CS/ECE 374 P09

Junquan Chen, Pengxu Zheng, Jiawei Tang

TOTAL POINTS

80 / 100

QUESTION 1

1 Problem 9.A. 30 / 30

✓ **+ 30 pts** Correct

+ 7.5 pts IDK

+ 0 pts Incorrect

✓ **+ 10 pts** Some version of the observation that if a string of form w01* is in L, then the string w10* would be in INC(L).

✓ **+ 10 pts** Idea of two copies of the DFA with guessing power

✓ **+ 10 pts** Idea of the second copy of DFA having "flipped" transition functions

QUESTION 2

2 Problem 9.B. 30 / 30

+ 30 pts Correct

+ 10 pts Description correct and clear but not formal enough, i.e., no set notation and tuple description

+ 7.5 pts IDK

+ 0 pts Incorrect

✓ **+ 5 pts** Correct Q'

✓ **+ 5 pts** Correct s'

✓ **+ 5 pts** Correct A'

✓ **+ 15 pts** Correct transition functions: transition functions in two copies of the DFA and transition functions connecting the two copies

- 5 pts Missing one case for transition functions; set notation not clear enough for transition function

- 10 pts Defines delta* instead of delta

Description of delta not quite formal enough

QUESTION 3

3 Problem 9.C. 10 / 30

+ 30 pts Correct

+ 7.5 pts IDK

+ 0 pts Incorrect

✓ **+ 15 pts** Correct proof for L(M) \in INC(L)

+ 15 pts Correct proof for INC(L) \in L(M)

✓ **- 5 pts** Hand-wavy on details of one direction of proof

- 10 pts Hand-wavy on details of both directions of proof

I don't think your contradiction is correct. You are trying to show that L(M') is a subset of INC(L). By definition, there can be items not in L(M') that are in INC(L).

QUESTION 4

4 Problem 9.D. 10 / 10

✓ **+ 10 pts** Correct

+ 2.5 pts IDK

+ 0 pts Incorrect

gradescope

HW Solution

CS/ECE 374: Algorithms & Models of Computation, Spring 2019 Version: **1.0**

Submitted by:

- ≪**Jiawei Tang**≫: ≪**jiaweit2**≫
- ≪**Junquan Chen**≫: ≪**junquan2**≫
- ≪**Pengxu Zheng**≫: ≪**pzheng5**≫

9

Solution:

9.A. We are given a DFA M for L. The definition of INC(L) informs that if we want to get a binary string of INC(L), we need to add 1 to the binary string of L. Since we want to construct an NFA M' for INC(L) , then we need to subtract 1 from the binary string input to get a string in L, which is accepted by the provided DFA. The NFA should be able to keep track of the pattern that

(1). if we encounter the last 1, then flip the 1 to zero and flip the remaining $0s$ to $1s$. We label the status as "after" once we encounter the last 1.

(2). if we encounter the 1 that is not the last 1, then we keep the one and label the status as "before" meaning the operation doesn't start.

(3). It's impossible to encounter the 1 when the status has been labeled as "after" as we have already encounter the last 1 as stated in (1).

(4). If we encounter a zero that has been flagged as "before" , then keep the zero and keep the "before" label.

(5). If we encounter a zero that has been flagged as "after", then the zero we encounter is on the right of the last 1, so we need to flipped the 0 to 1.

9.B. Let M = $(\Sigma, Q, s, A, \delta)$ be a DFA that accepts L. We construct an NFA M' = $(\Sigma', Q', s', A', \delta')$ that accepts INC(L) as follows:

$$Q' := Q \times \{before, after\}$$
$$s' := (s, before)$$
$$A' := \{(q, after) \mid q \in A\}$$

$$\delta'((q, before), 0) = \{(\delta(q, 0), before)\}$$
$$\delta'((q, after), 0) = \{(\delta(q, 1), after)\}$$
$$\delta'((q, before), 1) = \{(\delta(q, 1), before), (\delta(q, 0), after)\}$$
$$\delta'((q, after), 1) = \varnothing$$

- The state (q, before) means (the simulation) of M is in state q and M' has not yet flipped the binary digits.

- The state (q, after) means (the simulation) of M is in state q and M' has already flipped the binary digits.

