Today's learning goals

- Explain what it means for a problem to be decidable.
- Justify the use of encoding.
- Give examples of decidable problems.
- Prove that a computational problem about DFA, NFA, RegExp, etc. is decidable.

Group HW5 due Saturday
Practice Qs on website - Review session Monday evening
Exams 2
Recommended extra practice Qs
Additional office hours next week
Seat assignments on Plaza TBA
Computational problems over $\Sigma$

- $\text{ADFA}$: "Is a given string accepted by a given DFA?"
  \[ \{ \langle B, w \rangle \mid B \text{ is a DFA, } w \in \Sigma^*, \text{ and } w \text{ is in } L(B) \} \]
  \[ L(M_i) = \text{ADFA} \text{, } M_i \text{, decider} \]

- $\text{EDFA}$: "Is the language of a DFA empty?"
  \[ \{ \langle A \rangle \mid A \text{ is a DFA over } \Sigma, L(A) \text{ is empty} \} \]

- $\text{EQDFA}$: "Are the languages of two given DFAs equal?"
  \[ \{ \langle A, B \rangle \mid A \text{ and } B \text{ are DFA over } \Sigma, L(A) = L(B) \} \]
Computational problems

A computational problem is **decidable** iff the language encoding the problem instances is decidable.

*We won't specify the encoding.*

To prove decidable, define TM $M = \text{"On input \langle\ldots\rangle, 1. 2. \ldots\"}$

Show (1) $L(M) = \ldots$ and (2) $M$ is a decider.
Proving decidability

Claim: \( E_{\text{DFA}} \) is decidable

Proof: WTS that \( \{ <A> | A \text{ is a DFA over } \Sigma, L(A) \text{ is empty} \} \) is decidable.

Informally: what do you look for in the state diagram of a DFA to determine if it accepts *at least one* string?

Q: is there a path from \( q_0 \) to any state in \( F \)?
Proving decidability

Claim: $E_{DFA}$ is decidable

Proof: WTS that \{ $<A>$ | $A$ is a DFA over $\Sigma$, $L(A)$ is empty \} is decidable.

Test cases:

e.g. $<$ is in $E_{DFA}$; $<$ is not in $E_{DFA}$

TM deciding $E_{DFA}$ should accept and should reject
Proving decidability

Claim: $E_{DFA}$ is decidable

Proof: WTS that $\{ \langle A \rangle \mid A \text{ is a DFA over } \Sigma, L(A) \text{ is empty} \}$ is decidable.

Step 1: construction

Idea: breadth-first search in state diagram to look for paths to $F$
Proving decidability

Claim: \( E_{\text{DFA}} \) is decidable

Proof: WTS that \{ \langle A \rangle \mid A \text{ is a DFA over } \Sigma, L(A) \text{ is empty} \} \) is decidable.

Step 1: construction

Idea: breadth-first search in state diagram to look for paths to \( F \)

Define TM \( M_2 \) by: \( M_2 = \{ \text{On input } \langle A \rangle; \}

1. Check whether input is a valid encoding of a DFA, if not, reject.
2. Mark the start state of \( A \).
3. Repeat until no new states get marked:
   i. Loop over states of \( A \) and mark any unmarked state that has an incoming edge from a marked state.
4. If no final state of \( A \) is marked, accept; otherwise, reject.
Proving decidability

Step 2: correctness proof

WTS (1) $L(M_2) = E_{DFA}$ and (2) $M_2$ is a decider.

Consider string $w$. First, assume $w \in E_{DFA}$. WTS $M_2$ accepts $w$.

Trace $M_2$ on $w$. By assuming $w$, it is of the right type so pass step 1. Go ahead to BFs in 2 & 3 and when get to step 4, accept. b/c $L(A) = \emptyset$ so no states in $F$ are reachable from $q_0$. 
Next, assume (instead) \( w \notin E_{\mathit{TA}} \) so \( w \neq <A> \) for any \( A \in \mathit{TA} \) or \( w = <A> \) and \( A \notin \mathit{TA} \) \( l(A) = \emptyset \).

Trace \( M_2 \) on \( w \) in step 1, fail type check and \( M_2 \) rejects.

After \( BTA \), have at least one accepting state be marked (let \( \chi \in L(A) \) consider \( \delta^*(<q_0, \chi>) \)) so in step 4, \( M_2 \) rejects.
Non-emptiness?

"Is the language of a DFA non-empty?"

