COMS 4236: Computational Complexity (Fall 2018)
Problem Set #1

Wenbo Gao - wg2313@columbia.edu

November 2, 2018

Problem 1

(Problem 3.14 on page 149 of TC (8 points)) Show that the collection of decidable languages is closed under the operations of **a)** union. **b)** concatenation. Given two languages L and L', their concatenation is the following language L^*: a string $\mathbf{w} = w_1 \dots w_2$ is in L^* if there exists an i such that $w_1 \dots w_i \in L$ and $w_{i+1} \dots w_n \in L'$. **c)** complementation. **d)** intersection.

(Xi: I removed the operation of "star" from the problem but its solution is similar to that of "concatenation".)

a) union

Solution. Let $\mathbb{M}$ be the set of all Turing machines and
$\mathbb{L} = \{L : \forall L \in \Sigma^*, \exists M \in \mathbb{M}, L = L(M)\}$ be the set of all decidable languages.

$$\forall L_1 \in \mathbb{L}, \exists M_1 \in \mathbb{M}, L_1 = L_1(M_1).$$
$$\Rightarrow M_1(w) = \begin{cases} 1; \text{ if } w \in L_1 \\ 0; \text{ if } w \in \Sigma^*/L_1 \end{cases}$$
$$\forall L_2 \in \mathbb{L}, \exists M_2 \in \mathbb{M}, L_2 = L_2(M_2).$$
$$\Rightarrow M_2(w) = \begin{cases} 1; \text{ if } w \in L_2 \\ 0; \text{ if } w \in \Sigma^*/L_2 \end{cases}$$

Let $L = L_1 \cup L_2 = \{w : \forall w \in \Sigma^*, w \in L_1 \vee w \in L_2\}$ be the union of L_2 and L_2.

construct (Turing machine D)**.**

$$D(w) = \begin{cases} 1; \text{ if } M_1(w) = 1 \vee M_2(w) = 1 \\ 0; \text{ otherwise} \end{cases} = \begin{cases} 1; \text{ if } w \in L_1 \vee w \in L_2 \\ 0; \text{ otherwise} \end{cases} = \begin{cases} 1; \text{ if } w \in L \\ 0; \text{ if } w \in \Sigma^*/L \end{cases}$$

Thus, $L = L(D) \Rightarrow L \in \mathbb{L} \Rightarrow \mathbb{L}$ is closed under union.

b) concatenation

Solution. Let $\mathbb{M}$ be the set of all Turing machines and
$\mathbb{L} = \{L : \forall L \in \Sigma^*, \exists M \in \mathbb{M}, L = L(M)\}$ be the set of all decidable languages.

$$\forall L_1 \in \mathbb{L}, \exists M_1 \in \mathbb{M}, L_1 = L_1(M_1).$$
$$\forall L_2 \in \mathbb{L}, \exists M_2 \in \mathbb{M}, L_2 = L_2(M_2).$$

Let $L = \{\mathbf{w} : \mathbf{w} = (w_1, \ldots, w_n), \exists i \in \mathbb{N}, (w_1, \ldots, w_i) \in L_1 \land (w_{i+1}, \ldots, w_n) \in L_2\}$ be the concatenation of L_1 and L_2.

construct (Turing machine D)**.**

```
$D(w) =$
    let
      $n = |w|$, and initialize variable $b = 0$ (false)
    in
      $i \leftarrow [0, 1, 2, \ldots, n]$ (try all possible break points)
        $b = b \lor (M_1(w_1 \ldots w_i) \land M_2(w_{i+1} \ldots w_n))$
      return $b$
```

Thus, $L = L(D) \Rightarrow L \in \mathbb{L} \Rightarrow \mathbb{L}$ is closed under concatenation.

c) complementation

Solution. Let $\mathbb{M}$ be the set of all Turing machines and
$\mathbb{L} = \{L : \forall L \in \Sigma^*, \exists M \in \mathbb{M}, L = L(M)\}$ be the set of all decidable languages.

$$\forall L_1 \in \mathbb{L}, \exists M_1 \in \mathbb{M}, L_1 = L_1(M_1).$$
$$\Rightarrow M_1(w) = \begin{cases} 1; \text{ if } w \in L_1 \\ 0; \text{ if } w \in \Sigma^*/L_1 \end{cases}$$

