Week 8

Decidability and Reductions

Anakin

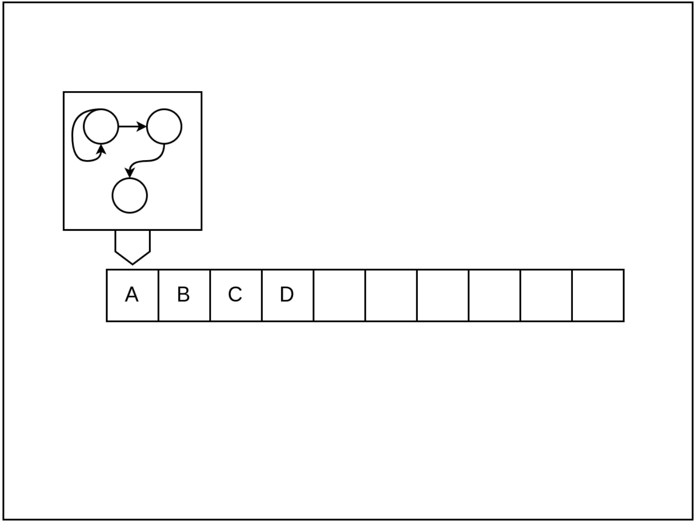
Outline

Σ

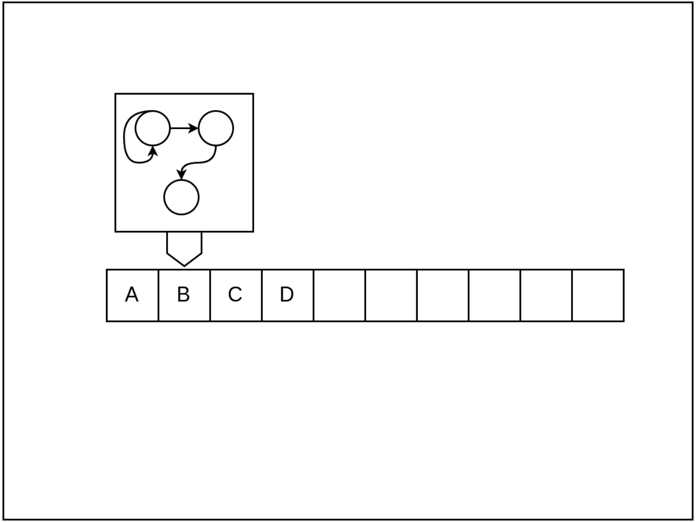
Section 1

Decidability & Recognizability

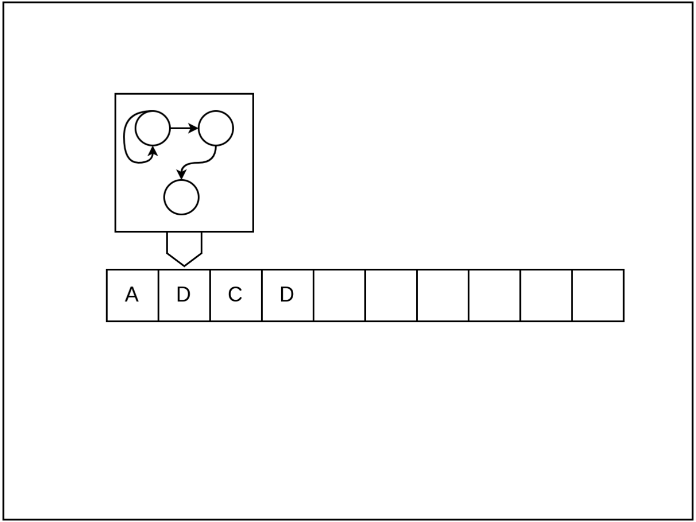
Turing Machine



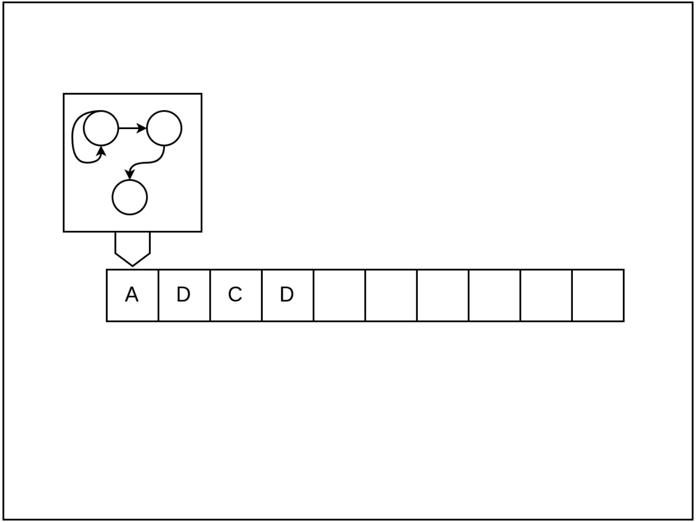
Turing Machine



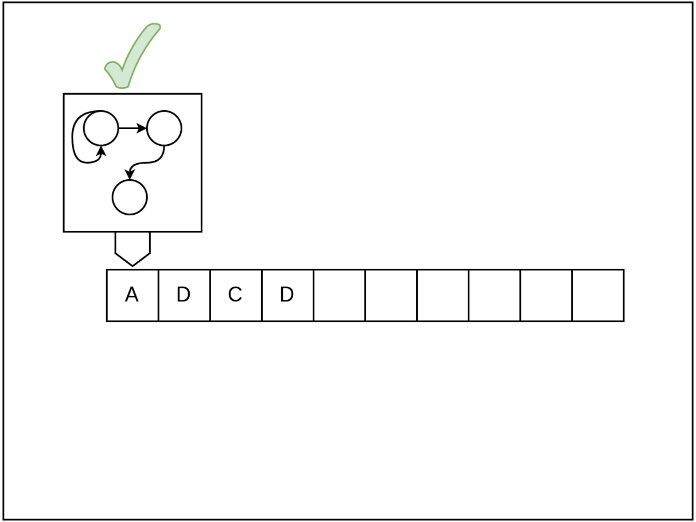
Turing Machine



Turing Machine



Turing Machine



Languages

- Like all the other machines we've seen before, we can talk about their **languages**
- However, there is some more subtlety
  - $\textsc{Accept}(M)$: the language of all inputs w where M accepts.
  - $\textsc{Reject}(M)$: the language of all inputs w where M rejects.
  - $\textsc{Halt}(M)$: the language of all inputs w where M halts.
    - $\textsc{Halt}(M) = \textsc{Accept}(M) \cup \textsc{Reject}(M)$
  - $\textsc{Diverge}(M)$: the language of all inputs w where M never halts.

Decision Languages

If a problem only has a true or false response, we can talk about the **decision language** of that problem:

$$\begin{aligned}
\text{SAT} &= \{w \in \Sigma^* \mid w \text{ is a satisfiable boolean formula}\} \\
\text{SORT} &= \{w \in \Sigma^* \mid w \text{ defines a sorted integer array}\} \\
\text{HALT} &= \{w \in \Sigma^* \mid w \text{ is a program that halts}\}
\end{aligned}$$

Recognizability and Decidability

- We say that M ***recognizes/accepts*** L if for any input $w \in L$, M accepts w.
  - If $w \in L$, then M must **accept** w
  - If $w \notin L$, then M can **reject** or even **never halt**
- M ***decides*** L if for any input w, M accepts if $w \in L$ and rejects otherwise.
  - If $w \in L$, then M must **accept** w
  - If $w \notin L$, then M must **reject** w
  - Either way, M must halt on all inputs

Classifying Languages

- L is **recognizable** if there exists some TM M that recognizes it
- L is **decidable** if there exists some TM M that decides it

Examples

- $\{ n \mid n \in \mathbb{Z} \text{ and } n \text{ is even} \}$
  - This is decidable
- $A_{\text{TM}} = \{ \langle M, w \rangle \mid M \text{ is a TM and } M \text{ accepts } w \}$
  - This is recognizable
  - It is NOT decidable (we'll see why later)

Questions?

Section 2

The Halting Problem

What is The Halting Problem?

