CS 341 Automata Theory
Elaine Rich
Homework 14
Due: Tuesday, April 23

This assignment covers Sections 21.5 - 21.7

Note: We have skipped Rice's Theorem. So problems 4 - 5 are here just in case you're interested in learning about it. They are optional.

1) For each of the following languages L, state whether it is in D, SD/D or not SD. Prove your answer. Do not use Rice's Theorem. If you claim that L is not in SD, first prove that it's not in D (for practice), then prove that it's not in SD. Assume that any input of the form $\langle M \rangle$ is a description of a Turing machine.

   a) $\{\langle M \rangle$: TM M accepts exactly two strings and they are of different lengths$\}$.

   Answer. Not $\in$ SD. $\square$

   Proof.
   To prove that this language is not $\in$ D:
   Create a reduction R from $\langle M, w\rangle$ to $\langle M \rangle$ as follows.

   1. Create a description $\langle M_{\#} \rangle$ of a machine $M_{\#}$ that does:
      1.1 Erase the tape.
      1.2 Write w on the tape.
      1.3 Run M on w.
      1.4 If x is `a` or `aa` accept.
   2. R returns $M_{\#}$

   Assume there is a machine $Oracle(\langle M \rangle)$ that decides this language. Then for any language-string pair $\langle M, w\rangle$ consider $C = oracle(R(\langle M, w\rangle))$.

   If $\langle M, w\rangle \in$ H: $M_{\#}$ accepts exactly two strings. Oracle accepts. If $\langle M, w\rangle \notin$ H: $M_{\#}$ accepts no strings. Oracle rejects.
   Therefore C decides H so $Oracle$ must not exist.

To prove that this language is not $\in$ SD:

Create a reduction R from $\langle M, w\rangle$ to $\langle M\rangle$ as follows.

1. Create a description $\langle M_\#\rangle$ of a machine $M_\#$ that does:
   1.1 If x is `a` or `aa` accept.
   1.2 Erase the tape.
   1.3 Write w on the tape.
   1.4 Run M on w.
   1.5 Accept.
2. R returns $M_\#$

Assume there is a machine $Oracle(\langle M\rangle)$ that decides this language. Then for any language-string pair $\langle M, w\rangle$ consider $C = oracle(R(\langle M, w\rangle))$.

If $\langle M, w\rangle \in \neg$ H: $M_\#$ Accepts exactly two strings and loops on all others. Oracle accepts.
If $\langle M, w\rangle \notin \neg$ H: $M_\#$ Accepts all strings. Oracle rejects.

Therefore C semi-decides $\neg$H so *Oracle* must not exist. □

b) $\{\langle M, x, y\rangle : M \text{ accepts } xy\}$.

Answer. In SD but not in D. □

Proof.
Let R be a reduction from $\langle M, w\rangle$ to $\langle M, x, y\rangle$ as follows:

1. Create a description $\langle M_\#\rangle$ of a machine $M_\#$ that does:
   1.1 Erase the tape.
   1.2 Write w on the tape.
   1.3 Run M on w.
   1.4 Accept.
2. Return $\langle M_\#, \epsilon, \epsilon\rangle$

Assume by way of contradiction that there exists some machine $Oracle(\langle M, x, y\rangle)$ that decides whether M accepts xy. Then for any machine description string pair $\langle M, w\rangle$ let $C = oracle(R(\langle M, w\rangle))$. Now there are two cases:

If $\langle M, w\rangle \in$ H: $M_\#$ accepts all strings, so it accepts xy. Oracle accepts.
If $\langle M, w\rangle \notin$ H: $M_\#$ accepts no strings, so it does not accept xy. Oracle rejects.

So C decides H. Therefore *Oracle* does not exist. □

c) $\{\langle M \rangle$: Turing machine M accepts all even length strings$\}$.

