CS/ECE 374 B Fall 2019	Jin Yucheng (yucheng9)
Homework 1 Problem 1	Fan Wenyan (wenyanf2)

Solution:

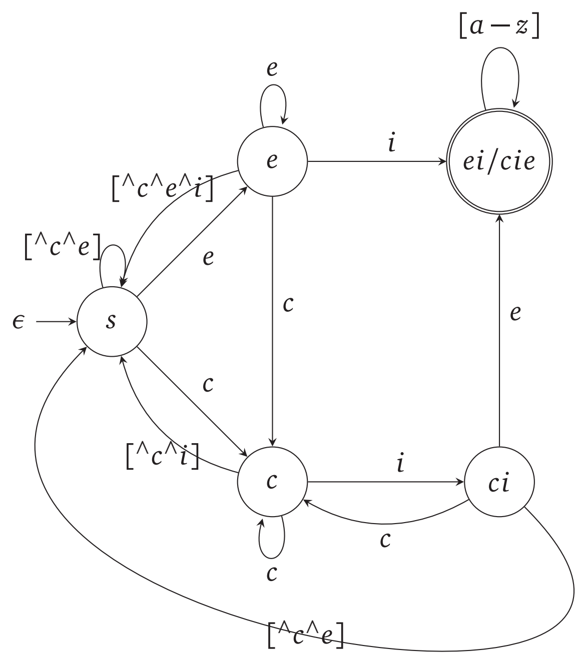
(a) The regular expression of the desired language L is defined as:

- $w \in L$ if w contains substring ei which is **NOT** preceded by c to be cei.
- $w \in L$ if w contains substring cie.

Therefore, the regular expression is:

$$[a-z]^*[\hat{}c]ei[a-z]^* + [\hat{}c]^*ei[a-z]^* + [a-z]^*cie[a-z]^*$$

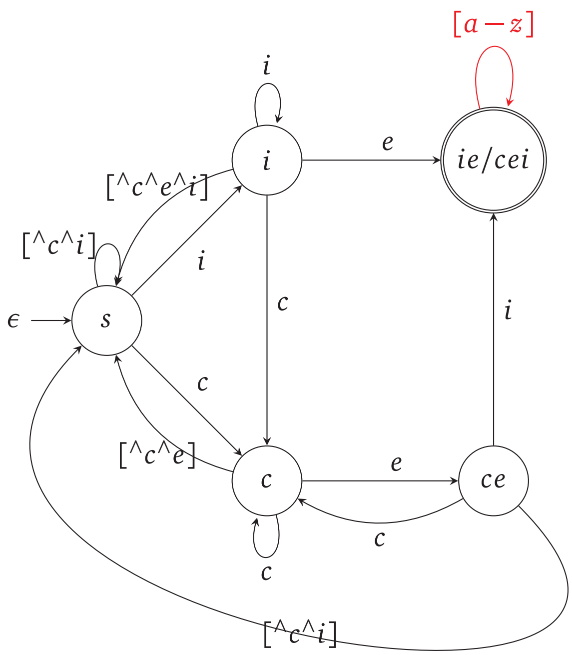
(b) The graphical representation is:



- s is the start state.
- c means we have just read the initial c of cie.
- e means we have just read the initial e of ei.
- ci means we have read the ci part of cie.
- ei/cie means we have read the substring ei (without preceding c) or cie, and it is the accepting state.

This graphical representation captures all the cases, because it traces two possible path, $e->ei$ and $c->ci->cie$, to reach the accepting state. Also it avoids "$c->ce->cei$", because it forces the state c to transit to state s if it receives e.