Let $L = \Sigma^*/L_1$ be the complement set of L_1.

construct (Turing machine D)**.**

$$D(w) = \neg M_1(w) = \begin{cases} 1; \text{ if } w \in \Sigma^*/L_1 \\ 0; \text{ if } w \in L_1 \end{cases} = \begin{cases} 1; \text{ if } w \in L \\ 0; \text{ if } w \in \Sigma^*/L \end{cases}$$

Thus, $L = L(D) \Rightarrow L \in \mathbb{L} \Rightarrow \mathbb{L}$ is closed under complement.

d) intersection

Solution. Let $\mathbb{M}$ be the set of all Turing machines and $\mathbb{L} = \{L : \forall L \in \Sigma^*, \exists M \in \mathbb{M}, L = L(M)\}$ be the set of all decidable languages.

$$\forall L_1 \in \mathbb{L}, \exists M_1 \in \mathbb{M}, L_1 = L_1(M_1).$$
$$\Rightarrow M_1(w) = \begin{cases} 1; \text{ if } w \in L_1 \\ 0; \text{ if } w \in \Sigma^*/L_1 \end{cases}$$
$$\forall L_2 \in \mathbb{L}, \exists M_2 \in \mathbb{M}, L_2 = L_2(M_2).$$
$$\Rightarrow M_2(w) = \begin{cases} 1; \text{ if } w \in L_2 \\ 0; \text{ if } w \in \Sigma^*/L_2 \end{cases}$$

Let $L = L_1 \cap L_2 = \{w : \forall w \in \Sigma^*, w \in L_1 \wedge w \in L_2\}$ be the intersection of L_2 and L_2.

construct (Turing machine D)**.**

$$D(w) = \begin{cases} 1; \text{ if } M_1(w) = 1 \wedge M_2(w) = 1 \\ 0; \text{ otherwise} \end{cases} = \begin{cases} 1; \text{ if } w \in L_1 \wedge w \in L_2 \\ 0; \text{ otherwise} \end{cases} = \begin{cases} 1; \text{ if } w \in L \\ 0; \text{ if } w \in \Sigma^*/L \end{cases}$$

Thus, $L = L(D) \Rightarrow L \in \mathbb{L} \Rightarrow \mathbb{L}$ is closed under intersection.

Problem 2

(Problem 3.17 on page 149 of TC (10 points))* Show that single-tape TMs that cannot write on the portion of the tape containing the input string can only decide regular languages.

(Xi: A regular language is a language accepted by a finite-state automaton. Given such a TM, what are the states of your automaton? You will receive 7 points by solving Problem 3.13, an easier version of 3.17.)

Solution. First, let's consider read-only TMs with no access to cells outside input cells.
A read-only TM with its pointer only moving right, M_1, is strictly a finite-state automaton (FSA).
A read-only TM with its pointer moving left and right, M_2, is equivalent to M_1 with a longer input tape which contains the entire path of M_2's pointer from start to halt. (Figure 1)

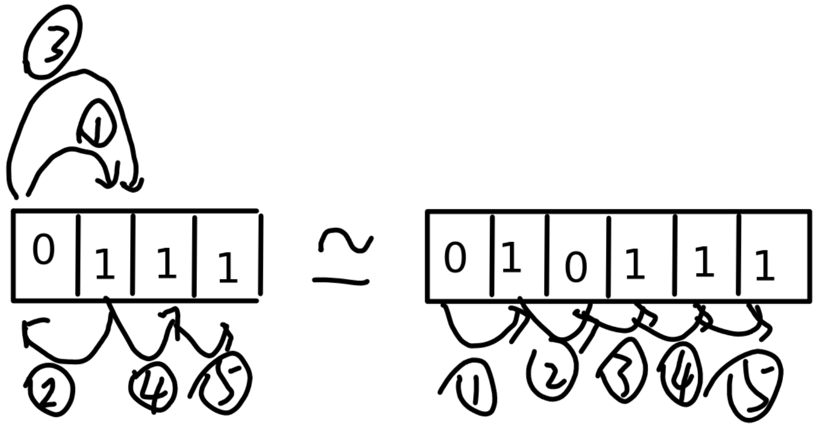

Figure 1: an example

If the read-only TMs access the empty cells outside input cells, there are three new possible termination cases:

1. halt inside input cells
2. halt outside input cells
3. run forever.

