CS 611: Theory of Computation

Hongmin Li

Department of Computer Science
California State University, East Bay

Notation for encodings and TMs

- If O is some object (e.g., polynomial, automaton, graph, etc.), we write $\langle O \rangle$ to be an encoding of that object into a string.
- If $O_1, O_2, \ldots, O_k$ is a list of objects then we write $\langle O_1, O_2, \ldots, O\_k \rangle$ to be an encoding of them together into a single string.

Notation for writing Turing machines:
We will use high-level English descriptions of algorithms when we describe TMs, knowing that we could (in principle) convert those descriptions into states, transition function, etc. Our notation for writing a TM M is

```
M = ''On input w
    [English description of the algorithm]''
```

TM - example revisited

TM M recognizing $B = \{a^k b^k c^k | k \geq 0\}$

```
M = ``On input w
    1.  Check if w ∈ a*b*c*, reject if not
    2.  Count the number of a, b, c in w
    3.  Accept if all counts are equal; reject if not.''
```

High-level description is ok. You do not need to manage tapes, states, etc ...

Acceptance Problem for DFAs

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

Theorem:

A_{DFA} is decidable

Proof: Give TM $D_{A_{DFA}}$ that decides A_{DFA}.

Acceptance Problem for DFAs

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

Theorem:

A_{DFA} is decidable

Proof: Give TM $D_{A_{DFA}}$ that decides A_{DFA}.

$D_{A_{DFA}}$ = `On input` $\langle B, w\rangle$

1. `Simulate the computation of` B `on` w
2. `If` B `ends in an accept state then accept.`
   `If not then reject`

Acceptance Problem for NFAs

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

Theorem:

A_{NFA} is decidable

Proof: Give TM $D_{A_{NFA}}$ that decides A_{NFA}.

Acceptance Problem for NFAs

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

Theorem:

A_{NFA} is decidable

Proof: Give TM $D_{A_{NFA}}$ that decides A_{NFA}.

$D_{A_{NFA}}$ = On input $\langle B, w\rangle$

1. Simulate the computation of B on w
2. If B ends in an accept state then accept.
   If not then reject

Acceptance Problem for NFAs

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

Theorem:

A_{NFA} is decidable

Proof: Give TM $D_{A_{NFA}}$ that decides A_{NFA}.

Acceptance Problem for NFAs

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

Theorem:

A_{NFA} is decidable

Proof: Give TM $D_{A_{NFA}}$ that decides A_{NFA}.

$D_{A_{NFA}}$ = On input $\langle B, w\rangle$

1. Convert NFA B to equivalent DFA B'
2. Run TM $D_{A_{DFA}}$ on input $\langle B', w\rangle$
   (Recall $D_{A_{DFA}}$ decides A_{DFA})
3. If $D_{A_{DFA}}$ accepts then accept. Reject if not.

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

Emptiness Problem for DFAs

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

Theorem:

E_{DFA} is decidable

Proof: Give TM $D_{E_{DFA}}$ that decides E_{DFA}.

Emptiness Problem for DFAs

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

Theorem:

E_{DFA} is decidable

Proof: Give TM $D_{E_{DFA}}$ that decides E_{DFA}.

$D_{E_{DFA}}$ = ‘‘On input $\langle B \rangle$

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

Theorem:

EQ_{DFA} is decidable

Proof: Give TM $D_{EQ_{DFA}}$ that decides EQ_{DFA}.

Equivalence Problem for DFAs

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

Theorem:

EQ_{DFA} is decidable

Proof: Give TM $D_{EQ_{DFA}}$ that decides EQ_{DFA}.

$D_{EQ_{DFA}}$ = ‘‘ On input $\langle A, B\rangle$
1. Construct DFA C
   where $L(C) = (L(A) \cap \overline{L(B)}) \cup (\overline{L(A)} \cap L(B))$
2. Run $D_{E_{DFA}}$ on $\langle C\rangle$
3. Accept if $D_{E_{DFA}}$ accept.
   Reject if $D_{E_{DFA}}$ reject."

Acceptance Problem for CFGs

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

Theorem:

A_{CFG} is decidable

Proof: Give TM $D_{A_{CFG}}$ that decides A_{CFG}.

