Oracle machines

We shall not go any further into the nature of this oracle apart from saying that it cannot be a machine.

Alan Turing, 1939

Introduction

- Introduced by Turing 1939 in his PhD thesis
- An oracle machine is a Turing machine with a magical black box
- The box can solve two kinds of problems
  - Decision problems; Is element x in set A?
  - Functional problems; What is f(x)?

Definition

- The oracle machine has the same tape, read head and state machine as a usual Turing machine
- It also has
  - Oracle tape, semi-infinite tape, can have different symbols than work tape
  - Oracle head, moves independent of work head
  - Two special states, ASK and RESPONSE

Functionality

- When the machine is switched to the ASK state:
  - The content of the oracle tape is consided the question
  - The oracle tape is replaced with the answer
  - The machine is switched to the RESPONSE state
- This happens in one step

Alternative definition

- The oracle is defined as a language {0,1}*
- Oracle machine has three extra states, ASK, YES, NO
- The machine asks a questions by writing on the oracle tape and moving to state ASK
- The oracle immediately (in one step) moves to YES or NO depending on if the question is in the language or not

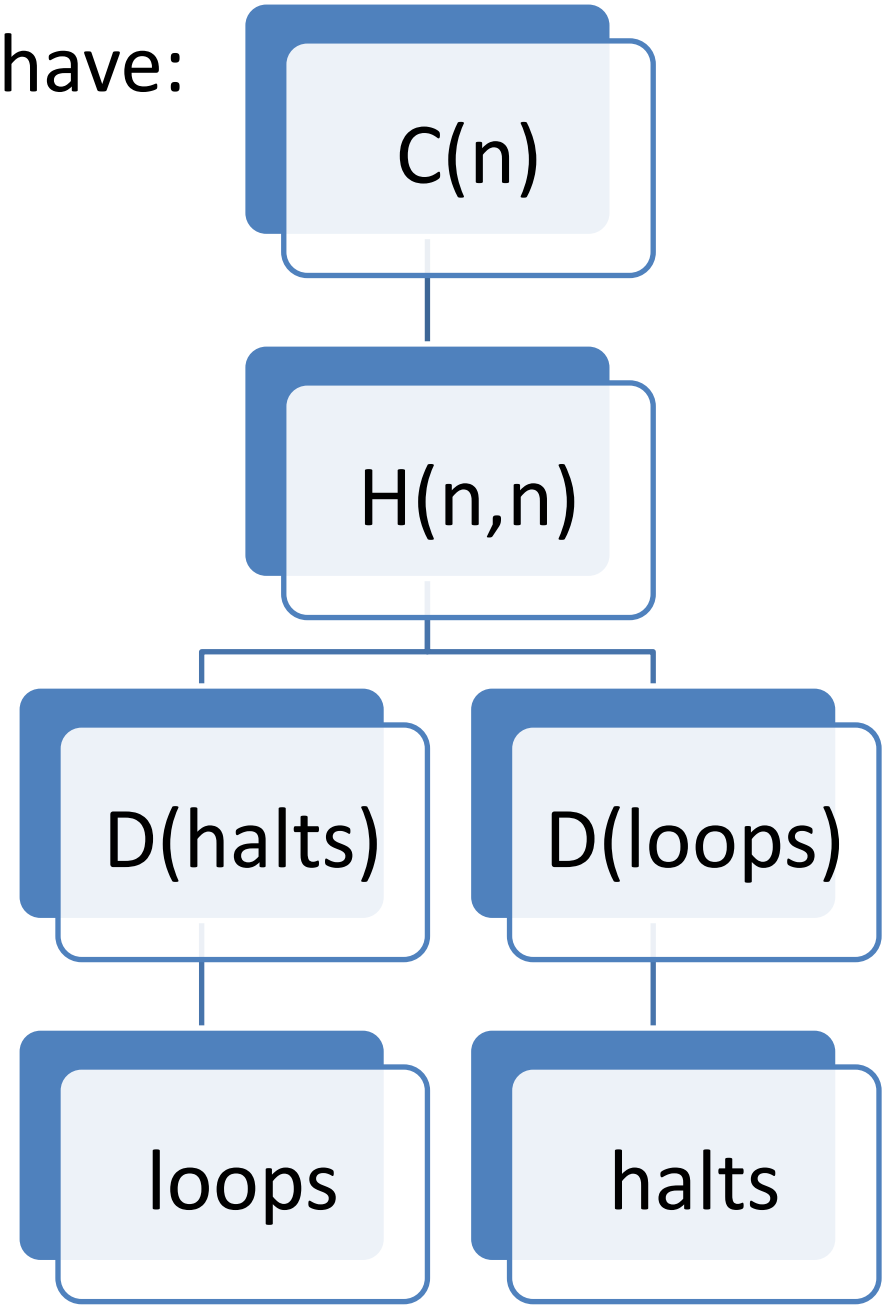
Can we use this?

