18.404/6.840 Lecture 7

Last time:

- Equivalence of variants of the Turing machine model
    a. Multi-tape TMs
    b. Nondeterministic TMs
    c. Enumerators
- Church-Turing Thesis
- Notation for encodings and TMs

Today:

- Decision procedures for automata and grammars

Will have mini chat-breaks (experiment)

TMs and Encodings – review

A TM has 3 possible outcomes for each input w:

1. <u>*Accept*</u> w (enter q_{acc})
2. <u>*Reject*</u> w by halting (enter q_{rej})
3. <u>*Reject*</u> w by looping (running forever)

A is <u>T-recognizable</u> if $A = L(M)$ for some TM M.

A is <u>T-decidable</u> if $A = L(M)$ for some TM decider M.

halts on all inputs

$\langle O_1, O_2, \ldots, O_k \rangle$ encodes objects $O_1, O_2, \ldots, O_k$ as a single string.

Notation for writing a TM M is

$M =$ "On input w

[English description of the algorithm]"

Acceptance Problem for DFAs

Let $A_{\text{DFA}} = \{\langle B, w\rangle \mid B$ is a DFA and B accepts $w\}$

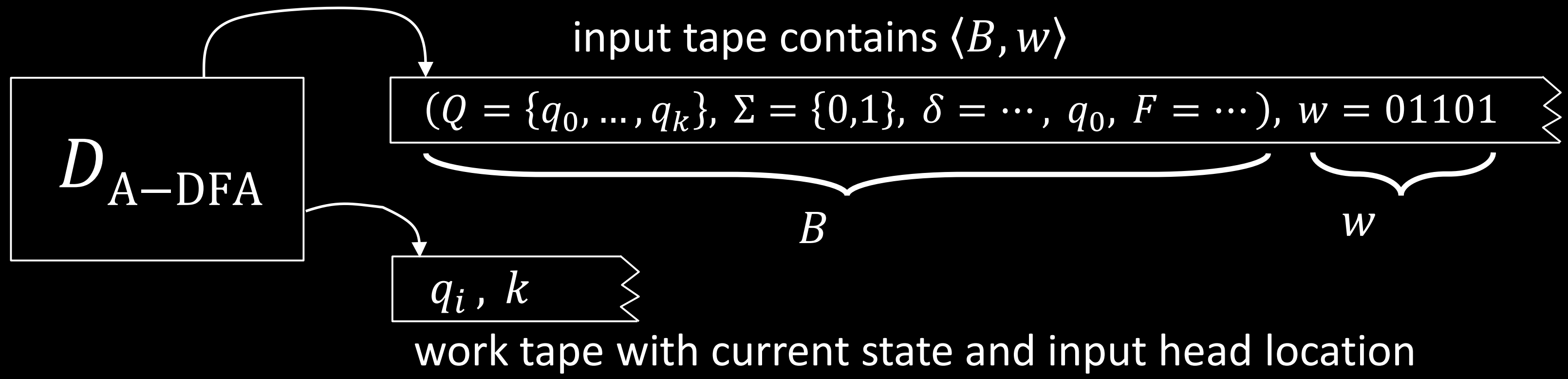
Theorem: A_{DFA} is decidable

Proof: Give TM $D_{\text{A-DFA}}$ that decides A_{DFA}.

$D_{\text{A-DFA}} =$ "On input s

1. Check that s has the form $\langle B, w\rangle$ where B is a DFA and w is a string; *reject* if not.
2. Simulate the computation of B on w.
3. If B ends in an accept state then *accept*. If not then *reject*."

Shorthand:
On input $\langle B, w\rangle$



Acceptance Problem for NFAs

Let $A_{\mathrm{NFA}} = \{\langle B, w\rangle |\ B$ is a NFA and B accepts $w\}$

Theorem: A_{NFA} is decidable

Proof: Give TM $D_{\mathrm{A-NFA}}$ that decides A_{NFA}.

$D_{\mathrm{A-NFA}} =$ "On input $\langle B, w\rangle$

1. Convert NFA B to equivalent DFA B'.
2. Run TM $D_{\mathrm{A-DFA}}$ on input $\langle B', w\rangle$. [Recall that $D_{\mathrm{A-DFA}}$ decides A_{DFA}]
3. *Accept* if $D_{\mathrm{A-DFA}}$ accepts.
   Reject if not."

New element: Use conversion construction and previously constructed TM as a subroutine.

Emptiness Problem for DFAs

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

Theorem: E_{DFA} is decidable

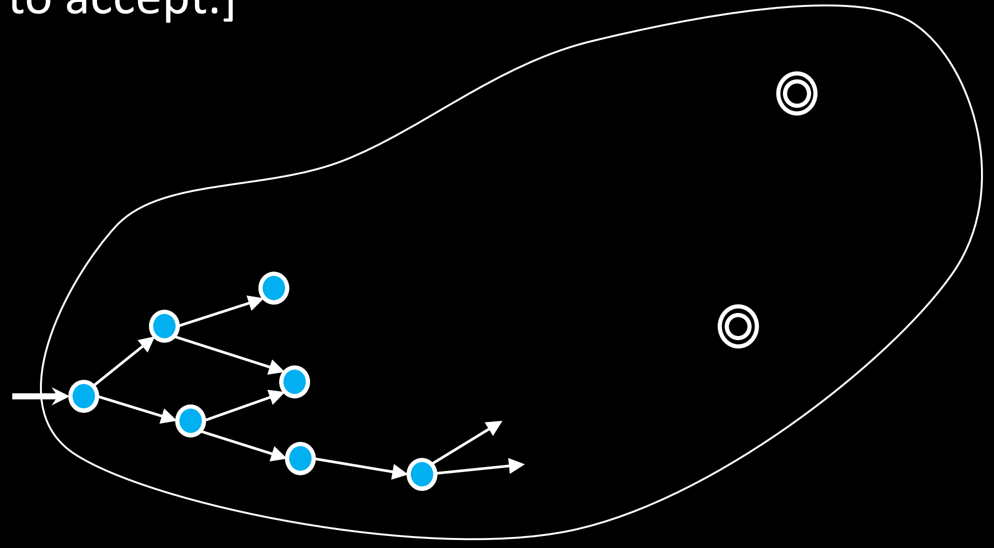
Proof: Give TM $D_{\text{E-DFA}}$ that decides E_{DFA}.

