COMP 330 Assignment 1

Name: Yuhao Wu
ID Number: 260711365

2018-9-15

Question 1:

Fix a finite alphabet $\sum$ and let $\emptyset \neq L \subseteq \sum^*$. We define the following relation R on words from $\sum^*$:

$$\forall x, y \in \Sigma^*, xRy \text{ if } \forall z \in \Sigma^*, xz \in L \text{ iff } yz \in L$$

Prove that this is an equivalence relation.

Proof.

$$\forall x, y \in \Sigma^*, xRy \text{ if } \forall z \in \Sigma^*, xz \in L \text{ iff } yz \in L$$

As we need to prove this is an equivalence relation, we need to prove that: (1) reflexivity (2) symmetry (3) transitivity

- reflexivity: we need to show $\forall x \in \Sigma^*, xRx$
  $\forall z \in \Sigma^*, xz \in L \Leftrightarrow xz \in L$ (I think this is obvious, no more proof), according the relation R defined in the question, we can imply that xRx
- symmetry: we need to show $\forall x, y \in \Sigma^*, xRy \implies yRx$

$$\left[\forall z \in \Sigma^*, xz \in L \Leftrightarrow yz \in L\right] \implies \left[\forall z \in \Sigma^*, yz \in L \Leftrightarrow xz \in L\right]$$
$$xRy \implies yRx$$

- transitivity: we need to show $\forall x, y, z \in \Sigma^*, xRy, yRz \implies xRz$
  This is equivalent to if we have:

$$(1)\left[\forall w \in \Sigma^*, xw \in L \Leftrightarrow yw \in L\right] \qquad (2)\left[\forall w \in \Sigma^*, yw \in L \Leftrightarrow zw \in L\right]$$
$$\text{could we imply that: } \left[\forall w \in \Sigma^*, xw \in L \Leftrightarrow zw \in L\right]$$

Suppose w is some word in Σ^* and suppose that $xw \in L$, then (1) tells us that $yw \in L$. If $yw \in L$, (2) tells us that $zw \in L$. Thus $xw \in L \implies zw \in L$.
Suppose w is some word in Σ^* and suppose that $zw \in L$, then (2) tells us that $yw \in L$. If $yw \in L$, (1) tells us that $xw \in L$. Thus $zw \in L \implies xw \in L$.
Thus, we can say that:

$$(1)\left[\forall w \in \Sigma^*, xw \in L \Leftrightarrow yw \in L\right] \qquad (2)\left[\forall w \in \Sigma^*, yw \in L \Leftrightarrow zw \in L\right] \implies \left[\forall w \in \Sigma^*, xw \in L \Leftrightarrow zw \in L\right]$$
$$\forall x, y, z \in \Sigma^*, xRy, yRz \implies xRz$$

□

Question 2:

Consider, pairs of natural numbers $< m, n >$ where $m, n \in \mathbb{N}$. We order them by the relation
$< m, n > \sqsubseteq < m', n' >$ if $m < m'$ or $(m = m') \wedge (n \le n')$, where $\le$ is the usual numerical order.
(1): Prove that the relation $\sqsubseteq$ is a partial order.
(2): Prove that the relation $\sqsubseteq$ is a well-founded order.

(1):

Proof. We need to prove (1) reflexive, (2) antisymmetric, (3) transitive.

- To begin with, we need to prove that $\forall s =< m, n >, s \sqsubseteq s$
  Obviously, we have $(m = m) \wedge (n \le n)$ which implies that $s \sqsubseteq s$
- Then, we need to prove that $\forall s =< m, n >$ and $t =< m', n' >, s \sqsubseteq t$ and $t \sqsubseteq s \implies s = t$
  If $s \sqsubseteq t$, then we say that $m < m'$ or $(m = m') \wedge (n \le n')$
  If $t \sqsubseteq s$, then we say that $m' < m$ or $(m' = m) \wedge (n' \le n)$
  For $s \sqsubseteq t$, if we choose $m < m'$, we will find it is impossible to conclude $t \sqsubseteq s$. It is the same for $t \sqsubseteq s$.

