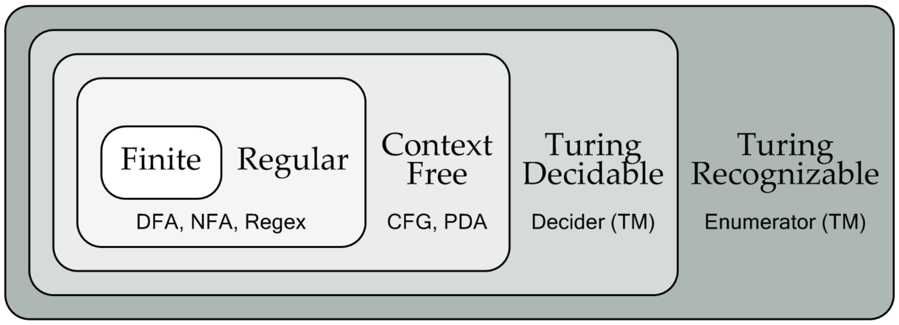
ECS120 Introduction to the Theory of Computation
Fall Quarter 2007

Discussion 8 Notes
Wednesday November 21, 2007

Beyond Turing Machines

Lets revisit the language hierarchy we have covered so far:



Are there languages beyond Turing-recognizable languages? The answer is yes, **some languages are not Turing-recognizable** (corollary 4.18). The intuition behind this proof uses two key observations.

The first key observation is that **the set of all Turing machines is countable**. Each Turing machine has a string encoding, and the set of all strings Σ^* is countable.

Intuition: While there are infinitely many strings, any single string has finite length. Futhermore, there are a finite number of strings of that length. Therefore, this set contains an infinite number of elements of finite length (similar to $\mathcal{N}$).

The second key observation is that **the set of all languages is uncountable**.

Intuition: Languages may be infinite, unlike strings. There are also an infinite number of languages. Therefore, this set contains an infinite number of elements of infinite size (similar to $\mathcal{R}$).

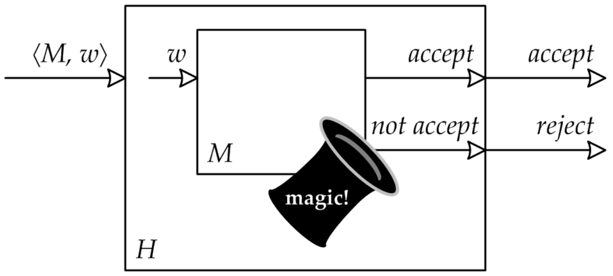
Since we have more languages than Turing machines, there must be some languages that cannot be represented by a Turing machine. (These are often called Turing-unrecognizable languages.)

What implication does this have? This means for some problems there is no way to "effectively compute" them, even with the most powerful computers.

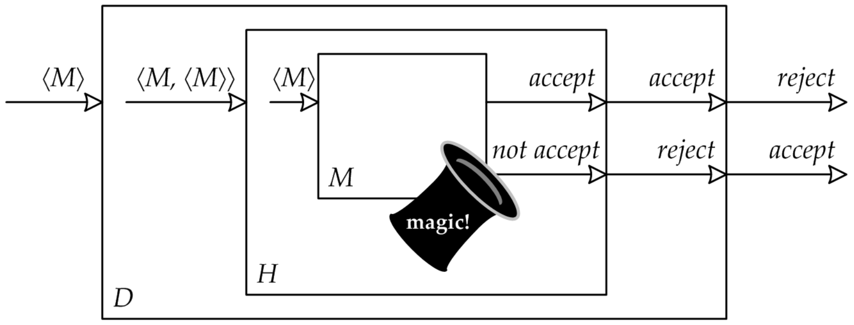
The Acceptance Problem

The language $A_{\mathsf{TM}} = \{ \langle M, w \rangle \mid M \text{ is a } \mathsf{TM} \text{ and } M \text{ accepts } w \}$ is undecidable. It is important to understand this result, as we will reuse this in several reductions.

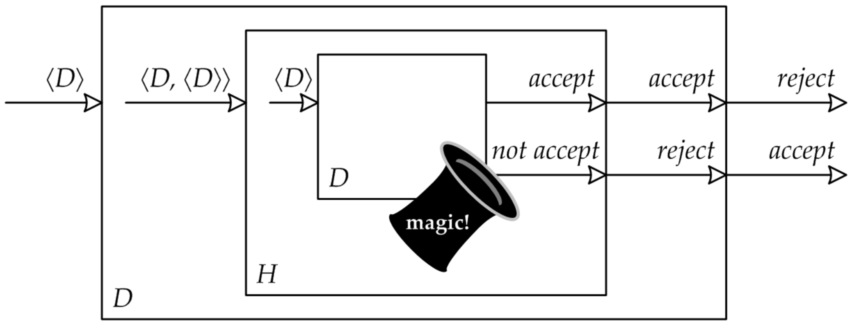
First, we design a TM H that is magically able to tell when M accepts w, even if M loops forever. Basically, we assume H is a decider for A_{TM}:



