Recitation 13 - Homework 6 and More Reduction Examples

John Chilton

November 29, 2006

- Homework 6 Problems and Examples

Problem 1. Show EQ_{CFG} is undecidable.

- Pick a problem you know is undecidable, ALL_{CFG} seems relevent

Problem 1. Show EQ_{CFG} is undecidable.

- Pick a problem you know is undecidable, ALL_{CFG} seems relevent
- Direct Reduction: Show how deciding EQ_{CFG} allows for deciding ALL_{CFG}, by describing a TM for deciding ALL_{CFG} using a decider for EQ_{CFG}

Problem 1. Show EQ_{CFG} is undecidable.

- Pick a problem you know is undecidable, ALL_{CFG} seems relevent
- Direct Reduction: Show how deciding EQ_{CFG} allows for deciding ALL_{CFG}, by describing a TM for deciding ALL_{CFG} using a decider for EQ_{CFG}
- Mapping Reduction: Show $ALL_{CFG} \leq_M EQ_{CFG}$. For a given grammar G, describe how to make pair (G_1, G_2) such that G generates all strings iff $L(G_1) = L(G_2)$.

Problem 2. If $A \leq_M B$ and B is regular, does that imply A is regular? Why or why not?

- Consider carefully the power of a computable mapping function
- Consider a simple regular language B, such as $\{1\}$.

Problem 3. Show that T is undecidable in two ways.

$$T = \{M \mid M \text{ accepts } w^R \text{ whenever } M \text{ accepts } w\}$$

Show two ways.

- Do this by applying Rice's Theorem.
- Do this by a reduction

$$T = \{M \mid M \text{ accepts } w^R \text{ whenever } M \text{ accepts } w\}$$

Prove T is undecidable by Rice's Theorem. Do this by showing two things:

- Show T is non-empty and does not contain all possible Turing machines
- Show whenever $L(M_1) = L(M_2)$, $M_1 \in T$ iff $M_2 \in T$

$$T = \{M \mid M \text{ accepts } w^R \text{ whenever } M \text{ accepts } w\}$$

Prove T is undecidable by reduction. Show $A \leq_M T$ for some undecidable language A or explicitly lay out a reduction proof like 5.2 or 5.3.

Read through proof that $REGULAR_{TM}$ is undecidable.

$$SEVEN_{TM} = \{M \mid M \text{ accepts some } w \text{ such that } |w| = 7\}$$

Show $SEVEN_{TM}$ is undecidable. Will do this using reduction and Rice's Theorem.

By contradiction and reduction. Assume $SEVEN_{TM}$ is decidable, then the following decides A_{TM}, a contradiction:

By contradiction and reduction. Assume $SEVEN_{TM}$ is decidable, then the following decides A_{TM}, a contradiction:

$T =$ On input (M, w) where M is a TM.

1. Construct the following TM, S:

   $S =$ On input x:

   1. If $|x| \neq 7$, *accept*.
   2. Else, Simulate M on w, accept if it does.

2. Run $SEVEN_{TM}$ decider on S.
3. If decider accepted, *accept*, else *reject*.

By contradiction and reduction. Assume $SEVEN_{TM}$ is decidable, then the following decides A_{TM}, a contradiction:

$T =$ On input (M, w) where M is a TM.

1. Construct the following TM, S:

   $S =$ On input x:

   1. If $|x| \neq 7$, *accept*.
   2. Else, Simulate M on w, accept if it does.

2. Run $SEVEN_{TM}$ decider on S.
3. If decider accepted, *accept*, else *reject*.

S will accept a string of length 7 iff M accepts w, hence deciding $SEVEN_{TM}$ would allow us to decide A_{TM}. Since A_{TM} is undecidable, $SEVEN_{TM}$ must be undecidable.

By Rice's Theorem. Let P be the property that a given TM accepts some string of length 7.

By Rice's Theorem. Let P be the property that a given TM accepts some string of length 7.

TMs can decide $\{0\}$ and $\{0000000\}$, so some TMs posses property P, but not all

$\therefore P$ is non-trivial

By Rice's Theorem. Let P be the property that a given TM accepts some string of length 7.

TMs can decide $\{0\}$ and $\{0000000\}$, so some TMs posses property P, but not all

$\therefore P$ is non-trivial

If M_1 and M_2 are TMs s.t. $L(M_1) = L(M_2)$, then $w \in L(M_1)$ iff $w \in L(M_2)$, so M_2 accepts some string of length 7 iff M_1 does

$\therefore P$ is property of languages not machines

By Rice's Theorem. Let P be the property that a given TM accepts some string of length 7.

TMs can decide $\{0\}$ and $\{0000000\}$, so some TMs posses property P, but not all

$\therefore P$ is non-trivial

If M_1 and M_2 are TMs s.t. $L(M_1) = L(M_2)$, then $w \in L(M_1)$ iff $w \in L(M_2)$, so M_2 accepts some string of length 7 iff M_1 does

$\therefore P$ is property of languages not machines

Since P is a non-trivial property of languages of Turing machines it is undecidable by Rice's Theorem.

Problem 4. Consider

$$A = \{(M, w) \mid M \text{ moves left on left most tape pos. on } w\}$$

Show A is undecidable.

Problem 4. Consider

$$A = \{(M, w) \mid M \text{ moves left on left most tape pos. on } w\}$$

Show A is undecidable.

- Rice's Theorem valid?