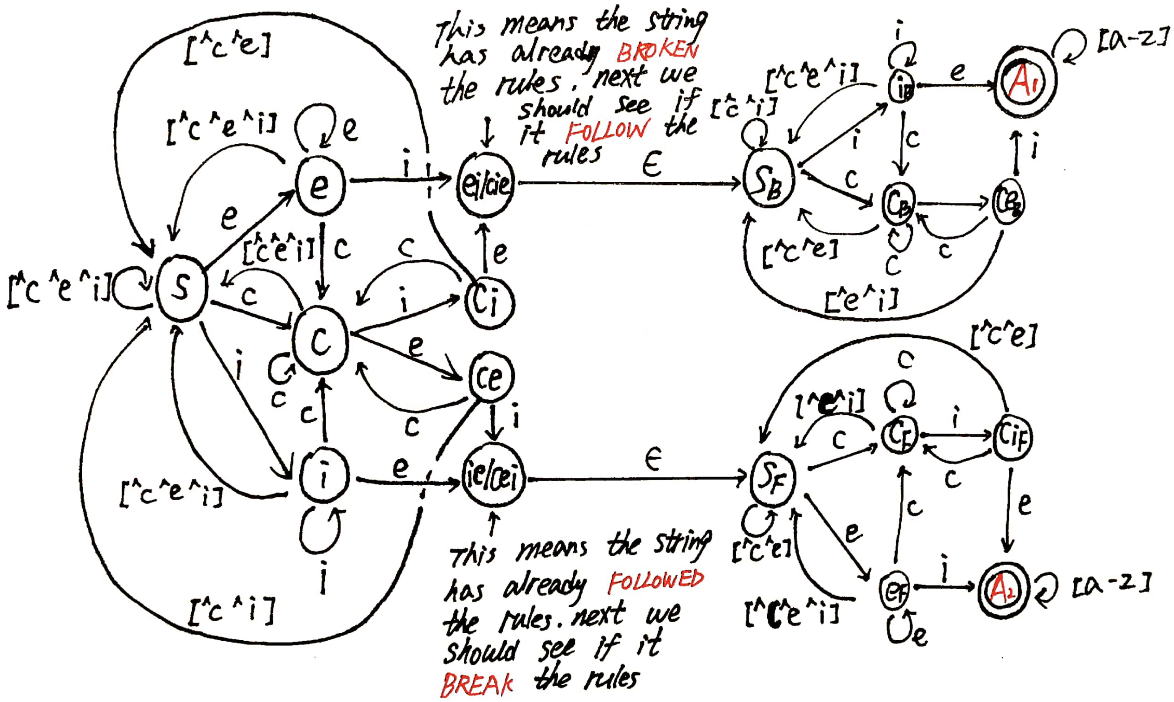
(c) The graphical representation of the DFA that satisfies "all strings that **END UP WITH THE SUBSTRING** ie (without preceding c) or cei (without preceding substring ie (without preceding c) or cei)" is:



- s is the start state.
- c means we have just read the initial c of cei.
- i means we have just read the initial i of ie.
- ce means we have read the ce part of cei.
- ie/cei means we have finally verified that the string ends up with substring ie (without preceding c) or cei without preceding ie (without preceding c) or cei.

A very important thing is that the transition function of this DFA is not a total function, this is shown as we've indicated with a red loop, since we will not discuss the states after we find the first substring ie (without preceding c) or cei.

Then we show how to construct a DFA that "recognizes words that both break the rules and follow it". We can break the definition into two parts: first, the words (strings) break the rules; second, they also follow the rules, and we can see the desired DFA as a "concatenation" of these two parts. The DFA given above in (c) just verifies the words (strings) that have a substring that follows the rules, the DFA in (b) verifies the words (strings) that have a substring that breaks the rules, therefore, we can "concatenate" them together:



Where subscripts B and F in states of the right half represent we have already found the words (strings) that break/follow the rules, A1 and A2 are accepting states where the strings that go into A1 first break the rules, and the strings that go into A2 first follow the rules.

■

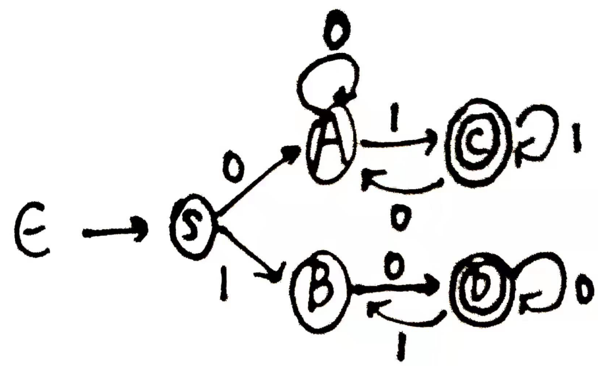
CS/ECE 374 B Fall 2019	Jin Yucheng (yucheng9)
Homework 1 Problem 2	Fan Wenyan (wenyanf2)

Solution:

(a) The regular expression for (a) is:

$$0(0+1)^*1 \ + \ 1(0+1)^*0$$

The DFA is:



