Today’s lecture

- Examples of undecidable languages
- Examples of decidable languages
- Reductions
- Reading: Chapter 5
Previously ...

- Diag = \{ <M> | M is a TM s.t. <M> is not in L(M) \} 
  - Undecidable (in fact, not even RE)

- DD = \{ <M,<M>> | M is a TM s.t. <M> is not in L(M) \} 
  - Just like Diag: Undecidable (in fact, not even RE)
  - Proof: Assume DD is decidable. It follows that Diag is decidable. Contradiction!

- $A_{TM} = \{ <M,w> | M is a TM such that M(w) accepts \}$ 
  - $A_{TM}$ is RE, but not Decidable. Therefore, also not coRE.
  - Proof: Assume $A_{TM}$ is decidable. It follows that MM is decidable. Contradiction!
A_{TM} is undecidable (Proof details)

- Assume for contradiction A_{TM} is decidable
- MM = \{<M,<M>> | M is a TM\} is decidable
- MM - A_{TM} is decidable because the class of decidable languages is closed under set difference.
- Notice: MM - A_{TM}
  
  = \{<M,<M>> | M is a TM\} - \{<M,w> | w \in L(M) \}
  
  = \{<M,<M>> | M is a TM and <M> \notin L(M) \} = DD

- So, DD is decidable. Contradiction!
Some example languages

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

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

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

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

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

- **Equivalence problem:**
  - $EQ_{DFA} = \{ <M,M'> \mid M \text{ and } M' \text{ are DFAs and } L(M) = L(M') \}$ DECIDABLE
$E_{DFA}$ is decidable

- $P_{EDFA}(w) =$
  
  1. Parse input $w$ as $<Q, \Sigma, \delta, s, F>$. If parse fails, then reject.
  2. Let $X = \{s\}$
  3. For all $q$ in $X$ and $a$ in $\Sigma$
     a) Let $q' = \delta(q, a)$
     b) 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.
$E_{\text{DFA}}$ is decidable

- $P_{\text{EDFA}}(w) =$
  1. Parse input $w$ as $<Q, \Sigma, \delta, s, F>$.
  2. Let $X = \{s\}$
  3. For all $q$ in $X$ and $a$ in $\Sigma$:
    a) Let $q' = \delta(q, a)$
    b) 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.

Is $P_{\text{EDFA}}$ a decider?

A) No, because it can loop in step 3
B) Yes, because DFAs always terminate
C) Yes, because step 3 is executed at most $|Q|$ times
D) Yes, because step 3 is executed at most $|\Sigma|$ times
E) I don’t know
E_{DFA} is decidable

- \( P_{EDFA}(w) = \)
  1. Parse input \( w \) as \( <Q, \Sigma, \delta, s, F> \). If parse fails, then reject.
  2. Let \( X = \{s\} \)
  3. For all \( q \) in \( X \) and \( a \) in \( \Sigma \)
     - a) Let \( q' = \delta(q,a) \)
     - b) 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.

What is the language of \( P_{EDFA} \)?

A) \{\langle P, w \rangle \mid P \text{ is a DFA and } P(w) \text{ accepts} \}
B) \{\langle P \rangle \mid P \text{ is a DFA} \}
C) \{\langle P \rangle \mid P \text{ is a DFA and } L(P) = \emptyset \}
D) \{\langle P \rangle \mid P \text{ is a DFA and } L(P) = \{\varepsilon\} \}
E) None of the above
$E_{DFA}$ is decidable

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

- \( P_{EQDFA}(<M,M'>) \)
  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 \( P_{EDFA}(<M''>) \). If \( P_{EDFA} \) accepts, then accept, else reject.

