COMP30026
Models of Computation

Lecture 19: Decidable Languages (cont.) and Countable Sets

Mak Nazecic-Andrlon and William Umboh

Semester 2, 2024

Some material from Michael Sipser's slides

Where are we?

Last time:
Equivalence of variants of the Turing machine model
- Multi-tape TMs
- Nondeterministic TMs
Notation for encodings and TMs
Decision procedures for automata and grammars
A_{DFA} , A_{NFA}, E_{DFA}

Today: (Sipser §4.1-4.2)
Techniques for TM Construction
Decision procedures for automata and grammars
EQ_{DFA} , A_{CFG} , E_{CFG} are decidable
Countable Sets

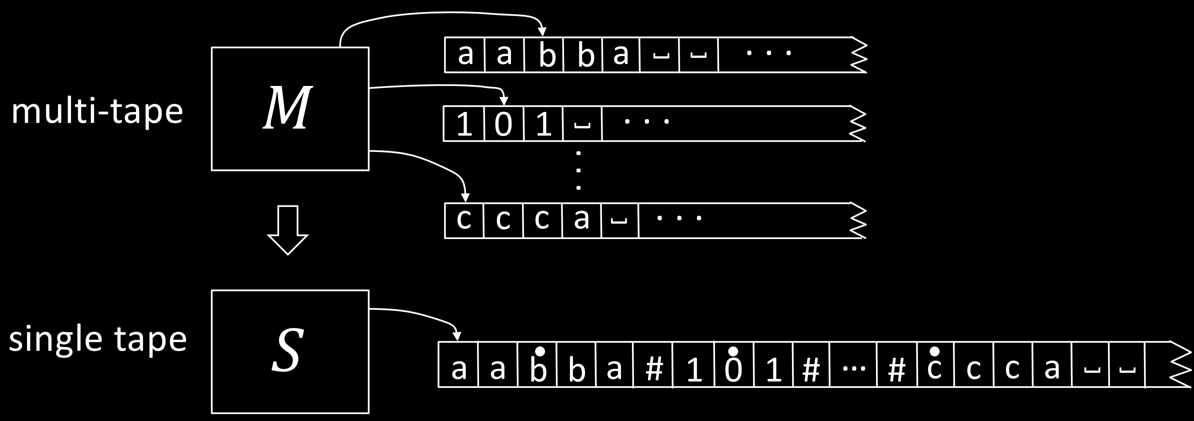
Techniques for TM Construction

Simulation

Construct a TM that simulates another machine

1. Equivalence of variants of the Turing machine model

- e.g. multi-tape TMs



2. Decision procedures for automata and grammars

A_{DFA}

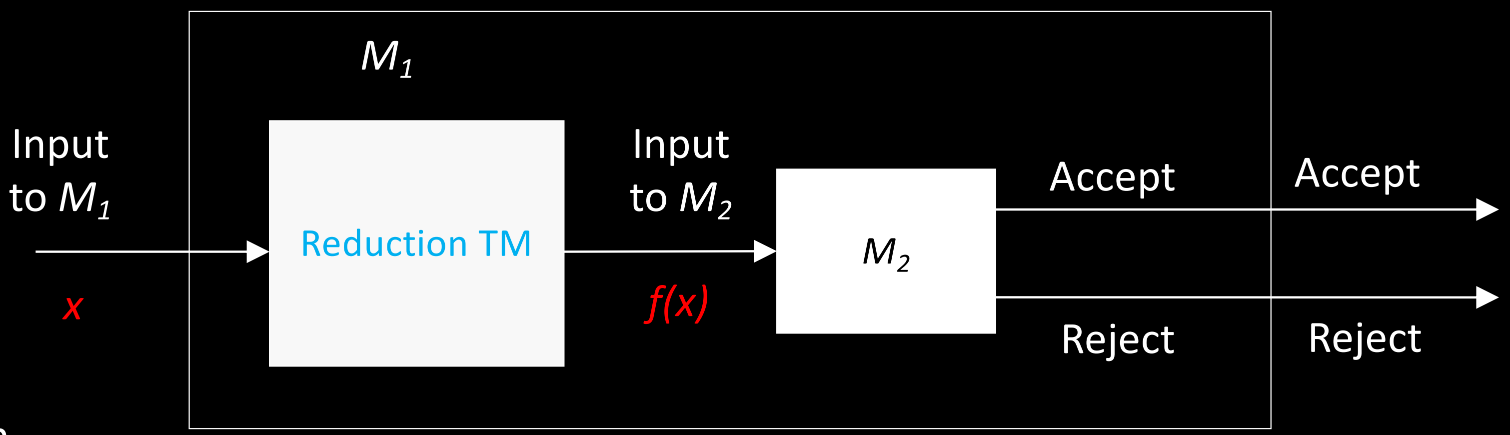
Techniques for TM Construction

Reduction/Reducibility

Suppose:

