CSE 105
THEORY OF COMPUTATION

Fall 2021

http://cseweb.ucsd.edu/classes/fa21/cse105-a/
Today’s lecture

- Examples of decidable languages
- Examples of undecidable languages
- Reading: Finish Sipser Chapter 4
- More about mapping reducibility
- Read Sipser Chapter 5.3 (again!)
Some example languages

• Acceptance problem:
  - $A_{DFA} = \{ <M,w> | M \text{ is a DFA and } M(w) \text{ accepts} \}$
  - $A_{TM} = \{ <M,w> | M \text{ is a TM and } M(w) \text{ accepts} \}$

• Emptyness problem:
  - $E_{DFA} = \{ <M> | M \text{ is a DFA and } L(M) \text{ is the empty set} \}$
  - $E_{TM} = \{ <M> | M \text{ is a TM and } L(M) \text{ is the empty set} \}$

• Equivalence problem:
  - $EQ_{DFA} = \{ <M,M'> | M \text{ and } M' \text{ are DFAs and } L(M) = L(M') \}$
Some example languages

• Acceptance problem:
  – $A_{DFA} = \{ <M,w> | M \text{ is a DFA and } M(w) \text{ accepts } \}$ DECIDABLE
  – $A_{TM} = \{ <M,w> | M \text{ is a TM and } M(w) \text{ accepts } \}$ UNDECIDABLE

• Emptiness problem:
  – $E_{DFA} = \{ <M> | M \text{ is a DFA and } L(M) \text{ is the empty set } \}$ DECIDABLE
  – $E_{TM} = \{ <M> | M \text{ is a TM and } L(M) \text{ is the empty set } \}$ UNDECIDABLE

• Equivalence problem:
  – $EQ_{DFA} = \{ <M,M'> | M \text{ and } M' \text{ are DFAs and } L(M) = L(M') \}$ DECIDABLE
A\textsubscript{DFA} is decidable

- P\textsubscript{ADFA}(x) =
  - Parse input x as \langle\langle Q, \Sigma, \delta, s, F \rangle, w \rangle. If parse fails, then reject.
  - Let q = s
  - For i=1..|w|
    - Let q = \delta(q,w[i])
  - If q is in F, then accept, otherwise reject
A_{DFA} is decidable

\[ P_{ADFA}(x) = \]

- Parse input x as \(\langle Q, \Sigma, \delta, s, F \rangle, w \rangle\). If parse fails, then reject.
- Let \( q = s \)
- For \( i = 1 \ldots |w| \)
  - Let \( q = \delta(q, w[i]) \)
- If \( q \) is in \( F \), then accept, otherwise reject

Which of the following is true?

A) \( L(P_{ADFA}) = A_{DFA} \)
B) \( P_{ADFA} \) is a decider
C) both A) and B)
D) Neither A) nor B)
$A_{PDA}$ is decidable

- $P_{APDA}(x) =$

1) Parse input $x$ as $\langle Q, \Sigma, \Gamma, \delta, s, F, w \rangle$. If parse fails, then reject.

2) Let $C = \{(s, w, [])\}$. (initial configuration)

3) if $(q, [], t) \in C$ for some $q \in F$ and $t \in \Gamma^*$, then accept

4) For any $(q, aw, ct) \in C$ and $(q', c') \in \delta(q, a, c)$
   
   If $(q', w, c't)$ is not in $C$,
   
   then $C \leftarrow C \cup \{(q', w, c't)\}$ and go to 3)

5) Reject
A\textsubscript{PDA} is decidable

- \textbf{P\textsubscript{APDA}}(x) =

  1) Parse input \(x\) as \(<<Q, \Sigma, \Gamma, \delta, s, F>, w>\).
  2) Let \(C = \{(s, w, [])\}\). (initial configuration)
  3) if \((q, [], t) \in C\) for some \(q \in F\) and \(t \in \Gamma^*\), then accept
  4) For any \((q, aw, ct) \in C\) and \((q', c') \in \delta(q, a, c)\)
     If \((q', w, c't)\) is not in \(C\),
     then \(C \leftarrow C \cup \{(q', w, c't)\}\) and go to 3)
  5) Reject

Which of the following is true?
A) \(L(P\textsubscript{APDA}) = A\textsubscript{PDA}\)
B) \(P\textsubscript{APDA}\) is a decider
C) both A) and B)
D) Neither A) nor B)
$E_{\text{DFA}}$ is decidable