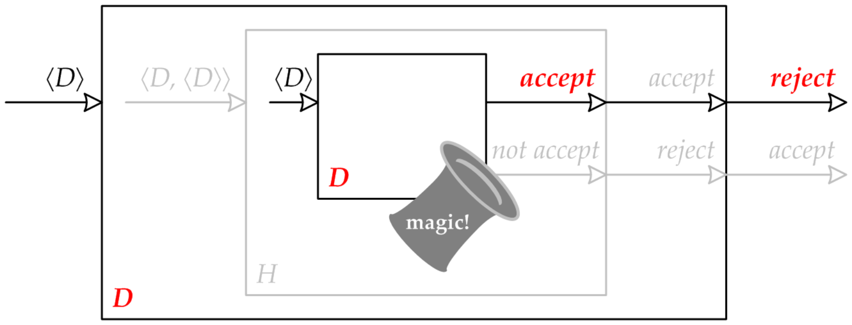
We want to show that if H exists, then we are able to do something we know to be impossible. This is why we build the TM D:



Essentially, D outputs the complement of H. Notice that we may give D its own description:



However, when we do this we get a contradiction (in red). The outer D must reject when the inner D accepts. A Turing machine cannot both accept and reject at the same time on the same input. Therefore, H must not exist.

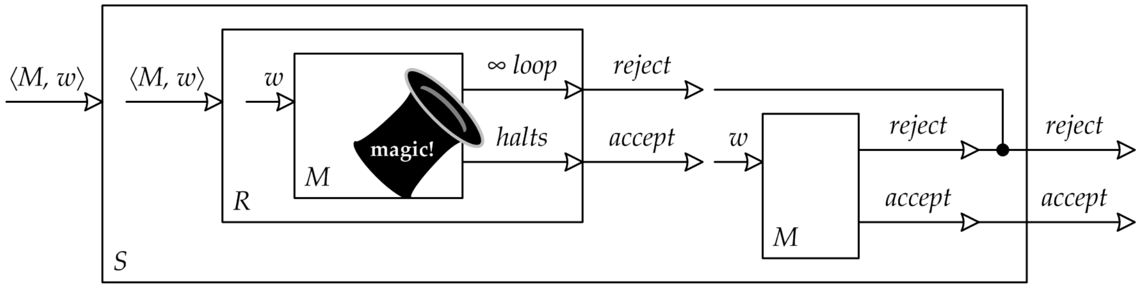


The Halting Problem

The halting problem, $HALT_{\mathsf{TM}} = \{ \langle M, w \rangle \mid M \text{ is a } \mathsf{TM} \text{ and } M \text{ halts on input } w \}$, is undecidable. You should already have some intuition behind why some Turing machines will never halt.

To prove this, we want to show that if a decider exists for $HALT_{\mathsf{TM}}$, then we are able to do something we know to be impossible. We know that A_{TM} is undecidable, so lets try to reuse that knowledge.

Suppose for the sake of contradiction that $HALT_{\mathsf{TM}}$ is decidable. Let R be a decider for $HALT_{\mathsf{TM}}$. We can build a decider S for A_{TM} as follows:



This shows that if R decides $HALT_{\mathsf{TM}}$, then we should be able to build a decider S for A_{TM}. We know this is impossible, so R must not exist.

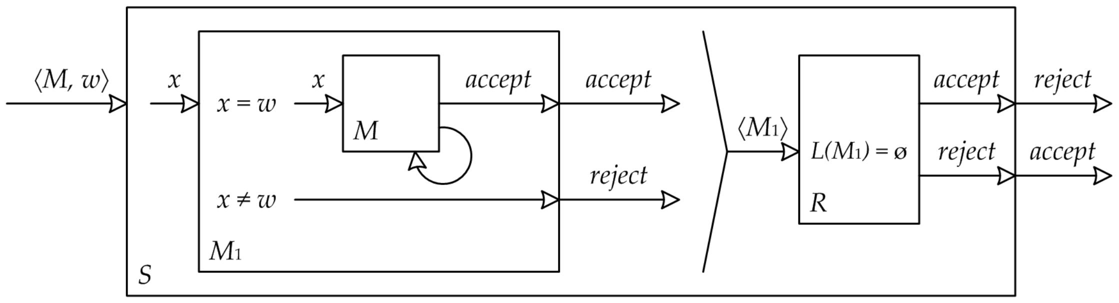
The Emptiness Problem

The emptiness problem, $E_{\mathsf{TM}} = \{ \langle M \rangle \mid M \text{ is a } \mathsf{TM} \text{ and } L(M) = \emptyset \}$, is undecidable.

We are going to follow the same pattern. Suppose for the sake of contradiction that E_{TM} is decidable. Let R be a decider for E_{TM}. The TM R accepts $\langle M \rangle$ as input, and outputs whether $L(M)$ is empty. Is it possible to build a decider S to solve A_{TM} with R?

We know that if $L(M)$ is empty, then it rejects all strings. However, if $L(M)$ is not empty, M could accept or not attempt the string.

To get around this, we introduce a new Turing machine M_1. We can then build a decider S for A_{TM}:



Of course, this results in a contradiction since we know A_{TM} is undecidable. Therefore, no such R exists and E_{TM} must be undecidable.

** Note: These are all in your book. I just tried to give diagrams to help your intuition for what is going on, and how we reduce these problems. Please refer to the book for the actual TM descriptions and proofs.