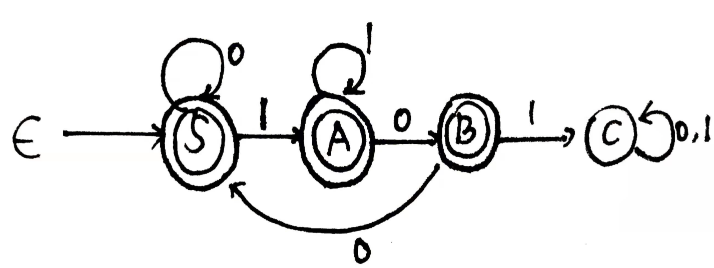
Where,

- S is the start state.
- A means the string starts with 0 but the current last digit is not 1.
- B means the string starts with 1 but the current last digit is not 0.
- C means the string starts with 0 and the last digit is 1.
- D means the string starts with 1 and the last digit is 0.

(b) The regular expression for (b) is:

$$0^*(1^*000^*)^*1^*0^*$$

The DFA is:



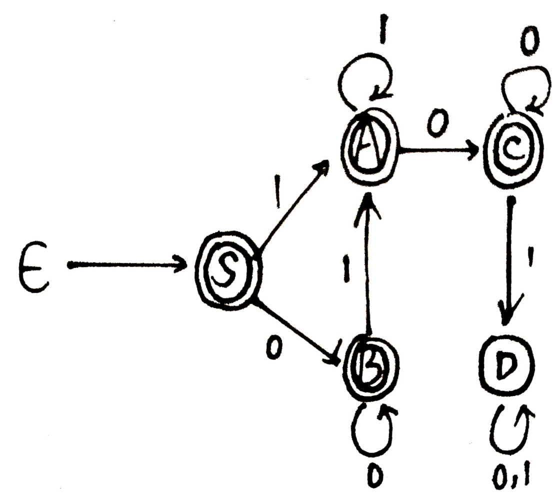
Where,

- S is the start state.
- A means the string currently does not have substring 101 but the current last digit is 1
- B means the string currently does not have substring 101 but the current last-two-digit is 10.
- C means the string has substring 101.

(c) The regular expression for (c) is:

$$0^*1^*0^*$$

The DFA is:



Where,

- S is the start state.
- A means the string currently does not have subsequence 101 but has subsequence 1.
- B means the string currently only has 0's.
- C means the string currently does not have subsequence 101 but has subsequence 10.
- D means the string has subsequence 101.

■

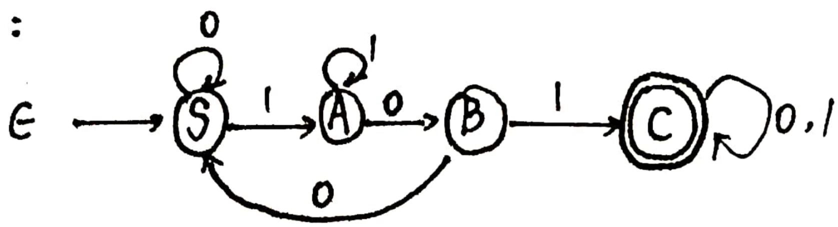
CS/ECE 374 B Fall 2019	Jin Yucheng (yucheng9)
Homework 1 Problem 3	Fan Wenyan (wenyanf2)

Solution: $L(M_1), L(M_2), L(M_3)$ are defined as follows:

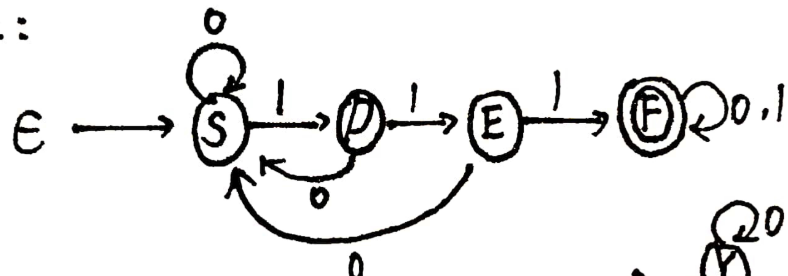
- $L(M_1)$ is the set of all strings that contain substring 101.
- $L(M_2)$ is the set of all strings that contain substring 111.
- $L(M_3)$ is the set of all strings that contain both substrings 101 and 111.