$$\begin{aligned} & s \sqsubseteq t \text{ and } t \sqsubseteq s \\ \implies & \Big[(m = m') \wedge (n \le n')\Big] \wedge \Big[(m' = m) \wedge (n' \le n)\Big] \\ \implies & (m = m') \wedge (n \le n') \wedge (n' \le n) \\ \implies & (m = m') \wedge (n = n') \\ \implies & s = t \end{aligned}$$

- In the last, we need to prove that

$$\forall s =< m, n >, t =< m', n' >, \text{ and } r =< m'', n'' > s \sqsubseteq t \text{ and } t \sqsubseteq r \implies s \sqsubseteq r$$

  If $s \sqsubseteq t$, then we say that $m < m'$ or $(m = m') \wedge (n \le n')$
  If $t \sqsubseteq r$, then we say that $m' < m''$ or $(m' = m'') \wedge (n' \le n'')$

  For $s \sqsubseteq t$, if $m < m'$: as for $t \sqsubseteq r, m' \le m''$ must hold, we can conclude that $m < m''$, which tells us that $s \sqsubseteq t$

  For $s \sqsubseteq t$, if $(m = m') \wedge (n \le n')$: as for $t \sqsubseteq r$:
  (1)if $m' < m''$, then we can conclude that: $m < m''$, which tells us $s \sqsubseteq r$
  (2)if $(m' = m'') \wedge (n' \le n'')$, then we can conclude that: $(m = m'') \wedge (n \le n'')$, which tells us $s \sqsubseteq r$

  So, we can conclude that:

$$\forall s =< m, n >, t =< m', n' >, \text{ and } r =< m'', n'' > s \sqsubseteq t \text{ and } t \sqsubseteq r \implies s \sqsubseteq r$$

Thus, we can say that the relation $\sqsubseteq$ is a partial order. □

(2):

Proof. Suppose that all the pairs of natural numbers $< m, n >$ where $m, n \in \mathbb{N}$ make up the set W.

We want to prove that the relation $\sqsubseteq$ is a well-founded order, which is to prove that:

For every non-empty subset U of $W, (U \neq \emptyset \quad \&\& \quad U \subseteq W), \exists u_0 \in U$ such that u_0 is minimal in U.

Suppose that there are k elements in the set U. As $(\mathbb{N}, \leq)$ is well-founded, we can always find the minimal m_0 among the k pairs.

It is possible that there are j pairs of $< m, n >$, where $m_1 = m_2 = \ldots = m_j =$ minimal among the k elements

As $(\mathbb{N}, \leq)$ is well-founded, we can always find the minimal n_0 among the j pairs.

In this case, we can say that $u_0 =< m_0, n_0 >$ is the minimal among all the k elements in the set U.

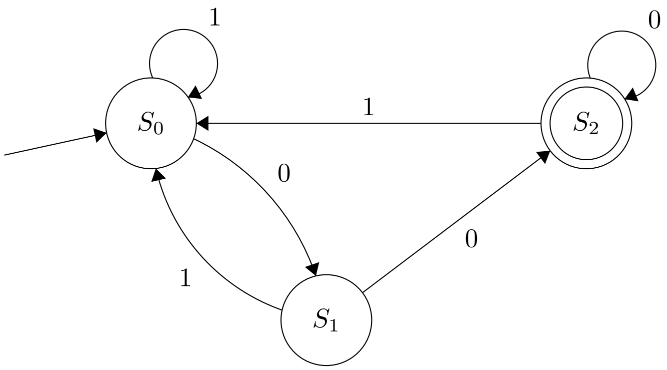
Suppose $u_i =< m_i, n_i >$ is any element among the k elements.
Due to the way we construct u_0, $m_0 \leq m_i$. If $m_0 < m_i$, according to the definition, we can say $u_0 \sqsubseteq u_i$
If $m_o = m_i$, due to the way we construct u_0, $n_0 \leq n_i$, according to the definition $(m_0 = m_i) \wedge (n_0 \leq n_i)$, we can say $u_0 \sqsubseteq u_i$

Thus, we can say that there is nothing smaller than u_0, so u_0 is minimal in set U.