Is this problem decidable?

A. Yes, using $M_3$ in the handout.

B. Yes, using $M_4$ in the handout.

C. Yes, both $M_3$ and $M_4$ work.

D. Yes, but not using the machines in the handout.

E. No.
$M_4 = "\text{On input } \langle A \rangle, \text{ A DFA over } \Sigma \text{. For each } x \in \Sigma^* \text{,}\}
\begin{align*}
1. & \text{Run } M_1 \text{ on } \langle A, x \rangle \\
2. & \text{if component accepts, accept} \\
\text{otherwise, reject}\}
\end{align*}$

Never stop if 
$\text{let } l(A) = \emptyset$. *

Could modify bound length of $x$ in loop.
Proving decidability

Claim: \( \text{EQ}_{\text{DFA}} \) is decidable

Proof: WTS that \( \{ <A, B> | A, B \text{ are DFA over } \Sigma, L(A) = L(B) \} \) is decidable. Idea: give high-level description

Step 1: construction

Will we be able to simulate \( A \) and \( B \)?
What does set equality mean?
Can we use our previous work?
Proving decidability

Claim: \( \text{EQ}_{\text{DFA}} \) is decidable

Proof: WTS that \( \{ <A, B> \mid A, B \text{ are DFA over } \Sigma, L(A) = L(B) \} \) is decidable. Idea: give high-level description

Step 1: construction

Will we be able to simulate A and B?

What does set equality mean?

Can we use our previous work?
Proving decidability

Claim: $\text{EQ}_{\text{DFA}}$ is decidable

Proof: WTS that $\{ <A, B> | A, B \text{ are DFA over } \Sigma, L(A) = L(B) \}$ is decidable. Idea: give high-level description

Step 1: construction

Very high-level:

**Build new DFA** recognizing symmetric difference of $L(A)$, $L(B)$. Check if this set is empty.
Proving decidability

Claim: $\text{EQ}_\text{DFA}$ is decidable

Proof: WTS that $\{ <A, B> \mid A, B$ are DFA over $\Sigma$, $L(A) = L(B) \}$ is decidable. Idea: give high-level description

Step 1: construction

Define TM $M_5$ by: $M_5 =$ "On input $<A,B>$ where $A,B$ DFAs:

1. Construct a new DFA, $D$, from $A,B$ using algorithms for complementing, taking unions of regular languages such that $L(D) =$ symmetric difference of $L(A)$ and $L(B)$.
2. Run machine $M_2$ on $<D>$.
3. If it accepts, accept; if it rejects, reject."

Proof Details:
Proving decidability

Step 1: construction
Define TM $M_5$ by: $M_5 =$ "On input $<A,B>$ where $A,B$ DFAs
1. Construct a new DFA, $D$, from $A,B$ using algorithms for complementing, taking unions of regular languages such that $L(D) =$ symmetric difference of $L(A)$ and $L(B)$.
2. Run machine $M_2$ on $<D>$.
3. If it accepts, accept; if it rejects, reject."

Step 2: correctness proof
WTS (1) $L(M_5) = EQ_{DFA}$ and (2) $M_5$ is a decider.
Computational problems

Which of the following computational problems are decidable?

A. $A_{NFA}$
B. $E_{NFA}$
C. $EQ_{NFA}$
D. All of the above
E. None of the above
Computational problems

Compare:

A. $A_{\text{REX}} = A_{\text{NFA}} = A_{\text{DFA}}$, $E_{\text{REX}} = E_{\text{NFA}} = E_{\text{DFA}}$, $EQ_{\text{REX}} = EQ_{\text{NFA}} = EQ_{\text{DFA}}$

B. They're all decidable, some are equal and some not.

C. They're of different types so all are different.

D. None of the above
Techniques

- **Subroutines**: can use decision procedures of decidable problems as subroutines in other algorithms
  - $A_{\text{DFA}}$
  - $E_{\text{DFA}}$
  - $\text{EQ}_{\text{DFA}}$
- **Constructions**: can use algorithms for constructions as subroutines in other algorithms
  - Converting DFA to DFA recognizing complement (or Kleene star).
  - Converting two DFA/NFA to one recognizing union (or intersection, concatenation).
  - Converting NFA to equivalent DFA.
  - Converting regular expression to equivalent NFA.
  - Converting DFA to equivalent regular expression.
Next time

• Are all computational problems decidable?

For Monday, pre-class reading: Section 4.3, page 207-209.