The third case doesn't add to the accepted inputs which can be ignored.
For the first two, since operations on empty cells won't add any information to the system but sorely rely on the FSA part of the TM, we can merge the state transitions when pointer is outside input cells to the transitions inside the input cells: Figure 2

Thus, whether the read-only TMs access the empty cells or not, they are equivalent to FSAs, thus they can only decide regular languages.

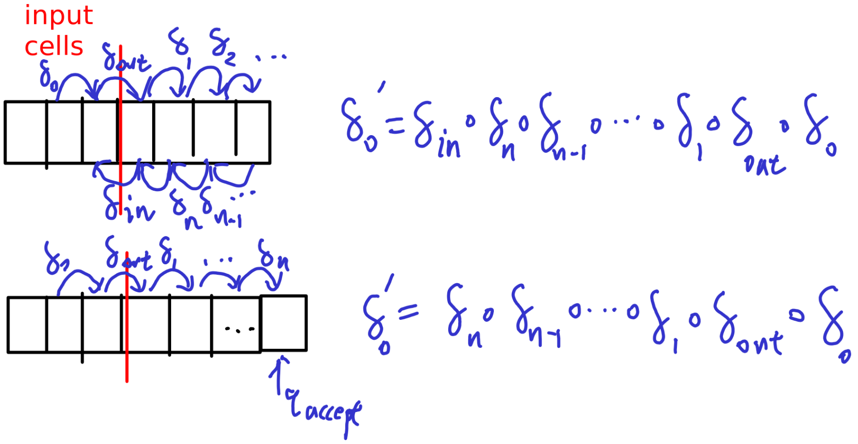
input cells
δ_0
δ_{out}
δ_1 δ_2 ...
δ_{in} δ_n δ_{n-1} ...
$\delta_0' = \delta_{in} \circ \delta_n \circ \delta_{n-1} \circ \cdots \circ \delta_1 \circ \delta_{out} \circ \delta_0$
δ_1 δ_{out} δ_1 ... δ_n
q_{accept}
$\delta_0' = \delta_n \circ \delta_{n-1} \circ \cdots \circ \delta_1 \circ \delta_{out} \circ \delta_0$

Figure 2: upper: the pointer moves back to input cells; lower: case 2 where the TM halts outside input cells.

Problem 3

(Problem 3.4.1 on page 66 of CC (12 points)) For each of the following problem involving Turing machines, determine whether or not it is decidable: **a)** Given a Turing machine M, does it halt on the empty string? **b)** Given a Turing machine M, is there a string which it halts? **c)** Given a Turing machine M, is $L(M)$ finite? (Here $L(M)$ denotes the set of strings $\mathbf{w}$ which $M(\mathbf{w})$="yes".) **d)** Given two Turing machines M and M', is $L(M) = L(M')$? (Xi: To prove a problem is decidable, show that if there is a TM that decides it, then one can construct a TM to decide the halting problem or*a problem that you have already shown to be undecidable.*)

a) EmptyHalt

Solution. Let $\mathbb{M}$ be the set of all Turing machines.

$$\text{Halt} = \{< M, w >: \forall M \in \mathbb{M}, \forall w \in \Sigma^*, M(w) = 1\}$$
$$\text{EmptyHalt} = \{< M >: \forall M \in \mathbb{M}, M(\epsilon) = 1\}$$

reduction from EmptyHalt to Halt:

Given. $< M, w >$, where $M \in \mathbb{M}$ and $w \in \Sigma^*$

construct (Turing machine M' from M with w encoded in its FSA)**.** $\forall$ input $w' \in \Sigma^*$,

1. erase input w' from the tape
2. write w to the tape
3. reset pointer to the beginning of the tape and reset state to q_{start}
4. simulate M on w

Therefore, $\exists M' \in \mathbb{M}, \forall w' \in \Sigma^*, M'(w') = M(w)$ and thus $M'(\epsilon) = M(w)$.
If EmptyHalt is decidable then Halt is decidable. But Halt is not decidable so EmptyHalt is not decidable.

b) Exist

Let $\mathbb{M}$ be the set of all Turing machines.

