COMP-170: Homework #3

Ben Tanen - February 12, 2017

Problem 1

Prove that a language L is decidable if and only if L and its complement $\overline{L}$ are recognizable.

* * *

We want to prove that a language L is decidable if and only if L and $\overline{L}$ are recognizable. To do this, we will use proof by construction. More specifically, we will construct recognizers for L and $\overline{L}$ using a decider for L and then we will construct a decider for L using recognizers for L and $\overline{L}$.

(1) L **is decidable** $\rightarrow$ L **and** $\overline{L}$ **are recognizable:** We are given the fact that the language L is decidable. This means there exists a machine that accepts x for all $x \in L$ and rejects y for all $y \in \overline{L}$. Let D be that deciding machine. Using this, let's construct a recognizer R_1 for L.

R_1 **on input** x**:**

Run D on x

If D accepts x, $ACCEPT$

If D rejects x, $REJECT$

END

We can also use D to construct a recognizer R_2 for $\overline{L}$.

R_2 **on input** x**:**

Run D on x

If D accepts x, $REJECT$

If D rejects x, $ACCEPT$

END

We will now claim that R_1 is a recognizer of L. Let x be any element from L. Because $x \in L$, we know that D would accept x. As a result, because R_1 accepts any string that D accepts, we can see that R_1 would accept x. Therefore, since R_1 accepts all $x \in L$, we can see that R_1 is indeed a recognizer of L.

We will also now claim that R_2 is a recognizer of $\overline{L}$. Let y be any element from $\overline{L}$. Because $y \in \overline{L}$, as in $y \notin L$, we know that D would reject y. As a result, because R_2 accepts any string that D rejects, we can see that R_2 would accept y. Therefore, since R_2 accepts all $y \in \overline{L}$, we can see that R_2 is indeed a recognizer of $\overline{L}$.

Since we were able to construct two recognizers for L and $\overline{L}$ using the decider D for L, we can see that if L is decidable, L and $\overline{L}$ are both recognizable.

(2) L and $\overline{L}$ are recognizable $\rightarrow$ L is decidable: We are given the fact that L and $\overline{L}$ are recognizable. This means there exists machines that recognize these sets. Let R_1 be a recognizer for L and let R_2 be a recognizer for $\overline{L}$. Using these, let's construct a decider D for L.

> **D on input x:**
>
> For $i = 1 \rightarrow \infty$
>
> > Run R_1 on x for i steps
> >
> > > If R_1 accepts x in i steps, $ACCEPT$
> > >
> > > If R_1 rejects x in i steps, $REJECT$
> >
> > Run R_2 on x for i steps
> >
> > > If R_2 accepts x in i steps, $REJECT$
> > >
> > > If R_2 rejects x in i steps, $ACCEPT$
>
> END

We will now claim that D is a decider of L. First, let x be any element from L. Because $x \in L$, we know that R_1 *will* accept x after some finite number of steps and R_2 *might* reject x after some finite number of steps. Suppose R_1 accepts x after m step. We can now consider the following cases:

1. Suppose that R_2 does actually reject x after n steps.
   (a) If $m \leq n$, then we know that R_1 will accept x first. This will cause D to accept x, as it should.
   (b) If $m > n$, then we know that R_2 will reject x first. This will cause D to accept x, as it should.
2. Alternately, suppose R_2 never actually rejects x (as in it just loops on x). We know that R_1 will still eventually accept x after m steps. This will cause D to also accept x anyway.

Thus, in any case, we can see that if $x \in L$, D accepts x.

Now, let x be any element from $\overline{L}$. Because $x \notin L$, we know that R_1 *might* reject x after some finite number of steps and we know that R_2 *will* reject x after some finite number of steps. Suppose R_2 accepts x after m steps. We can now consider the following cases:

1. Suppose that R_1 does actually reject x after n steps.
   (a) If $m \leq n$, then we know that R_2 will accept x first. This will cause D to reject x, as it should.

(b) If $m > n$, then we know that R_1 will reject x first. This will cause D to reject x, as it should.

2. Alternately, suppose R_1 never actually rejects x (as in it just loops on x). We know that R_2 will still eventually accept x after m steps. This will cause D to reject x anyway.

Thus, in any case, we can see that if $x \notin L$, D rejects x.

Ultimately, since we've shown D accepts x if $x \in L$ and D rejects x if $x \notin L$, we can see that D does indeed decide the language L. Therefore, if we are given recognizers for L and $\overline{L}$, we can construct a machine D that can decide L. This shows if L and $\overline{L}$ are recognizable, then L must be decidable.

Now, using the conclusions from **(1)** and **(2)**, we can see that a language L is decidable if and only if L and $\overline{L}$ are recognizable. ⊠