Different Kinds of Turing Machines

Wed October 21, 2020
HW5 out

• The README requirement is back
  • Time spent
  • Who you discussed hw with
  • Other sources consulted
CS420, So Far

- **Turing Machines (TMs)**
  - Infinite tape (memory), arbitrary read/write
  - Models “computers”

- **PDAs: recognize context-free languages**
  - Infinite stack (memory), push/pop only
  - Can’t recognize langs w. arbitrary dependency, e.g., \( \{ww \mid w \in \{0,1\}^*\} \)

- **DFAs / NFAs: recognize regular langs**
  - Finite states (memory)
  - Can’t recognize langs w. dependency e.g., \( \{0^n1^n \mid n \geq 0\} \)

Turing-recognizable

Decidable

Context-free

Regular

Algorithms (halting TMs) (next week)
Last time: Turing Machines

- Turing Machines can read and write to input “tape”
- The read-write “head” can move arbitrarily left or right
- The tape is infinite
- A Turing Machine can accept/reject at any time
Multi-Tape Turing Machines

\[ M \]

\[ \begin{array}{c}
0 & 1 & 0 & 1 & 0 & \square \\
\end{array} \]

\[ \begin{array}{c}
a & a & a & \square \\
\end{array} \]

\[ \begin{array}{c}
b & a & \square \\
\end{array} \]
Single-tape TM $\Leftrightarrow$ Multi-tape TM

• $\Rightarrow$ If a single-tape TM recognizes a language, then a multi-tape TM recognizes the language
  • A single-tape TM is a multi-tape TM that does not use other tapes

• $\Leftarrow$ If a multi-tape TM recognizes a language, then a single-tape TM recognizes the language
  • Convert multi-tape TM to single-tape TM
Multi-tape TM $\Rightarrow$ Single-tape TM

- Use delimiter (\#) on single-tape to simulate multiple tapes
- Add “dotted” version of every char to simulate multiple heads

\[ M \]
\[ S \]

\[
\begin{array}{c}
\text{01010 □} \\
\text{aaa □} \\
\text{ba □} \\
\text{#01010#aaa#ba#□} \\
\end{array}
\]
Single-tape TM $\Leftrightarrow$ Multi-tape TM

• $\Rightarrow$ If a single-tape TM recognizes a language, then a multi-tape TM recognizes the language
  • A single-tape TM is a multi-tape TM that does not use other tapes

• $\Leftarrow$ If a multi-tape TM recognizes a language, then a single-tape TM recognizes the language
  • Convert multi-tape TM to single-tape TM (DONE!)
Non-deterministic Turing Machines (NTMs)
Last time: Turing Machine formal def

**Definition 3.3**

A **Turing machine** is a 7-tuple, \((Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})\), where \(Q, \Sigma, \Gamma\) are all finite sets and

1. \(Q\) is the set of states,
2. \(\Sigma\) is the input alphabet not containing the **blank symbol** \(\sqcup\),
3. \(\Gamma\) is the tape alphabet, where \(\sqcup \in \Gamma\) and \(\Sigma \subseteq \Gamma\),
4. \(\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}\) is the transition function,
5. \(q_0 \in Q\) is the start state,
6. \(q_{\text{accept}} \in Q\) is the accept state, and
7. \(q_{\text{reject}} \in Q\) is the reject state, where \(q_{\text{reject}} \neq q_{\text{accept}}\).
Non-deterministic Turing Machine formal definition:

A Non-deterministic Turing Machine is a 7-tuple, \((Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})\), where \(Q\), \(\Sigma\), \(\Gamma\) are all finite sets and

1. \(Q\) is the set of states,
2. \(\Sigma\) is the input alphabet not containing the \textit{blank symbol} \(\_\),
3. \(\Gamma\) is the tape alphabet, where \(\_\) \(\in\) \(\Gamma\) and \(\Sigma \subseteq \Gamma\),
4. \(\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}\)
5. \(q_0 \in Q\) is the start state,
6. \(q_{\text{accept}} \in Q\) is the accept state, and
7. \(q_{\text{reject}} \in Q\) is the reject state, where \(q_{\text{reject}} \neq q_{\text{accept}}\).
Deterministic TM $\iff$ Nondeterministic TM

- $\Rightarrow$ If a deterministic TM recognizes a language, then a nondeterministic TM recognizes the language
  - Deterministic TM $\to$ nondeterministic TM
  - Wrap output of delta in a set

- $\Leftarrow$ If a nondeterministic TM recognizes a language, then a deterministic TM recognizes the language
  - Nondeterministic TM $\to$ deterministic TM
  - ???
Nondeterminism

Deterministic computation

- start
- ...
- accept or reject

Nondeterministic computation

- ...
- reject
- ...

In nondeterministic computation, every step can branch into a set of states

What is a “state” for a TM?

\[ \delta : Q \times \Gamma \rightarrow \mathcal{P}(Q \times \Gamma \times \{L, R\}) \]
TM Configuration = Representation of 1 step

States

1 step

0 1 1 0 0 0 # 0 1 1 0 0 0 □ ...  
x 1 1 0 0 0 # 0 1 1 0 0 0 □ ...  
x 1 1 0 0 0 # x 1 1 0 0 0 □ ...  
x 1 1 0 0 0 # x 1 1 0 0 0 □ ...  
x x 1 0 0 0 # x 1 1 0 0 0 □ ...  
x x x x x x x # x x x x x x □ ...  
x x x x x x x # x x x x x x □ ...  
x x x x x x □ ...  
accept
TM Configuration = State + Head + Tape

Textual representation of “configuration”
Nondeterminism

Deterministic computation

Nondeterministic computation

For TMs, each node is a configuration

start

accept or reject

reject

accept
Nondeterministic TM $\rightarrow$ Deterministic: 1st way

• Deterministic TM simulates nondeterministic TM:
  • When computation branches, deterministic TM keeps multiple configurations on its (single) tape
  • (Similar to how a single-tape TM simulates multitapes)
  • It steps each config one-by-one, adding or removing when needed
  • Accept if accepting config is found
  • Note: Must step configs breadth-first (why?)

Deterministic TM keeps all configs at each step on one tape
Nondeterministic TM $\Rightarrow$ Deterministic: 2nd way

- Deterministic TM simulating nondeterministic TM:
  - Number the nodes at each step
  - Deterministically check every tree path, in breadth-first order
    - 1
    - 1-1
    - 1-2
    - 1-1-1
    - 1-1-2
    - and so on
  - Accept if accepting config found
Nondeterministic TM $\Rightarrow$ Deterministic: 2nd way

Always has input, never changes

Used to run each path

Tracks which node we are on, e.g., 1-1-2, etc.

Needs 3 tapes

$D$

Input tape

Simulation tape

Address tape
Nondeterministic TM $\Leftrightarrow$ Deterministic TM

- $\Rightarrow$ If a deterministic TM recognizes a language, then a nondeterministic TM recognizes the language
  - Deterministic TM $\rightarrow$ nondeterministic TM
  - Wrap output of delta in a set

- $\Leftarrow$ If a nondeterministic TM recognizes a language, then a deterministic TM recognizes the language
  - Nondeterministic TM $\rightarrow$ deterministic TM (DONE!)
All Equivalent TMs!

- Single-tape Turing Machine
- Multi-tape Turing Machine
- Nondeterministic Turing Machine
Check-in Quiz 10/21
On gradescope
End of Class Survey 10/21
See course website