9.C. To prove $INC(L) = L(M')$, we have to prove $INC(L) \subseteq L(M')$ and $L(M') \subseteq INC(L)$. First, we want to prove $INC(L) \subseteq L(M')$, which means $\forall x \in INC(L), x \in L(M')$. Since every string in $INC(L) = \{binary(i+1) | binary(i) \in L\}$ by definition, if there is a number $binary(j) \in INC(L)$, there must be a number $binary(i)$ which is accepted by DFA M, where $i = j - 1$. If given $binary(j)$, to obtain $binary(i)$, we find the last 1 to flip it to 0. We also flip the following 0s to 1s. Also, since the string $binary(i)$ starts with 1 by definition, every string in $INC(L)$ must have at least one 1. Therefore, NFA M' must have 1 as input symbol. When the input is the last 1 in $binary(j)$, we have

$$\delta'((q, before), 1) = \{(\delta(q, 0), after)\}$$

, serving a purpose of transferring to a state with "q,after" label from "q,before" label and flipping 1 to 0. Then we have

$$\delta'((q, after), 0) = \{(\delta(q, 1), after)\}$$

, serving to take 0 as input for the NFA M' but take 1 as input for the DFA M. It is equivalent to flipping the 0s to 1s as mentioned above. Therefore, we can guarantee that the final state $(q, after) \in A'$ because $q \in A$ for the reason that to the DFA M, $binary(i)$ is the input string and it is accepted by M by definition. Therefore $INC(L) \subseteq L(M')$.
Second, we want to prove $L(M') \subseteq INC(L)$, which means $\forall x \in L(M'), x \in INC(L)$. To prove it by contradiction, we claim that if $\exists x \notin L(M'), x \in INC(L)$. Since $x \notin L(M')$, $\delta'((s, before), x) \cap A' = \emptyset$. Assume x=binary(i). It means that $\delta'^*((s, before), x) = \{(q_1, before), (q_2, after)\}$ where $q_1 \in Q, q_2 \notin A$. If after the NFA M' takes the input x, it ends up on state $(q_1, before)$. It means that there is no 1 in x. So $x \notin INC(L)$ as $INC(L)$ is defined to start with 1. If it ends up on the state $(q_2, after)$, by our algorithm(transitions of M') to find the binary(i-1), which is to flip the last 1 to 0 and the following 0s to 1s, it means that the binary(i-1) is not in L since $q_2 \notin L$. So $x = binary(i)$ where $binary(i-1) \notin L$. Therefore, $x \notin INC(L)$. Thus, the claim is false. It concludes that every element in L(M') must also be in $INC(L)$.
Thus, $L(M') = INC(L)$.

9.D.

$$Q' := Q \times \{before, after\}$$

$$s' := (s, before)$$

$$A' := \{(q, after) \mid q \in A\}$$

$$\delta'((q, before), 0) = \{(\delta(q, 0), before)\}$$

$$\delta'((q, after), 0) = \{(\delta(q, 1), after)\}$$

$$\delta'((q, before), 1) = \{(\delta(q, 1), before), (\delta(q, 0), after), (q, after)\}$$

$$\delta'((q, after), 1) = \varnothing$$

Since L contains 1^*, the string accepted by INC(L) would have one extra binary bit, For example, when 111 is accepted by L, then 1000 should be accepted by INC(L). There will be another transition $\delta'((q, before), 1) = (q, after)$ created in the NFA if we are currently in the state $(s, before)$ when a 1 is taken in. In this way, we consume the extra binary bit from input and s stays unchanged.

1 Problem 9.A. 30 / 30

- ✓ + **30 pts** Correct
- \+ **7.5 pts** IDK
- \+ **0 pts** Incorrect
- ✓ + **10 pts** Some version of the observation that if a string of form w01* is in L, then the string w10* would be in INC(L).
- ✓ + **10 pts** Idea of two copies of the DFA with guessing power
- ✓ + **10 pts** Idea of the second copy of DFA having "flipped" transition functions

gradescope

2 Problem 9.B. 30 / 30

+ **30 pts** Correct

+ **10 pts** Description correct and clear but not formal enough, i.e., no set notation and tuple description

+ **7.5 pts** IDK

+ **0 pts** Incorrect

✓ + **5 pts Correct Q'**

✓ + **5 pts Correct s'**

✓ + **5 pts Correct A'**

✓ + **15 pts Correct transition functions: transition functions in two copies of the DFA and transition functions connecting the two copies**

- **5 pts** Missing one case for transition functions; set notation not clear enough for transition function

- **10 pts** Defines delta* instead of delta

Description of delta not quite formal enough

gradescope

3 Problem 9.C. **10 / 30**

+ **30 pts** Correct
+ **7.5 pts** IDK
+ **0 pts** Incorrect

✓ + **15 pts** **Correct proof for L(M) \in INC(L)**

+ **15 pts** Correct proof for INC(L) \in L(M)

✓ - **5 pts** **Hand-wavy on details of one direction of proof**

- **10 pts** Hand-wavy on details of both directions of proof

I don't think your contradiction is correct. You are trying to show that L(M') is a subset of INC(L). By definition, there can be items not in L(M') that are in INC(L).

gradescope

4 Problem 9.D. 10 / 10

✓ + **10 pts** **Correct**

+ **2.5 pts** IDK

+ **0 pts** Incorrect

gradescope