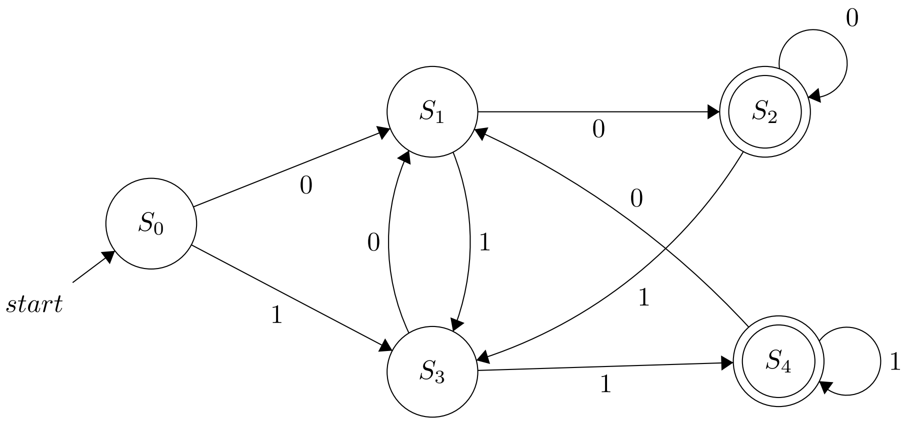
□

Question 3:

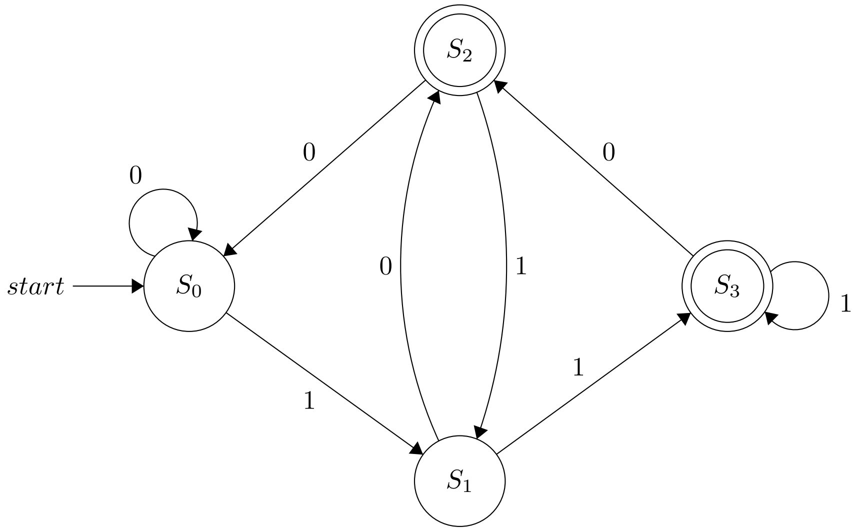
(1):



(2):



(3):



Question 4:

Suppose that we have a language L described by a DFA $D = \{S, \Sigma, \delta, s_0, F\}$.
We can construct a NFA $N = \{S', \Sigma, \delta', s'_0, F'\}$ which recognize LeftHalf:

- $S' = S \times S$ (Cartesian Product)
- $s'_0 = (s_0, f)$ $(f \in F)$
- $F' = \{(s, s) | s \in S\}$
- $\delta'((s, t), a) = (\delta(s, a), t')$ where state t' can move to t with at least one single letter in the alphabet

EXPLAINATION:

- The new state of the NFA is a Cartesian Product of the states of DFA, the first coordinate is for w_1 in question and the second is for w_2
- The start state for the first coordinate is s_0, which is the start state of DFA, The start state for the first coordinate is $f \quad f \in F$. We want them to go from two directions and ends in the same state. As they both move the same number of steps, $|w_1| = |w_2|$, and as they end in the same state, which means $w_1 w_2 \in L$
- $\delta'((s, t), a) = (\delta(s, a), t') \quad a \in \Sigma$ where state t' can move to t with at least one letter in the alphabet When we read a letter a from w_1, we move s to $\delta(s, a)$ and we move t back to t'. NOTE: we don't need to know which letter or which path we need to choose, we just need to determine if there is a path so that both s and t can go to the same state. As in the definition, we just need to make sure there **exists** w_2.