- $P_{\text{EDFA}}(w) =$

  1) Parse input $w$ as $\langle Q, \Sigma, \delta, s, F \rangle$. If parse fails, then reject.
  2) Let $X = \{s\}$
  3) For all $q$ in $X$ and $a$ in $\Sigma$
      - Let $q' = \delta(q,a)$
      - If $q'$ is not in $X$, then $X \leftarrow X \cup \{q'\}$ and restart the loop at 3)
  4) If $X$ intersects $F$, then accept, else reject.
\( \mathbb{E}_{DFA} \) is decidable

\( \mathbb{P}_{EDFA}(w) = \)

1) Parse input \( w \) as \( \langle Q, \Sigma, \delta, s, F \rangle \). If parse fails, then reject.
2) Let \( X = \{s\} \)
3) For all \( q \) in \( X \) and \( a \) in \( \Sigma \)
   - Let \( q' = \delta(q,a) \)
   - If \( q' \) is not in \( X \), then \( X \leftarrow X \cup \{q'\} \) and restart the loop at 3)
4) If \( X \) intersects \( F \), then accept, else reject.

Which of the following is true?

A) \( L(\mathbb{P}_{EDFA}) = \mathbb{E}_{DFA} \)
B) \( \mathbb{P}_{EDFA} \) is a decider
C) both A) and B)
D) Neither A) nor B)
$\text{EQ}_{\text{DFA}}$ is decidable

- $P_{\text{EQDFA}}(<M,M'>)$
  1) Check if both $M$ and $M'$ are DFA. If not, then reject.
  2) Use closure properties of regular languages to build a DFA $M''$ for
     \[ L(M'') = (L(M) - L(M')) \cup (L(M') - L(M)) \]
     
     - Run $P_{\text{EDFA}}(<M'>).$ If $P_{\text{EDFA}}$ accepts, then accept, else reject.

- **Notice** $L(M) = L(M')$ if and only if $L(M'')$ is empty
  - $P_{\text{EQDFA}}$ is a decider
  - $L(P_{\text{EQDFA}}) = \text{EQ}_{\text{DFA}}$
$\text{EQ}_{\text{DFA}}$ is decidable (by reduction)

- $F(<M,M'>)$:
  1) If input does not parse ($M,M'$ are not DFAs), then output “garbage”
  2) Use closure properties of regular languages to build a DFA $M''$ for
     \[ L(M'') = (L(M) - L(M')) \cup (L(M') - L(M)) \]
  3) Output $<M''>$

- Notice $L(M)=L(M')$ if and only if $L(M'')$ is empty

- Equivalently, $<M,M'> \in \text{EQ}_{\text{DFA}}$ if and only if $F(<M,M'>) \in E_{\text{DFA}}$
\textbf{EQ}_{\text{DFA}} \text{ is decidable (C)}

- \textbf{F}(<M,M'>):
  1) If input does not parse (\(M,M'\) are not DFAs), then output "garbage"
  2) Use closure properties of regular languages to build a DFA \(M''\) for \(L(M'') = (L(M) - L(M')) \cup (L(M') - L(M))\)
  3) Output \(<M''>\)

- Notice \(L(M)=L(M')\) if and only if \(L(M'')\) is empty

- Equivalently, \(<M,M'> \in \text{EQ}_{\text{DFA}}\) if and only if \(F(<M,M'>) \in \text{E}_{\text{DFA}}\)

Is \(F\) a mapping reduction from \(A=\text{EQ}_{\text{DFA}}\) to \(B=\text{E}_{\text{DFA}}\)?

A) Yes, because \(w \in A\) iff \(F(x) \in B\)
B) No, because \(B\) is decidable
C) No, it is a reduction from \(B\) to \(A\)
D) What is a reduction anyway?
Mapping reducibility

- Let $A, B \subseteq \Sigma^*$ be any two languages
- Definition: $A$ is map-reducible to $B$ (written “$A \leq_m B$”) if there is a function $f$ such that:
  1) The function $f$ is computable (by a TM $M$)
  2) For all $w \in A$, we have $f(w) \in B$
  3) For all $w$ not in $A$, we have that $f(w)$ is not in $B$
- Equivalently: $(w \in A) \iff (f(w) \in B)$
Reductions and undecidability

What we proved:

• Theorem: if $A \prec_m B$, and $A$ is undecidable, then $B$ is undecidable.