Acceptance Problem for CFGs

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

Theorem:

A_{CFG} is decidable

Proof: Give TM $D_{A_{CFG}}$ that decides A_{CFG}.

$D_{A_{CFG}}$ = ‘‘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.’’

Acceptance Problem for CFGs

Recall Chomsky Normal Form (CNF) only allows rules: $A \to BC$
$B \to b$

Lemma 1:

Can convert every CFG into CNF.

Proof and construction in book.

Acceptance Problem for CFGs

Recall Chomsky Normal Form (CNF) only allows rules: $A \to BC$
$B \to b$

Lemma 1:

Can convert every CFG into CNF.

Proof and construction in book.

Lemma 2:

If G is in CNF and $w \in L(G)$ then every derivation of w has $2|w| - 1$ steps.

Proof: exercise.

Acceptance Problem for CFGs

Recall Chomsky Normal Form (CNF) only allows rules: $A \rightarrow BC$
$B \rightarrow b$

Corollary:

Every CFL is decidable

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

M_G = ``On input $\langle w \rangle$
1. Run $D_{A_{CFG}}$ on $\langle G, w \rangle$
2. Accept if $D_{A_{CFG}}$ accepts. Reject if it rejects.

Emptiness Problem for CFGs

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

Theorem:

E_{CFG} is decidable

Proof: Give TM $D_{E_{CFG}}$ that decides E_{CFG}.

Emptiness Problem for CFGs

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

Theorem:

E_{CFG} is decidable

Proof: Give TM $D_{E_{CFG}}$ that decides E_{CFG}.

$D_{E_{CFG}}$ = ‘‘On input $\langle G \rangle$

1. Mark all occurences of terminal in G
2. Repeat until no new variables are marked
   Mark all occurrences of variable A
   if $A \rightarrow B_1 B_2 \dots B_k$ is a rule
   and all B_i were already marked
3. Reject if the start variable is marked
   Accept if not.’’

Equivalence Problem for CFGs

Let $EQ_{CFG} = \{\langle G_1, G_2 \rangle | G_1, G_2$ are CFGs and $L(G_1) = L(G_2)\}$

Theorem:

EQ_{CFG} is undecidable

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

Theorem:

$AMBIG_{CFG}$ is undecidable

Acceptance Problem for TMs

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

Theorem:

A_{TM} is undecidable

Proof: next lecture

Theorem:

A_{TM} is T-recognizable.

Acceptance Problem for TMs

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

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 w
2. If simulated M accepts w, then accept else reject (by moving to q_{rej})

Acceptance Problem for TMs

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

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 w
2. If simulated M accepts w, then accept else reject (by moving to q_{rej})

U (the Universal TM) accepts $\langle M, w\rangle$ iff M accepts w. i.e.,

$$L(U) = A_{\text{TM}}$$

Acceptance Problem for TMs

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

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 w
2. If simulated M accepts w, then accept else reject (by moving to q_{rej})

U (the Universal TM) accepts $\langle M, w\rangle$ iff M accepts w. i.e.,

$$L(U) = A_{\text{TM}}$$

But U does not decide A_{TM}

Acceptance Problem for TMs

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

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 w
2. If simulated M accepts w, then accept
   else reject (by moving to q_{rej})

U (the Universal TM) accepts $\langle M, w\rangle$ iff M accepts w. i.e.,

$$L(U) = A_{\text{TM}}$$

But U does not decide A_{TM}: If M rejects w by not halting, U rejects $\langle M, w\rangle$ by not halting.

Acceptance Problem for TMs

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

Theorem:

A_{TM} is T-recognizable.

Proof: The following TM U recognizes A_{TM}

```
U = ``On input ⟨M, w⟩
    1.  Simulate M on w
    2.  If simulated M accepts w, then accept
        else reject (by moving to q_rej)
```

U (the Universal TM) accepts $\langle M, w\rangle$ iff M accepts w. i.e.,

$$L(U) = A_{\rm TM}$$

But U does not decide $A_{\rm TM}$: If M rejects w by not halting, U rejects $\langle M, w\rangle$ by not halting. Indeed (as we shall see) no TM decides $A_{\rm TM}$.