COMP2210: Theory of Computing

Lecture 18

Reductions

Corina Cirstea
Objectives for Today

• Reductions

By the end of this lecture, you should be able to:

• understand the notion of reduction between two languages
• explain how reductions are used to prove (un)decidability results
• produce reductions that prove certain problems not to be decidable (semi-decidable)
Properties of Languages

- HP and MP talk about arbitrary inputs to TMs. What about “simpler” problems?
- Q. Can we decide, given any TM $M$ as input, whether $M$ accepts $\epsilon$?
- i.e. is $A \overset{\text{def}}{=} \{ M \mid \epsilon \in L(M) \}$ a recursive set?
- A. No! We will see that we can reduce HP to it.

“HP $\leq A$”
The proof

\[ A \overset{\text{def}}{=} \{ M \mid \epsilon \in L(M) \} \]

- Suppose that \( K \) is a total TM that accepts \( A \).
- We construct \( N \) that, on input \( M \# x \):
  1. Constructs \( M' \), that on any input \( y \) ignores its input and simulates \( M \) on \( x \). If \( M \) accepts or rejects, then \( M' \) accepts.
  2. Simulates \( K \) on \( M' \); If \( K \) accepts then accept, if it rejects then reject.

\[
N \text{ accepts } M \# x \iff K \text{ accepts } M' \\
\iff \epsilon \in L(M') \iff M \text{ halts on } x
\]
Recap

\[
\text{MP} \overset{\text{def}}{=} \{ M \# x \mid x \in L(M) \}
\]

\[
A \overset{\text{def}}{=} \{ M \mid \epsilon \in L(M) \}
\]

- We have shown, through reduction from HP, that \( \text{MP} \) and \( A \) are undecidable.

- The proof:
  - if \( A \) or \( \text{MP} \) was decidable, there would be a total TM for it;
  - and we could use this TM to solve HP!
Reduction, formally

- A function \( f : \Sigma^* \rightarrow \Delta^* \) is **computable** when there exists a *total* TM \( K \) that when started with \( x \in \Sigma^* \) on its tape, eventually halts with \( f(x) \in \Delta^* \) on its tape;

- A *reduction* of \( A \subseteq \Sigma^* \) to \( B \subseteq \Delta^* \) is a computable function \( f : \Sigma^* \rightarrow \Delta^* \) s.t. \( x \in A \iff f(x) \in B \);

- Write \( A \leq B \) if there is a reduction from \( A \) to \( B \).
Examples

• $\text{HP} \leq \text{MP}$;
  
  $\circ M \#x \mapsto M' \#x$;

• $\text{HP} \leq \{ M \mid \epsilon \in L(M) \}$;
  
  $\circ M \#x \mapsto M'$;

• ...
Properties of reductions

• If $A \leq B$ and $B$ is r.e. then so is $A$. Equivalently, if $A$ is not r.e. then $B$ is not r.e.;
  
  Proof. If $A \leq B$ then there is a computable function $f$ that takes $a \in A$ to $f(a) \in B$. If $B$ is r.e. then there is a TM $M$ that accepts it. The following is a TM $N$ for $A$: on $x$, first compute $f(x)$, then run $M$ on $f(x)$, accept if it accepts, reject if it rejects.

  \[ x \in L(N) \iff f(x) \in L(M) \iff f(x) \in B \iff x \in A \]

• If $A \leq B$ and $B$ is recursive then so is $A$. Equivalently, if $A$ isn’t recursive then neither is $B$.
  
  Proof. Similar to above.