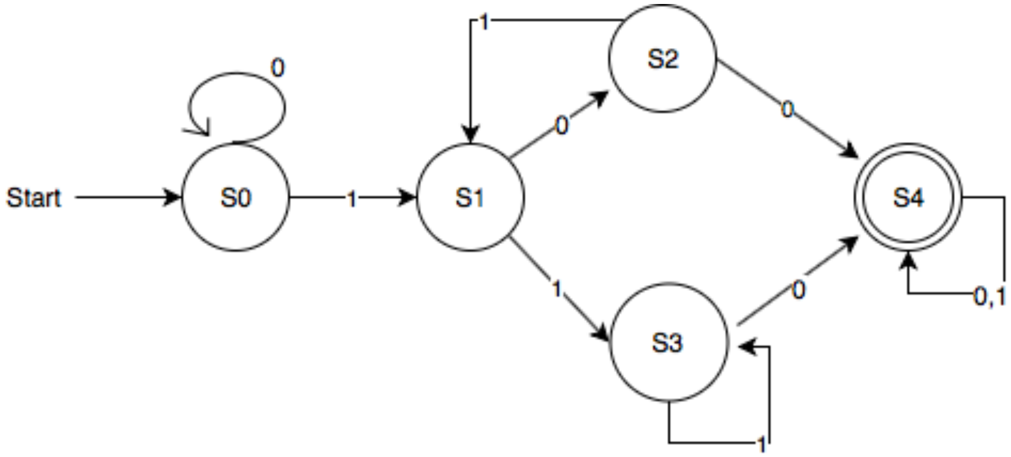
Since a NFA can recognize a regular language, thus, we can say that LeftHalf(L) is regular.

Question 5:

(1): Give a deterministic finite automaton accepting the following language over the alphabet $\{0, 1\}$: The set of all words containing 100 or 110.

(2): Show that any DFA for recognizing this language must have at least 5 states.

(1):



- S_0 means the last three bits are 000
- S_1 means the last three bits are 001, 101
- S_2 means the last three bits are 010, when the next is 1, as the automata can't recognize something like 101, so have to go back to S_1
- S_3 means the last three bits are 011, 111, when the next is 1, have to go back to itself
- S_4 means the word contains 100 or 110, since the automata just wants to recognize words containing 100 or 110, when reading more letters, it stays in S_4, which is our accept state.

(2):

Proof. As I have already draw a DFA for Question1 using 5 states, what we need to do is to prove that it is impossible for any DFA recognizing this language having only 4 states.

As the purpose of the automata is to recognize words containing 100 or 110, which means the automata at least needs to store 3 bits, otherwise how could it be able to recognize 3 bits words.

All the possible permutation of 3 bits words is below:

$$\{000, 001, 010, 011, 100, 101, 110, 111\}$$

Obviously, we can separate $\{100, 110\}$ from above, as they are accept states and others are reject states.

Then, we can separate $\{010, 011, 111\}$ from above, as they ends in 11 or 10, when the next bit is 0, they can change into accept states.
But, in fact, we have to separate the three elements, as when the following bits is 1, 010 will change into 101, which is not in this sets.

Under this case, we have already had four states:

- $\{100, 110\} \implies$ accepting states
- $\{011, 111\} \implies$ after adding 1, it will still be in this state, after adding 0, it will go to accepting states
- $\{010\} \implies$ after adding 1, it will go to the fourth state, after adding 0, it will go to accepting states
- $\{000, 001, 101\}$ this is the remaining

For the fourth state, when the next coming bit is 0, 000 will stay in the same states, while 001 and 101 will go to the third state, and get a contradiction.
So, we can conclude that any DFA for recognizing this language must have at least 5 states.

□