- In 1928 at the Second International Congress of Mathematicians in Paris, David Hilbert posed the "Entscheidungsproblem" asking "is mathematics decidable?"
- Alan Turing in 1936 showed that a solution to the Entscheidungsproblem is **impossible**
  - Alonzo Church independently proved this 3 months prior
- We are going to look the proof of this today

The Language *HALT*

- Showing that a language is decidable is easy
  - Just build a decider
- Showing that a language is undecidable is much harder
  - You have to show no machine can decide the language, no matter what you try (done using contradiction).
- $HALT = \{ \langle M, w \rangle \mid M \text{ is a TM and } M \text{ halts on input } w \}$
  - We will show this language is undecidable

The Proof

Suppose *HALT* is decidable. Then we have a Turing Machine H that decides it:

```
H(⟨M, w⟩):
    If M halts on input w:
        return accept
    else:
        return reject
```

The Proof

Using H, we can build a new machine SelfHalt:

```
SelfHalt(⟨M⟩):
    return H(⟨M, M⟩)
```

Then we can build a machine OPP that does the opposite of SelfHalt

```
Opp(⟨M⟩):
    If SelfHalt(⟨M⟩) accepts:
        loop forever
    else reject
```

Lets call Opp(⟨Opp⟩) and see what happens

The Proof

$\underline{\textsc{Opp}(\langle \textsc{Opp} \rangle)}$:

    If $\textsc{SelfHalt}(\langle \textsc{Opp} \rangle)$ accepts:

        loop forever

    else *reject*

$\underline{\textsc{SelfHalt}(\langle \textsc{Opp} \rangle)}$:

    return $\mathrm{H}(\langle \textsc{Opp}, \textsc{Opp} \rangle)$

$\underline{\mathrm{H}(\langle \textsc{Opp}, \textsc{Opp} \rangle)}$:

    If $\textsc{Opp}$ halts on input $\langle \textsc{Opp} \rangle$:

        return *accept*

    else:

        return *reject*

Questions?

Section 3

An Unrecognizable Language

The Language A_{TM}

- $A_{\text{TM}} = \{ \langle M, w \rangle \mid M \text{ is a TM and } M \text{ accepts } w \}$
  - Intuitively, this should be undecidable since it's very similar to the Halting Problem
  - We will prove this very quickly!

A_{TM} is Undecidable

Suppose H decides A_{TM}. So H is the following machine:

$$H(\langle M, w\rangle) = \begin{cases} \mathit{accept} & \text{if } M \text{ accepts } w \\ \mathit{reject} & \text{if } M \text{ does not accept } w \end{cases}$$

Now consider a machine D that uses H as a subroutine

$$D(\langle M\rangle) = \text{ the opposite of } H(\langle M, M\rangle)$$

$$D(\langle M\rangle) = \begin{cases} \mathit{accept} & \text{if } M \text{ does not accept } \langle M\rangle \\ \mathit{reject} & \text{if } M \text{ accepts } \langle M\rangle \end{cases}$$

A_{TM} is Undecidable

$$D(\langle M \rangle) = \begin{cases} \textit{accept} & \text{if } M \text{ does not accept } \langle M \rangle \\ \textit{reject} & \text{if } M \text{ accepts } \langle M \rangle \end{cases}$$

What is $D(\langle D \rangle)$?

$$D(\langle D \rangle) = \begin{cases} \textit{accept} & \text{if } D \text{ does not accept } \langle D \rangle \\ \textit{reject} & \text{if } D \text{ accepts } \langle D \rangle \end{cases}$$

Co-recognizable Languages

- L is **recognizable** if there exists some TM M that recognizes it
- L is **co-recognizable** if there exists some TM M that recognizes it's complement $\Sigma^* \setminus L$
  - $\Sigma^* \setminus L =$ all strings not in L
- Important theorem: L is decidable if and only if L is recognizable AND L is co-recognizable
  - Decider = run the recognizer and co-recognizer in parallel
  - Decidable $\implies$ recognizable and co-recognizable

Complement of A_{TM} is Unrecognizable

- We know a language is decidable if and only if recognizable AND co-recognizable
- A_{TM} is recognizable
  - Just run input M on input w, accept if M accepts
- A_{TM} is undecidable
- Therefore the complement of A_{TM} is unrecognizable

Questions?

No, I'm not interested in developing a powerful brain. All I'm after is just a mediocre brain, something like the President of the American Telephone and Telegraph Company.

— ALAN TURING, on the possibilities of a thinking machine (1943)