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

- DD is decidable $\rightarrow$ Diag is decidable
  - Since Diag is undecidable, then DD is also undecidable

- $A_{TM}$ is decidable $\rightarrow$ DD is decidable
  - Since DD is undecidable, then $A_{TM}$ is undecidable

- $E_{DFA}$ is decidable $\rightarrow$ $EQ_{DFA}$ is decidable
  - Since $E_{DFA}$ is decidable, then $EQ_{DFA}$ is decidable
Proving implications

• Claim: If A is decidable, then B is decidable

• Proof:
  – Assume A is decidable
  – Show that B is also decidable

• Proof methods:
  – Using closure properties of decidable languages: transform A into B
  – Let M be a decider for A. Use M to build a decider for B

• Notice: We do not need to know if A is decidable
Reductions

• “If A is decidable then B is decidable”
  – Can be proved using closure properties, or assuming a decider for A
  – Makes sense even when A or B are undecidable

• When decidability of B is unknown
  – This reduces the problem of solving B to the (possibly easier) problem of solving A

• Proof is called a “reduction” from B to A
  – Oftern written $B \leq A$
Reductions

• “If A is decidable then B is decidable”
  - Can be proved using closure properties, or assuming a decider for A
  - Makes sense even when A or B are undecidable

• When decidability of B is unknown
  - This reduces the problem of solving B to the (possibly easier) problem of solving A

• Proof is called a “reduction”
  - Often written $B < A$

Assume $B < A$ and A is decidable. What can you conclude?

A) B is also decidable
B) B is a subset of A
C) B is undecidable
D) B may be decidable or undecidable
E) I don’t know
Reductions

• “If A is decidable then B is decidable”
  - Can be proved using closure properties for A
  - Makes sense even when A or B are undecidable

• When decidability of B is unknown
  - This reduces the problem of solving B to the possibly easier problem of solving A

• Proof is called a “reduction”
  - Often written $B \leq A$

Assume $A \leq B$ and A is decidable. What can you conclude?

A) B is also decidable
B) B is a subset of A
C) B is undecidable
D) B may be decidable or undecidable
E) I don’t know
Reductions

• “If A is decidable then B is decidable”
  - Can be proved using closure properties for A
  - Makes sense even when A or B are undecidable

• When decidability of B is unknown
  - This reduces the problem of solving B to the (possibly easier) problem of solving A

• Proof is called a “reduction” from B to A
  - Often written $B \leq A$

Assume $A \leq B$ and $A$ is undecidable. What can you conclude?

A) B is also decidable
B) B is a subset of A
C) B is undecidable
D) B may be decidable or undecidable
E) I don’t know
Reductions

- “If A is decidable then B is decidable”
  - Can be proved using closure properties, or assuming a decider for A
  - Makes sense even when A or B are undecidable

- When decidability of B is unknown
  - This reduces the problem of solving B to the (possibly easier) problem of solving A

- Proof is called a “reduction” from B to A
  - Often written $B \leq A$

Assume $B \leq A$ and A is undecidable. What can you conclude?

A) B is also decidable
B) B is a subset of A
C) B is undecidable
D) B may be decidable or undecidable
E) I don’t know
Some problems on CFG

- $\text{EQ}_{\text{CFG}} = \{ <G_1, G_2> \mid G_1, G_2 \text{ CFG s.t. } L(G_1) = L(G_2) \}$
- $\text{SUB}_{\text{CFG}} = \{ <G_1, G_2> \mid G_1, G_2 \text{ CFG s.t. } L(G_1) \subseteq L(G_2) \}$
- $\text{SUP}_{\text{CFG}} = \{ <G_1, G_2> \mid G_1, G_2 \text{ CFG s.t. } L(G_1) \supseteq L(G_2) \}$

Can you give reductions between any two of these problems? In what direction?

- $\text{EQ}_{\text{CFG}} < \text{SUB}_{\text{CFG}}$ ?
- $\text{SUB}_{\text{CFG}} < \text{SUP}_{\text{CFG}}$ ?
- $\text{SUP}_{\text{CFG}} < \text{EQ}_{\text{CFG}}$ ?
Next Time

- We have class on Wednesday
- Thursday and Friday: Thanksgiving!
- **Haskell 4** due tonight
- **HW7** due Tue Nov 29
- Last week of classes after Thanksgiving