Answer. Not $\in$ SD. □

Proof.
This is the proof that this language is not in D:

Let R be a reduction from $\langle M, w \rangle$ to $\langle M \rangle$ as follows:

1. Create a description $\langle M_\# \rangle$ of a machine $M_\#$ that does:
   1.1 Erase the tape.
   1.2 Write w on the tape.
   1.3 Run M on w.
   1.4 Accept.
2. Return $\langle M_\# \rangle$

Assume by way of contradiction that there exists some machine $Oracle(\langle M \rangle)$ that decides whether M accepts all even length strings. Then for any machine description string pair $\langle M, w \rangle$ let $C = oracle(R(\langle M, w \rangle))$. Now there are two cases:

If $\langle M, w \rangle \in$ H: $M_\#$ accepts all strings, so it accepts all even length strings. Oracle accepts.
If $\langle M, w \rangle \notin$ H: $M_\#$ accepts no strings, so it does not accept all even length strings. Oracle rejects.

So C decides H. Therefore $Oracle$ does not exist.

Now the proof that this language is not in SD:
Let R be a reduction from $\langle M, w \rangle$ to $\langle M \rangle$ as follows:

1. Create a description $\langle M_\# \rangle$ of a machine $M_\#(x)$ that does:
   1.1 Copy the input x onto a second tape.
   1.2 Erase the tape.
   1.3 Write w on the tape.
   1.4 Run M on w for $|x|$ steps.
   1.5 If M did not halt naturally, accept.
2. Return $\langle M_\# \rangle$.

Assume by way of contradiction that there exists some machine $Oracle(\langle M \rangle)$ that decides whether M accepts all even length strings. Then for any machine description string pair $\langle M, w \rangle$ let $C = oracle(R(\langle M, w \rangle))$. Now there are two cases:

If $\langle M, w \rangle \in \neg$ H: $M_\#$ accepts all strings, so it accepts all even length strings. Oracle accepts.
If $\langle M, w \rangle \notin \neg$ H: $M_\#$ does not accept on even length strings of length less than $|x|$. Oracle rejects.

So C decides $\neg$ H. Therefore $Oracle$ does not exist. □

d) $\{\langle M \rangle : M$ rejects exactly three strings that start with $\texttt{a}\}$

Answer. Not $\in$ SD. □

Proof.
This is the proof that this language is not in D:

Let R be a reduction from $\langle M, w \rangle$ to $\langle M \rangle$ as follows:

1. Create a description $\langle M_\# \rangle$ of a machine $M_\#$ that does:
   1.1 Copy x to a second tape.
   1.2 Erase the tape.
   1.3 Write w on the tape.
   1.4 Run M on w.
   1.5 If x is `a` or `aa` or`aaa`, reject.
   1.6 Accept.
2. Return $\langle M_\# \rangle$

Assume by way of contradiction that there exists some machine $Oracle(\langle M \rangle)$ that decides this language. Then for any machine description string pair $\langle M, w \rangle$ let $C = oracle(R(\langle M, w \rangle))$. Now there are two cases:

If $\langle M, w \rangle \in$ H: $M_\#$ rejects exactly three strings that start with `a` Oracle accepts.
If $\langle M, w \rangle \notin$ H: $M_\#$ rejects no strings. Oracle rejects.

So C decides H. Therefore *Oracle* does not exist.

Now the proof that this language is not in SD:
Let R be a reduction from $\langle M, w \rangle$ to $\langle M \rangle$ as follows:

1. Create a description $\langle M_\# \rangle$ of a machine $M_\#(x)$ that does:
   1.1 If x is `a` or `aa` or`aaa`, reject.
   1.2 Erase the tape.
   1.3 Write w on the tape.
   1.4 Run M on w.
   1.5 Reject.
2. Return $\langle M_\# \rangle$.

Assume by way of contradiction that there exists some machine $Oracle(\langle M \rangle)$ that semi-decides the language. Then for any machine description string pair $\langle M, w \rangle$ let $C = oracle(R(\langle M, w \rangle))$. Now there are two cases:

If $\langle M, w \rangle \in \neg$ H: $M_\#$ rejects exactly three strings that start with `a`. Oracle accepts.
If $\langle M, w \rangle \notin \neg$ H: $M_\#$ rejects all strings. Oracle rejects.

