Agenda

• **Yesterday**
  - Reductions (Section 5.3)

• **Today**
  - Section 6.3
  - *We will not be covering section 6.4*
    - *I will discuss some basic issues of this section when covering chapter 7*
Announcement

• Revised homework assignment for next Tuesday
  - 5.4, 5.5, 5.7, 5.9, 6.3

• I will hold extended office hours next week
  - Tuesday & Wednesday 3:00 – 5:00
Mapping reducibility

Definition: Language $A$ is mapping reducible to language $B$, written $A \leq_m B$, if there is a computable function $f : \sum^* \to \sum^*$, where for every $w$,

$$w \in A \text{ iff } f(w) \in B$$
Using mapping reductions

- Test whether \( w \in A \) by finding a mapping reduction \( f \) from \( A \) to be and determining whether \( f(w) \in B \)

- Example
  - \( \text{ALL}_{\text{DFA}} = \{<A> \mid A \text{ is a DFA with } L(A)=\Sigma^*\} \)
  - Let \( D = \{B \mid B \text{ is a DFA}\} \) and let \( f:D \to D \)
    where \( f(A) = \bar{A} \)
  - Then \( A \in \text{ALL}_{\text{DFA}} \) iff \( \bar{A} \in \text{E}_{\text{DFA}} \)
    - Use membership of \( \bar{A} \) in \( \text{E}_{\text{DFA}} \) to determine membership of \( A \) in \( \text{ALL}_{\text{DFA}} \)
Mapping reductions & decidability

Theorem: If $A \leq_m B$ and $B$ is decidable, then $A$ is decidable.

Proof: Let $M$ be a decider for $B$ and let $f$ be a reduction from $A$ to $B$.

Consider the following TM, $N$:

$N = \text{“On input } w:\text{”}$

1. Compute $f(w)$
2. Run $M$ on $f(w)$ and report $M$'s output"

Then $N$ decides $A$
Mapping reductions & undecidability

Corollary: If $A \leq_m B$ and $A$ is undecidable, then $B$ is undecidable.

• We have been using this corollary implicitly already
  - E.g., we showed $A_{TM} \leq_m HALT_{TM}$ and concluded $HALT_{TM}$ is undecidable
Example

• Let \( EV = \{<A>| A \text{ is a DFA all strings in } L(A) \text{ have an even number of 1's}\} \)
  - How can we prove \( EV \) is decidable using a mapping reduction?

• Consider the following DFA \( B \)

\[
L(B) = \{w \in \sum^* | \text{ w has an even number of 1's}\}
\]
Mapping reduction of $L$

- Use $\text{EQ}_{\text{DFA}} = \{<A,B> \mid A \text{ and } B \text{ are DFA's with } L(A) = L(B)\}$
- Mapping from $L$ to $\text{EQ}_{\text{DFA}}$
  - $f(<A>) = <A, A \cap B>$
- $A$ has an even number of 1's if and only if $L(A) = L(A \cap B)$
  - I.e., $A \in \text{EV}$ iff $f(A) \in \text{EQ}_{\text{DFA}}$
Reductions & TM-recognizability

Theorem: If $A \leq_m B$ and $B$ is Turing-recognizable, then $A$ is Turing-recognizable.

Proof: (same as decidable proof) Let $M$ be a recognizer for $B$ and let $f$ be a reduction from $A$ to $B$.

Consider the following TM, $N$:

$N =$ “On input $w$:
1. Compute $f(w)$
2. Run $M$ on $f(w)$ and report $M$’s output”

Then $N$ recognizes $A$
Reductions & non-TM-recognizability

**Theorem:** If $A \leq_m B$ and $A$ is not Turing-recognizable, then $B$ is not Turing-recognizable.

**Question:** Which language have we seen that is not Turing-recognizable?

**Answer:** $\overline{A_{TM}}$
Proving non-Turing-recognizability

Question: If $A \leq_m U$ is $\bar{A} \leq_m \bar{U}$?

Answer: Yes since $x \in A$ iff $f(x) \in U$

(Use this fact when doing homework problem 5.5)

• How can we use the to prove non-Turing-recognizability?

• Prove $A_{TM} \leq_m U$ or prove $A_{TM} \leq_m \bar{U}$
Is mapping reducibility enough?

• Mapping reducibility does not completely capture our intuition about reductions
  - Example: \( A_{TM} \) and \( \overline{A_{TM}} \) are not mapping reducible
    - \( A_{TM} \) is Turing-recognizable and \( \overline{A_{TM}} \) isn’t
  - A solution to \( A_{TM} \) would also provide a solution to \( \overline{A_{TM}} \)
Oracle

• An oracle for a language $B$ is an external device that is capable of reporting whether any string $w$ is a member of $B$
  - We are not concerned how the oracle determines membership

• An oracle Turing machine is a Turing machine that can query an oracle
  - The machine $M_B^B$ can query an oracle for the language $B$
Example

• An oracle Turing machine with an oracle for $E_{QM}$ can decide $E_{TM}$

$T_{EQ-TM} = "On input <M>"

1. Create TM $M_1$ such that $L(M_1) = \emptyset$
   $M_1$ has a transition from start state to reject state for every element of $\sum$

2. Call the $EQ_{TM}$ oracle on input $<M,M_2>$

3. If it accepts, accept; if it rejects, reject"

• $T_{EQ-TM}$ decides $E_{TM}$

• $E_{TM}$ is decidable relative to $EQ_{TM}$
Turing reducibility

- A language $A$ is Turing reducible to a language $B$, written $A \leq_T B$, if $A$ is decidable relative to $B$
- Previous slide shows $E_{TM}$ is Turing reducible to $EQ_{TM}$
- Whenever $A$ is mapping reducible to $B$, then $A$ is Turing reducible to $B$
  - The function in the mapping reducibility could be replaced by an oracle
Applications

• If $A \leq_T B$ and $B$ is decidable, then $A$ is decidable
• If $A \leq_T B$ and $A$ is undecidable, then $B$ is undecidable
• If $A \leq_T B$ and $B$ is Turing-recognizable, then $A$ is Turing-recognizable
• If $A \leq_T B$ and $A$ is non-Turing-recognizable, then $B$ is non-Turing-recognizable
Happy studying!