$$\text{Halt} = \{< M, w >: \forall M \in \mathbb{M}, \forall w \in \Sigma^*, M(w) = 1\}$$
$$\text{Exist} = \{< M >: \forall M \in \mathbb{M}, \exists w \in \Sigma^*, M(w) = 1\}$$

Given. $< M, w >$, where $M \in \mathbb{M}$ and $w \in \Sigma^*$

Using the same construct as in **a)**, we have $\exists M' \in \mathbb{M}, \forall w' \in \Sigma^*, M'(w') = M(w)$.
If Exist is decidable then Halt is decidable. But Halt is not decidable so Exist is not decidable.

c) Finite

Let $\mathbb{M}$ be the set of all Turing machines.

$$\text{Halt} = \{< M, w >: \forall M \in \mathbb{M}, \forall w \in \Sigma^*, M(w) = 1\}$$
$$\text{Finite} = \{< M >: \forall M \in \mathbb{M}, L(M) \text{ is finite}\}$$

Given. $< M, w >$, where $M \in \mathbb{M}$ and $w \in \Sigma^*$

Using the same construct as in **a)** but with its answer flipped, we have

$$\exists M' \in \mathbb{M}, \forall w' \in \Sigma^*, M'(w') = \neg M(w)$$

Therefore, $M(w) = 1 \iff L(M') = \emptyset$ (which is finite). If Finite is decidable then Halt is decidable. But Halt is not decidable so Finite is not decidable.

d) Equal

Let $\mathbb{M}$ be the set of all Turing machines.

$$\text{Halt} = \{< M, w >: \forall M \in \mathbb{M}, \forall w \in \Sigma^*, M(w) = 1\}$$
$$\text{Equal} = \{< M_1, M_2 >: \forall M_1 \in \mathbb{M}, \forall M_2 \in \mathbb{M}, L(M_1) = L(M_2)\}$$

reduction from Equal to Halt:

Given. $< M, w >$, where $M \in \mathbb{M}$ and $w \in \Sigma^*$

construct (Turing machine M_1 and M_2).

$$M_1(w_1) = \begin{cases} 1; \text{ if } w_1 = \epsilon \\ 0; \text{ otherwise} \end{cases}$$

$$M_2(w_2) = \begin{cases} M(w); \text{ if } w_2 = \epsilon \\ 0; \text{ otherwise} \end{cases}$$

Therefore, $M(w) = 1 \iff L(M_1) = L(M_2)$.
If Equal is decidable then Halt is decidable. But Halt is not decidable so Equal is not decidable.

Problem 4

(Problem 3.4.5 on page 67 of CC (10 points)) Show that L is recursive if and only if there is a Turing machine M that enumerates L, and such that the strings in L are output by M in length-increasing fashion. (For the if direction, if L is finite there is nothing to prove. So suppose it is infinite.) (Xi: Read pages 61-62 of CC or pages 140-141 of TC for the direction of a Turing machine enumerating a language L.)

1) $\forall L \subset \Sigma^*, \exists M \in \mathbb{M}, L = L(M) \Rightarrow \exists M' \in \mathbb{M}, L = E(M')$

Proof.

Given. 1. $L \subset \Sigma^*$

2. $M \in \mathbb{M}$ such that $L = L(M)$

construct (Turing machine M' that enumerates L)**.** $\forall$ input $x \in \Sigma^*$,
enumerate Σ^* $(w \leftarrow \Sigma^*)$ in lexicographical order $(0, 1, 10, 11, 100, \dots)$:

1. simulate M on w, if $M(w) = 1$ then output $\sqcup w \sqcup$ at the end of the tape

2. if $w = x$, then accept.

□

2) $\forall L \subset \Sigma^*, \exists M \in \mathbb{M}, L = E(M) \Rightarrow \exists M' \in \mathbb{M}, L = L(M')$

Proof.

Given. 1. $L \subset \Sigma^*$

2. $M \in \mathbb{M}$ such that $L = E(M)$

construct (Turing machine M' that determines L)**.** $\forall$ input $x \in \Sigma^*$,
simulate M on ϵ

1. if the tape ends with $\sqcup x \sqcup$, then accept.

2. if the tape ends with $\sqcup w \sqcup$ where $|w| > |x|$, then reject.

□