- Post-Turing thesis, Turing [1939], Post [1944].
- *A set B is effectively reducible to another set A iff B is Turing reducible to A by a Turing oracle machine ($B \leq_T A$)*
- Example
  - A is the blank tape halting problem
  - B is the halting problem

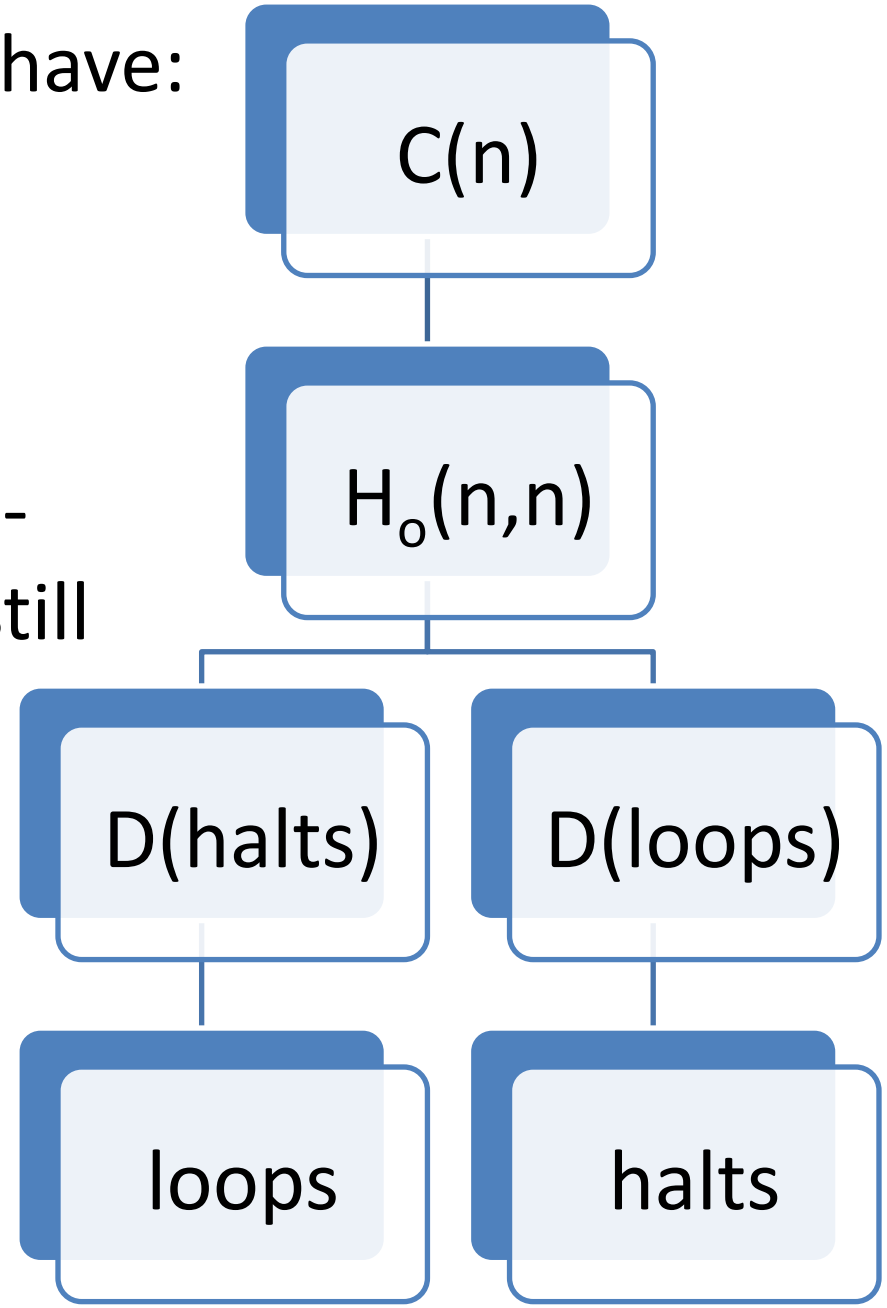
We can use this!

- We can finally solve the halting problem
- We create an oracle which can answer the question h(m) or h(m,n) with yes or no
- Any problems with this? Is there a machine we still can’t decide if it halts or not?

- In the original proof we have:
  - A Copying machine C
  - A Halting machine H
  - A Dithering machine D



- In the original proof we have:
  - A Copying machine C
  - A Halting machine H
  - A Dithering machine D
- We simply change the H-function and the proof still works



No perfect solution

- We cannot solve the halting problem for an oracle machine which has the halting problem as its oracle!
- We can however create a new oracle for this, creating a hiearchy of more and more complex problems.
- This introduces concepts such as *Turing degrees* and *arithmetical hierarchy*

Other uses

- Another useful scenario for oracle machines is in cryptography
  - We prove that we have a function f() which is not solvable in reasonable time (without oracle)
  - We show that if there is an oracle that can break our encryption, then this oracle would also solve f()
  - Thus our encryption cannot be solved in reasonable time

Questions?

Sources

- Hypercomputation: philosophical issues (B. Jack Copeland 2003)
- Turing oracle machines, online computing, and three displacements in computability theory (Robert I. Soare 2009)
- https://en.wikipedia.org/wiki/Oracle_machine
- http://mathoverflow.net/questions/33046/arent-oracle-machines-unsound-concepts
- https://en.wikipedia.org/wiki/Turing_reduction
- http://blog.cryptographyengineering.com/2011/10/what-is-random-oracle-model-and-why.html
- (http://youtu.be/92WHN-pAFCs)