$D_{\text{E-DFA}}$ = "On input $\langle B \rangle$ [IDEA: Check for a path from start to accept.]

1. Mark start state.
2. Repeat until no new state is marked:

   Mark every state that has an incoming arrow
   from a previously marked state.

3. *Accept* if no accept state is marked.
   Reject if some accept state is marked."



Equivalence problem for DFAs

Let $EQ_{\mathrm{DFA}} = \{\langle A, B\rangle |\ A \text{ and } B \text{ are DFAs and } L(A) = L(B)\}$

Theorem: EQ_{DFA} is decidable

Proof: Give TM $D_{\mathrm{EQ-DFA}}$ that decides EQ_{DFA} .

Check-in 7.1

Let $EQ_{\mathrm{REX}} = \{\langle R_1, R_2\rangle |\ R_1 \text{ and } R_2 \text{ are regular expressions and } L(R_1) = L(R_2)\}$

Can we now conclude that EQ_{REX} is decidable?

a) Yes, it follows immediately from things we've already shown.

b) Yes, but it would take significant additional work.

c) No, intersection is not a regular operation.

Check-in 7.1


Coffee
Break

Teach at Splash!

esp.mit.edu/splash20

Splash is an annual teaching and learning extravaganza, brought to you by MIT ESP!

When? November 14-15

Where? Virtual

What? Teach anything! Any topic, length, or class size!

Who? Teach thousands of curious and motivated high schoolers

Acceptance Problem for CFGs

Let $A_{\mathrm{CFG}} = \{\langle G, w\rangle \mid G$ is a CFG and $w \in L(G)\}$

Theorem: A_{CFG} is decidable

Proof: Give TM $D_{\mathrm{A-CFG}}$ that decides A_{CFG} .

$D_{\mathrm{A-CFG}} =$ "On input $\langle G, w\rangle$

1. Convert G into CNF.
2. Try all derivations of length $2|w| - 1$.
3. *Accept* if any generate w.
   Reject if not.

Check-in 7.2

Can we conclude that A_{PDA} is decidable?

a) Yes.

b) No, PDAs may be nondeterministic.

c) No, PDAs may not halt.

Recall Chomsky Normal Form (CNF) only allows rules:

A → BC

B → b

Lemma 1: Can convert every CFG into CNF.
Proof and construction in book.

Lemma 2: If H is in CNF and $w \in L(H)$ then every derivation of w has $2|w| - 1$ steps.
Proof: exercise.

Check-in 7.2

Emptiness Problem for CFGs

Let $E_{\text{CFG}} = \{\langle G \rangle | \ G$ is a CFG and $L(G) = \emptyset \}$

Theorem: E_{CFG} is decidable

Proof:

$D_{\text{E-CFG}} =$ "On input $\langle G \rangle$ [IDEA: work backwards from terminals]

1. Mark all occurrences of terminals in G.
2. Repeat until no new variables are marked
   Mark all occurrences of variable A if
   $\text{A} \rightarrow \text{B}_1\text{B}_2 \cdots \text{B}_k$ is a rule and all B_i were already marked.
3. *Reject* if the start variable is marked.
   Accept if not."

S → R T a

R → T b

T → a

Equivalence Problem for CFGs

Let $EQ_{\text{CFG}} = \{\langle G, H\rangle |\ G, H$ are CFGs and $L(G) = L(H)\ \}$

Theorem: EQ_{CFG} is NOT decidable

Proof: Next week.

Let $AMBIG_{\text{CFG}} = \{\langle G\rangle |\ G$ is an ambiguous CFG $\}$

Check-in 7.3

Why can't we use the same technique we used to show EQ_{DFA} is decidable to show that EQ_{CFG} is decidable?

a) Because CFGs are generators and DFAs are recognizers.
b) Because CFLs are closed under union.
c) Because CFLs are not closed under complementation and intersection.

Check-in 7.3

Acceptance Problem for TMs

Let $A_{\mathrm{TM}} = \{\langle M, w\rangle \mid M$ is a TM and M accepts $w\}$

Theorem: A_{TM} is not decidable
Proof: Thursday.

Theorem: A_{TM} is T-recognizable
Proof: The following TM U recognizes A_{TM}

$U =$ "On input $\langle M, w\rangle$

1. Simulate M on input w.
2. *Accept* if M halts and accepts.
3. *Reject* if M halts and rejects.
4. ~~*Reject* if M never halts.~~" Not a legal TM action.

Turing's original "Universal Computing Machine"

Von Neumann said U inspired the concept of a stored program computer.

Quick review of today

1. We showed the decidability of various problems about automata and grammars:
   A_{DFA} , A_{NFA} , E_{DFA} , EQ_{DFA} , A_{CFG} , E_{DFA}
2. We showed that A_{TM} is T-recognizable.