# #1 Decidability—Review

Recall the following definitions.

Let $A \subseteq \Sigma^*$ be a language.

- We say that $M$ recognizes or accepts $A$ if and only if $M$ accepts every string in $A$ but nothing else. In this case, we say that $A$ is Turing-Recognizable.
- We say that $M$ decides $A$ if and only if $M$ accepts every string in $A$ and rejects every string in $\Sigma^* \setminus A$. Equivalently, $M$ decides $A$ iff $M$ recognizes $A$ and halts on all inputs. In this case, we say that $A$ is decidable.

A language that either accepts or halts (but never loops) on all inputs is called a decider.

# #2 A very simple question of decidability

Consider the following language $E$.

$$E = \{\langle n \rangle \mid n \in \mathbb{N} \text{ is even } \}$$

How might we decide this?
Answer:

1. Inspect the rightmost bit. If 0, accept. Otherwise, reject.

Why is this decidable and not just recognizable?

# #3 Decidability of Finite Automata

We begin our exploration of decidability by considering the decidability of previous models of computation we have seen in this course. We do this because it leads smoothly into discussing undecidability. However, do not be fooled—most questions that concern computer scientists are decidable. Any time you write an algorithm, you are really trying to decide a question. For example, when I write Dikjstra's algorithm, I am giving a proof that any two points in a connected graph have a shortest path. We consider problems like these later on in our study of complexity.

For now, let's stick to deciding language theoretic properties. The simplest is the question "Does a given DFA halt on a given input?"

The language $A_{DFA}$ is decidable.

$$A_{DFA} = \{\langle D, w\rangle \mid D \text{ accepts input string w}\}$$

The sketch is simple. Let $M$ decide $A_{DFA}$ as follows:

1. $M$ has input $\langle B, w\rangle$ with $B$ a DFA and $w$ a string.
2. Simulate $B$ on input $w$.
3. If the simulation ends in an accept state, *accept*. Otherwise reject.

N.b. this works because DFAs will certainly halt after processing finite input!

Other regular models of computation are decidable.

The following are also decidable.

1. $A_{NFA} = \{\langle B, w\rangle \mid B \text{ is an NFA that accepts input string } w\}$.
2. $A_{REX} = \{\langle B, w\rangle \mid R \text{ is a regular expression that generates string } w\}$.

Let's think in class about how we can use the equivalence of regular models of computation to show these results.

Answer: For $A_{NFA}$, convert input to a DFA and use $A_{DFA}$; for $A_{REX}$, convert input to an NFA and use $A_{NFA}$. We call this reducibility.)

## #4 Deciding emptiness

Consider:

$$E_{DFA} = \{\langle D\rangle \mid D \text{ is a DFA and } L(D) = \emptyset\}$$

In other words, $E_{DFA}$ tells us if $D$ accepts the empty language. Do we think this is decidable?
Yes. Define a TM $T$ that decides $E_{DFA}$ as follows.

1. $T$ takes input $\langle D\rangle$ where $D$ is a DFA.
2. Mark the start state of $D$.
3. Repeat until no new states get marked:
   (a) Mark any unmarked state reachable in one step from a marked state.
4. If no accept state is marked, accept; otherwise, reject.

In effect, we are simply determining if any accept state is reachable from the start state.

To show correctness:

1. If an accept state $q_F$ is marked, then there exists a trace of states from $q_0$ to $q_F$ along some string $w_1...w_n$ of transitions. Such a string is in $L(D)$.
2. If no accept state is marked, $L(D)$ must be empty, as no string from the start state may reach an accept state.

Finally, $T$ must halt in either an accept or reject state, as it has only a finite set of states to mark.

Note how our description of $T$ demonstrates that, for finite automata, we can also "inspect" the properties of the encoded machine to answer questions about it. In other words, we can do more than

simply simulate inputs.

Consider now the task of determining equivalence of two arbitrary DFAs.

$$EQ_{DFA} = \{\langle A, B\rangle \mid A \text{ and } B \text{ are DFAS and } L(A) = L(B)\}$$

Do we think this is decidable? Your lab today explores a proof.