Problem 4. Consider

$$A = \{(M, w) \mid M \text{ moves left on left most tape pos. on } w\}$$

Show A is undecidable.

- Rice's Theorem valid?
- One Approach: Reduce a simple undecidable problem about Turing machines such as A_{TM} to A

Problem 4. Consider

$$A = \{(M, w) \mid M \text{ moves left on left most tape pos. on } w\}$$

Show A is undecidable.

- Rice's Theorem valid?
- One Approach: Reduce a simple undecidable problem about Turing machines such as A_{TM} to A
- How would a decider for this problem allow you to decide A_{TM}

Problem 4. Consider

$$A = \{(M, w) \mid M \text{ moves left on left most tape pos. on } w\}$$

Show A is undecidable.

- Rice's Theorem valid?
- One Approach: Reduce a simple undecidable problem about Turing machines such as A_{TM} to A
- How would a decider for this problem allow you to decide A_{TM}
- Key Idea: When simulating a Turing machine, is it ever necessary to move left on the left most tape position?

Here is a somewhat similar problem. Show the following language is undecidable:

$$NO\$_{TM} = \{(M, w) \mid M \text{ never writes a \$ to the tape on } w\}$$

Will show that a decider for $NO\$_{TM}$, would allow for construction of a decider for A_{TM}. Since A_{TM} is undecidable, the decider for $NO\$_{TM}$ cannot exist and $NO\$_{TM}$ must be undecidable.

If $NO\$_{TM}$ were decidable then some deciding TM would decide it and the following TM would decide A_{TM}.

$T=$ On input (M, w) where M is a TM.

1. M' := Replace \$ with \$′ in formal def. of M
2. w' := Replace \$ with \$′ in w
3. Construct the following TM, S:

   $S=$ On input y:

   1. Simulate modified M' on y.
   2. If M' accepts, write a \$ to the tape halt.
4. Run $NO\$$ decider on (S, w'), if it accepts, *reject*, else *accept*.

Problem 5. Consider

$$A = \{(M, w) \mid M's \text{ head ever moves left on } w\}$$

Show A is decidable.

$$A = \{(M, w) \mid M's \text{ head ever moves left on } w\}$$

The following TM T *recognizes* A.

$T =$ On input (M, x) where M is a TM.

1. Simulate M on input x.
2. If at any point M moves left, *accept*.
3. If M halts, *reject*.

Only recognizes A, because it does not halt if M just continues to move to the right forever. A decider will need to know when this is happening. How can it tell?

Problem 6.

$J = \{w \mid w = 0x \text{ for some } x \in A_{TM} \text{ or } w = 1y \text{ for some } y \in \overline{A_{TM}}\}$

Show J and $\overline{J}$ are not Turing-*recognizable*.
The only not Turing-recognizable language we have seen is $\overline{A_{TM}}$, so try to give a reduction from this to J and then to $\overline{J}$. This should be fairly straight-forward.

Problem 7. Read through pages 199-204. If everything makes sense, this problem should be pretty straight forward. If not, reread until the problem seems straight forward.

Post correspondence problem.

Post correspondence problem.

$$\left\{ \begin{bmatrix} cac \\ c \end{bmatrix}, \begin{bmatrix} b \\ abb \end{bmatrix}, \begin{bmatrix} bba \\ b \end{bmatrix} \right\}$$

Post correspondence problem.

$$\left\{ \begin{bmatrix} cac \\ c \end{bmatrix}, \begin{bmatrix} b \\ abb \end{bmatrix}, \begin{bmatrix} bba \\ b \end{bmatrix} \right\}$$

Solution:

PCP $\begin{bmatrix} bba \\ b \end{bmatrix} \begin{bmatrix} bba \\ b \end{bmatrix} \begin{bmatrix} b \\ abb \end{bmatrix} \begin{bmatrix} b \\ abb \end{bmatrix}$

MPCP Needs to start with $\begin{bmatrix} cac \\ c \end{bmatrix}$, no solution exists.

A MPCP instance, called P', is described on page 200-204 which reduces A_{TM} to $MPCP$. Show that P' always has a trivial match if we have no requirements about the first domino in a match, i.e. if we treat P' as PCP problem and not a MPCP problem.

Some other things to talk about.

- Show J is undecidable.
- Show E_{TM} is not Turing recognizable.
- Show EQ_{CFG} is co-Turing recognizable.
- A computation history problem
- Talk some more about the PCP problem.

Reducing A_{TM} to $MPCP$. Given some (M, w) create an instance of $MPCP$ that has a match iff (M, w) has some accepting computation history.

Big Idea:

$$\left[\frac{\#C_1\#C_2\#C_3\#\ldots\#C_n\#}{\#C_1\#C_2\#C_3\#\ldots\#C_n\#}\right]$$

Part 1. First domino requires bottom starts with a computation history.

$$\left[\frac{\#}{\# q_0 w_1 w_2 \dots w_n \#}\right]$$

Part 2&3.

$$\delta(q, a) = (r, b, R) \text{ add } \begin{bmatrix} qa \\ br \end{bmatrix}$$

$$\delta(q, a) = (r, b, L) \text{ add } \begin{bmatrix} cqa \\ rcb \end{bmatrix} (\forall c \in \Gamma)$$

Part 4&5 For all $\forall a \in \Gamma$ add $\begin{bmatrix} a \\ a \end{bmatrix}$. Add $\begin{bmatrix} \# \\ \# \end{bmatrix}$ and $\begin{bmatrix} \# \\ \sqcup\# \end{bmatrix}$