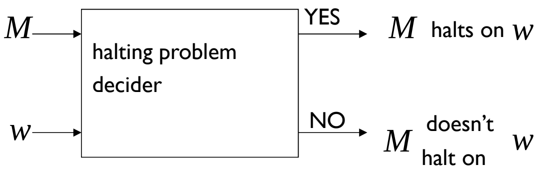
Reducibility

Suppose we have a decider for the blank-tape halting problem:



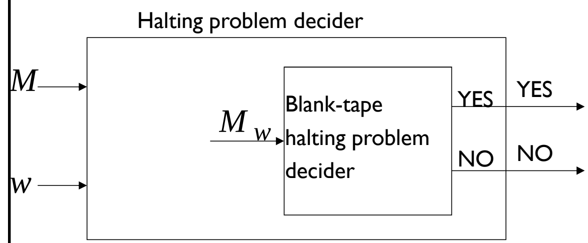
Reducibility

We want to build a decider for the halting problem:



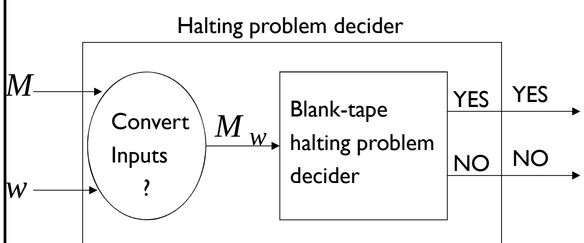
Reducibility

We want to reduce the halting problem to the blank-tape halting problem:



Reducibility

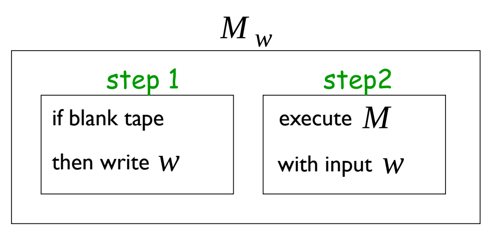
We need to convert one problem instance to the other problem instance



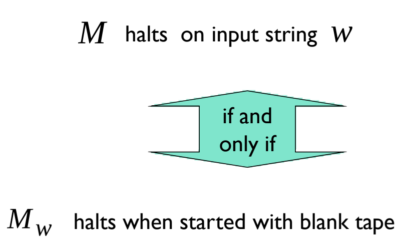
Reducibility

Construct a new machine M_w

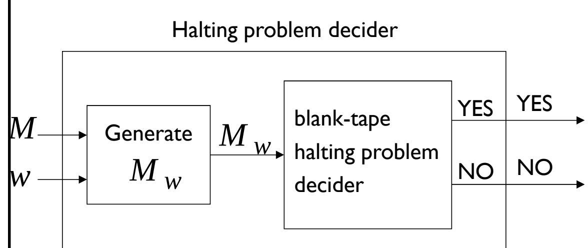
When started on blank tape, writes w, then continues execution like M



Reducibility



Reducibility



Reducibility

We reduced the halting problem to the blank-tape halting problem. Since the halting problem is undecidable, the blank-tape halting problem is undecidable

END OF PROOF

Summary of Undecidable Problems

Halting Problem:

Does machine M halt on input w ?

Membership problem:

Does machine M accept string w ?

Reducibility

Blank-tape halting problem:

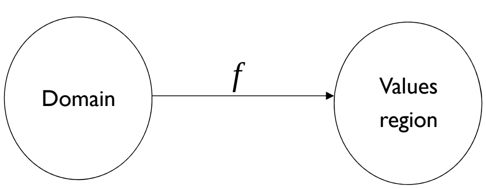
Does machine M halt when starting on blank tape?

State-entry Problem:

Does machine M enter state q on input w?

Uncomputable Functions

Uncomputable Functions



A function is uncomputable if it cannot be computed for all of its domain

An uncomputable function:

$$f(n) = \begin{cases} \text{maximum number of moves until} \\ \text{any Turing machine with } n \text{ states} \\ \text{halts when started with the blank tape} \end{cases}$$

Theorem:

Function $f(n)$ is uncomputable

Proof:

Assume for contradiction that $f(n)$ is computable. Then the blank-tape halting problem is decidable

Decider for blank-tape halting problem:

Input: machine M

1. Count states of M: m

2. Compute $f(m)$

3. Simulate M for $f(m)$ steps starting with empty tape

If M halts then return YES

otherwise return NO

Therefore, the blank-tape halting problem is decidable.
However, the blank-tape halting problem is undecidable

Contradiction!!!

Therefore, function $f(n)$ in uncomputable

END OF PROOF