M_1, M_2, and M_3 are represented as:

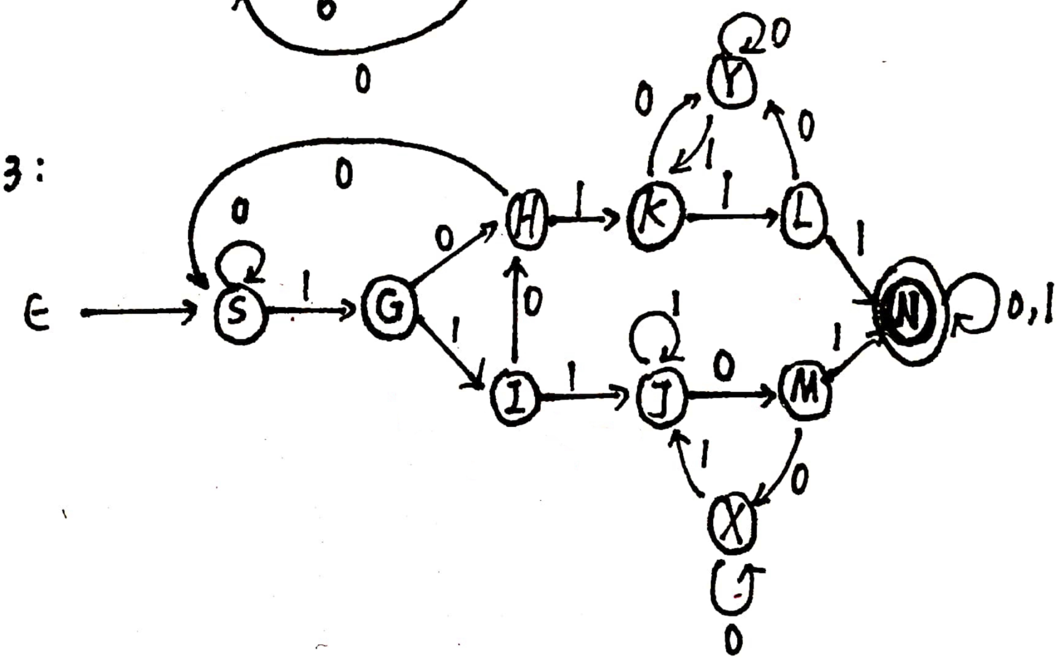
M_1:



M_2:



M_3:



Where in M_1:

- S is the start state.
- A means we have just read the initial 1 of substring 101.
- B means we have read the 10 part of substring 101.
- C means we have found substring 101.

Where in M_2:

- S is the start state.
- D means we have just read the initial 1 of substring 111.
- E means we have read the 11 part of substring 111.
- F means we have found substring 111.

Where in M_3:

- S is the start state.
- G means we have not yet found both substrings 101 and 111 but have just read the initial 1 of substrings 101 and 111.
- H means we have not yet found both substrings 101 and 111 but have just read the initial 10 of substring 111.
- I means we have not yet found both substrings 101 and 111 but have just read the initial 11 of substring 111.
- J means we have not yet found substring 101 but have found substring 111, and we have found the initial 1 of substring 101.
- K means we have not yet found substring 111 but have found substring 101, and we have found the initial 1 of substring 111.
- L means we have not yet found substring 111 but have found substring 101, and we have found the 11 part of substring 111.
- M means we have not yet found substring 101 but have found substring 111, and we have found the 10 part of substring 101.
- N means we have found both substrings 101 and 111.
- X means we have not yet found substring 101 but have found substring 111, and we have to start over to find 111.
- Y means we have not yet found substring 111 but have found substring 101, and we have to start over to find 101.

Obviously, M_1, M_2, and M_3 satisfy all of the points above:

- (a) $|Q_1| = |Q_2| = 4 > 2$.
- (b) M_1 and M_2 use the minimal number of states since the substrings have length 3, and we have to determine them one symbol by one symbol, and there are 4 cases: not match, the first symbol matches, the first two symbols match, all three symbols match.
- (c) By definition $L(M_1) \neq L(M_2)$.
- (d) By definition $L(M_3) = L(M_1) \cap L(M_2)$.
- (e) There are infinite strings that have substrings 101 and 111.
- (f) $|Q_3| = 11 < 16 = |Q_1| \times |Q_2|$

■