- Want a recogniser/decider M_1 for language L_1
- Have a recogniser/decider M_2 for language L_2

Then we can build M_1 using a reduction



In Python

```
def M1(x):
  return M2(f(x))
```

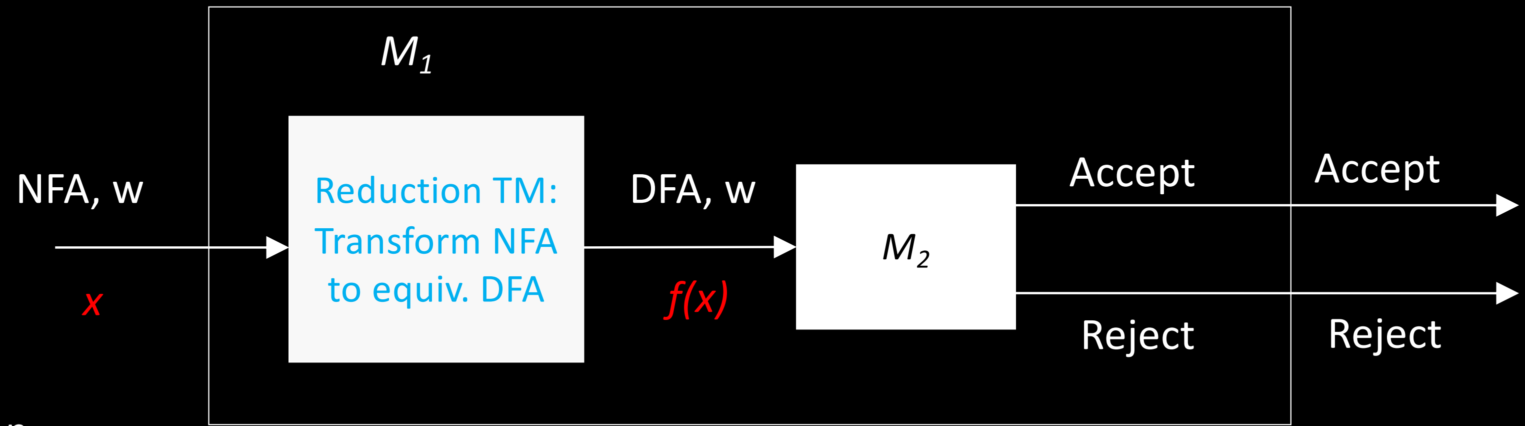
Techniques for TM Construction

Reduction/Reducibility

Suppose:

- Want to recognise/decide language A_{NFA}
- Have a recogniser/decider M_2 for language A_{DFA}

Then we can obtain recogniser/decider M_1 for A_{NFA} using a reduction



In Python
def $M_1(x)$:
  return $M_2(f(x))$

Equivalence problem for DFAs

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

Theorem: EQ_{DFA} is decidable

Proof: Give TM $D_{\text{EQ−DFA}}$ that decides EQ_{DFA} .

$D_{\text{EQ−DFA}}$ = "On input $\langle A, B\rangle$ [IDEA: Make DFA C that accepts w where A and B c

1. Construct DFA C where $L(C) = \left(L(A) \cap \overline{L(B)}\right) \cup \left(\overline{L(A)} \cap L(B)\right)$.
2. Run $D_{\text{E−DFA}}$ on $\langle C\rangle$.
3. *Accept* if $D_{\text{E−DFA}}$ accepts.
   Reject if $D_{\text{E−DFA}}$ rejects."

L

Equivalence problem for DFAs

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

Theorem: EQ_{DFA} is decidable

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

$D_{\mathrm{EQ-DFA}} =$ "On input $\langle A, B\rangle$ [IDEA: Make DFA C that accepts w where A and B c

1. Construct DFA C where $L(C) = \left(L(A) \cap \overline{L(B)}\right) \cup \left(\overline{L(A)} \cap L(B)\right)$.
2. Run $D_{\mathrm{E-DFA}}$ on $\langle C\rangle$.
3. *Accept* if $D_{\mathrm{E-DFA}}$ accepts.
   Reject if $D_{\mathrm{E-DFA}}$ rejects."

L

Equivalence problem for DFAs

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

Theorem: EQ_{DFA} is decidable

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

Check-in 19.1

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

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

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

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

c) No, intersection is not a regular operation.