• Equivalently, if $A \prec_m B$ (i.e., $A$ map reduces to $B$), then
  - $(A$ is undecidable $) \rightarrow (B$ is undecidable $)$

• Equivalently, if $A \prec_m B$ (i.e., $A$ map reduces to $B$), then
  - $\neg(B$ is undecidable $) \rightarrow \neg(A$ is undecidable $)$
  - $(B$ is decidable $) \rightarrow (A$ is decidable $)$
Reductions and undecidability

- Let $F$ be a TM computing a map reduction from $A$ to $B$
- What we proved: if $B$ is decidable, then $A$ is decidable
  - Let $M$ is a decider for $B$
  - Goal: build a decider $M'$ for $A$
- $M'(w)$:
  1) Compute $w' = F(w)$
  2) Run $M(w')$
  3) If $M(w')$ accepts, then accept $w$, otherwise reject $w$
Reductions and undecidability

- Let F be a TM computing a map reduction from A to B
- WTS: if B is in RE, then A is in RE
  - Let M be a TM such that L(M) = B
  - Goal: build a TM M’ such that L(M’) = A
- M’(w):
  1) Compute w’ = F(w)
  2) Run M(w’)
  3) If M(w’) accepts, then accept w, otherwise reject w
Reductions and undecidability

- Let \( F \) be a TM computing a map reduction from \( A \) to \( B \).
- \( \text{WTS: if } B \text{ is in RE, then } A \text{ is in RE} \)
  - Let \( M \) be a TM such that \( L(M) = B \).
  - Goal: build a TM \( M' \) such that \( L(M') = A \).

\[ M'(w): \]
1) Compute \( w' = F(w) \)
2) Run \( M(w') \)
3) If \( M(w') \) accepts, then accept \( w \), otherwise reject \( w \).

\( M' \) is a decider

A) Yes
B) No, because it may loop in step 1
C) No, because it may loop in step 2
D) It depends on \( w \)
E) I don’t know
Reductions and undecidability

Let $F$ be a TM computing a map reduction from $A$ to $B$

WTS: if $B$ is in RE, then $A$ is in RE

1. Let $M$ be a TM such that $L(M)=B$
2. Goal: build a TM $M'$ such that $L(M')=A$

$M'(w)$:

1. Compute $w' = F(w)$
2. Run $M(w')$
3. If $M(w')$ accepts, then accept $w$, otherwise reject $w$

What can you say about $L(M')$?

A) $L(M') = A$
B) $L(M') \subset A$
C) $L(M') \supset A$
D) None of the above
E) I don’t know
Proof of $A \subseteq L(M')$

- Let $F : A <_m B$, and $L(M) = B$

- $M'(w)$:
  1) Compute $w' = F(w)$
  2) Run $M(w')$, and accept iff $M(w')$ accept

- Assume $w \in A$, then
  - $F(w) \in B$ (by definition of reduction)
  - $M(w')$ accepts
  - $M(w)$ accepts
Proof of $A \supset L(M')$

- Let $F: A \prec_m B$, and $L(M) = B$

- $M'(w)$:
  1) Compute $w' = F(w)$
  2) Run $M(w')$, and accept iff $M(w')$ accept

- Assume $w$ is not in $A$, then
  - $F(w)$ is not in $B$ (by definition of reduction)
  - $M(w')$ rejects or loops
  - $M(w)$ rejects or loops
... therefore

- If $A \leq_m B$ and $B$ is RE, then $A$ is RE
- What about coRE?
- Claim: if $F$ is a reduction from $A$ to $B$, then $F$ is also a reduction

A) From $B$ to $A$
B) From $B$ to $A$
C) From $A$ to $B$
D) I don’t know
If $A \leq_m B$ and $B$ is RE, then $A$ is RE

What about coRE?

Theorem: if $F$ is a reduction from $A$ to $B$, then $F$ is also a reduction from $A$ to $B$

- $(w \in A) \iff (F(w) \in B)$
- $\neg(w \in A) \iff \neg(F(w) \in B)$
- $(w \in A) \iff (F(w) \in B)$

Corollary: if $A \leq_m B$ and $B$ is coRE, then $A$ is coRE
Mapping reducibility Summary

- Assume $A \leq_m B$, i.e., there is a map reduction from $A$ to $B$
- Then, we have
  - If $B$ is RE, then $A$ is RE
  - If $B$ is coRE, then $A$ is coRE
  - If $B$ is decidable, then $A$ is decidable
  - If $A$ is undecidable, then $B$ is undecidable
  - If $A$ is not in RE, then $B$ is not in RE
  - If $A$ is not in coRE, then $B$ is not in coRE
For next Time

- Try to prove that $\text{HALT}^\text{TM}$ is undecidable
- Reading: Sipser *Chapters 5*