So C decides $\neg$ H. Therefore *Oracle* does not exist. □

e) $\{\langle M_a, M_b\rangle : L(M_a) - L(M_b) = \emptyset\}$.

Answer. Not $\in$ SD. □

Proof.
This is the proof that this language is not in D:

Let R be a reduction from $\langle M, w\rangle$ to $\langle M\rangle$ as follows:

1. Create descriptions $\langle M_1\rangle$ and $\langle M_1\rangle$ of machines M_1 and M_2 where M_1 always accepts and M_2 does:

    1.1 Erase the tape.
    1.3 Write w on the tape.
    1.4 Run M on w.
    1.5 Accept.

2. Return $\langle M_1, M_2\rangle$

Assume by way of contradiction that there exists some machine $Oracle(\langle M\rangle)$ that decides this language. Then for any machine description string pair $\langle M, w\rangle$ let $C = oracle(R(\langle M, w\rangle))$. Now there are two cases:

If $\langle M, w\rangle \in$ H: M_2 accepts everyhing and $L(M_1) - L(M_2) = \emptyset$. Oracle accepts. If $\langle M, w\rangle \notin$ H: M_2 accepts nothing and $L(M_1) - L(M_2) = \Sigma^*$. Oracle rejects.
So C decides H. Therefore *Oracle* does not exist.

Now the proof that this language is not in SD:
Let R be a reduction from $\langle M, w\rangle$ to $\langle M\rangle$ as follows:

1. Create descriptions $\langle M_1\rangle$ and $\langle M_1\rangle$ of machines M_1 and M_2 where M_2 always rejects and M_1 does:

    1.1 Erase the tape.
    1.2 Write w on the tape.
    1.3 Run M on w.
    1.4 Accept.

2. Return $\langle M_1, M_2\rangle$

Assume by way of contradiction that there exists some machine $Oracle(\langle M\rangle)$ that semi-decides the language. Then for any machine description string pair $\langle M, w\rangle$ let $C = oracle(R(\langle M, w\rangle))$. Now there are two cases:

If $\langle M, w\rangle \in \neg$ H: M_1 accepts nothing and $L(M_1) - L(M_2) = \emptyset$. Oracle accepts.
If $\langle M, w\rangle \notin \neg$ H: M_2 accepts everything and $L(M_1) - L(M_2) = \Sigma^*$. Oracle rejects.

So C decides $\neg$ H. Therefore *Oracle* does not exist. □

2) Prove that TM_{REG} is not in SD.

Proof.
Let R be a reduction from $\langle M, w\rangle$ to $\langle M\rangle$ as follows:

1. Create a description $\langle M_\#\rangle$ of a machine $M_\#(x)$ that does:
   1.1 Write x to the second tape.
   1.2 Erase the tape.
   1.3 Write w on the tape.
   1.4 Run M on w.
   1.5 If $x \in a^n b^n$ accept.
2. Return $\langle M_\#\rangle$.

Assume by way of contradiction that there exists some machine $Oracle(\langle M\rangle)$ that semi-decides the language. Then for any machine description string pair $\langle M, w\rangle$ let $C = oracle(R(\langle M, w\rangle))$. Now there are two cases:

If $\langle M, w\rangle \in \neg$ H: $L(M_\#) = \emptyset$ which is regular. Oracle accepts.
If $\langle M, w\rangle \notin \neg$ H: $L(M_\#) = a^n b^n$ which is regular. Oracle rejects.

So C decides $\neg$ H. Therefore $Oracle$ does not exist. □

3) For any nonempty alphabet Σ, let L be any decidable language other than $\emptyset$ or Σ^*. Prove that $L \leq_M \neg L$.

Proof. □

4) * Do the other half of the proof of Rice's Theorem, i.e., show that the theorem holds if $P(\emptyset) = True$. (Hint: use a reduction that is not a mapping reduction.)

Proof. □

5) * Use Rice's Theorem to prove that $\{\langle M\rangle$: Turing machine M accepts at least two odd length strings$\}$ is not in D.

Proof. □