Acceptance Problem for CFGs

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

Theorem: A_{CFG} is decidable

Brute-force search all possible derivations?

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.

Corollary: Every CFL is decidable.

Proof: Let A be a CFL, generated by CFG G.

Construct TM $M_G =$ "on input w

1. Run $D_{\mathrm{A-CFG}}$ on $\langle G, w\rangle$.
2. *Accept* if $D_{\mathrm{A-CFG}}$ accepts
   Reject if it rejects."

Chomsky Normal Form (CNF) only allows rules:

S → ε (only rule producing ε)

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.

Acceptance Problem for CFGs

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

Theorem: A_{CFG} is decidable

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

$D_{\text{A}-\text{CFG}} =$ "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 19.2

Can we conclude that A_{PDA} is decidable?

a) Yes.

b) No, PDAs may be nondeterministic.

c) No, PDAs may not halt.

Chomsky Normal Form (CNF) only allows rules:

S → ε (only rule producing ε)

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.

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
   $\mathrm{A} \rightarrow \mathrm{B}_1\mathrm{B}_2 \cdots \mathrm{B}_k$ 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

Invariant: every marked symbol can generate non-empty string of terminals

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 $\}$

Theorem: $AMBIG_{\text{CFG}}$ is NOT decidable

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: Coming up.

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.

Acceptance Problem for TMs

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

Proof uses the diagonalization method,
so we will introduce that first.

The Size of Infinity

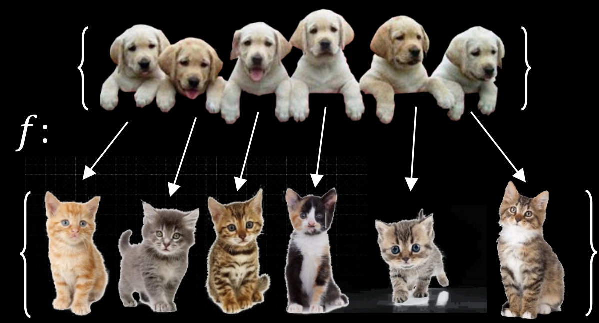
How to compare the relative sizes of infinite sets?

Cantor (~1890s) had the following idea.

Informally, two sets have the same size if we can pair up their members.

Defn: Say that set A and B <u>have the same size</u> if there is a $\underbrace{\text{one-to-one}}_{\substack{x \neq y \to \\ f(x) \neq f(y) \\ \text{"injective"}}}$ and $\underbrace{\text{onto}}_{\substack{\text{Range}\,(f) = B \\ \text{"surjective"}}}$ function $f: A \to B$

We call such an f a <u>1-1 correspondence</u>



This definition works for finite sets.

Apply it to infinite sets too.

Countable Sets

Let $\mathbb{N} = \{1,2,3, \dots\}$ and let $\mathbb{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\}$

Show $\mathbb{N}$ and $\mathbb{Z}$ have the same size

n	$f(n)$
$\mathbb{N}$	$\mathbb{Z}$

Let $\mathbb{Q}^+ = \{ {}^{m}/_{n} \mid m, n \in \mathbb{N}\}$

Show $\mathbb{N}$ and $\mathbb{Q}^+$ have the same size

$\mathbb{Q}^+$	1	2	3	4	…
1	1/1	1/2	1/3	1/4	
2	2/1	2/2	2/3	2/4	…
3	3/1	3/2	3/3	3/4	
4	4/1	4/2	4/3	4/4	
⋮		⋮			

n	$f(n)$
$\mathbb{N}$	$\mathbb{Q}^+$

Think of table as a grid graph and $f(n)$ is the n th number in BFS traversal starting from top-left corner

Defn: A set is <u>countable</u> if it is finite or it has the same size as $\mathbb{N}$.

Both $\mathbb{Z}$ and $\mathbb{Q}^+$ are countable.

Bonus: Countable Sets

Construction similar to one for converting TM to enumerator and NTMs to DTMs

$C_{t,i}$ = t-th configuration of TM M on w_i

	w_1	w_2	w_3	…
1	$C_{1,1}$	$C_{1,2}$	$C_{1,3}$	
2	$C_{2,1}$	$C_{2,2}$	$C_{2,3}$	…
3	$C_{3,1}$	$C_{3,2}$	$C_{3,3}$	
4	$C_{4,1}$	$C_{4,2}$	$C_{4,3}$	
⋮		⋮		

Quick review of today

1. Simulation and reduction
2. We showed the decidability of various problems about automata and grammars:
   E_{DFA} , A_{CFG} , E_{CFG}
3. A_{TM} is T-recognizable
4. Countable Sets