Brock Boehler

[bvb2@illinois.edu](mailto:bvb2@illinois.edu)

ECE 374 B Spring 2019

Homework 0

**Problem 1**

**(a)** Strings that break “I before E”:

Strings that break “except after C”:

All strings that break either rules:

**(b)**

*Notation Note:* [^(A + B)] specifies the set of all characters included in not including any given characters A and B

C

C

I

E

E

I

E

[^(C+I)]

[^E]

]

[^(E+I)]

[^(E+C)]

**(b) continued:**

The states are labeled as follows:

* START – we have not seen anything yet or we not have seen the string to include ‘CIE’ or ‘EI’ substrings yet
* C – we have encountered a C
* CI – we have encountered the substring C-I
* CIE – we have encountered the substring C-I-E
* E – we have encountered an E
* EI – we have encountered the substring I-E

**(c)** Define another DFA that recognizes strings that *do* follow the rule:

C

C

E

I

I

E

I

[^(C+E)]

[^I]

]

[^(E+I)]

[^(I+C)]

Where the states similarly follow:

* START – the DFA has not read anything yet or has not begun to read any substrings resembling CEI or IE
* C – the DFA has read an initial C
* CE – the DFA has read the substring CE
* CEI – the DFA has read the substring CEI and recognized it as following the rule
* I – the DFA has read an initial I
* IE – the DFA has read the substring IE and recognized it as following the rule

Now that we have two DFA’s – one for recognizing words that follows the rule and the other one recognizes words that don’t, we can use the product construction to create a new DFA the determines whether a word breaks and follows the rules simultaneously. Let M1 denote the DFA that recognizes words *not* following the rule, and M2 denote the DFA of words that follows the rule, and M3 denote the DFA that is the construction of both.

Notation Notes:

– collection of states found in Mn

– starting state of Mn

– state transition function of Mn. See the above DFA diagrams for transition functions of M1 and M2

– set of accepting states of An

As a simple way of explaining the above DFA, imagine that the two DFA’s are running in parallel and whenever both states are in an accepting state, that means we have found a word that both violates and follows the rule.

**Problem 2**

**(a)** All strings that start and end with a different symbol

The following DFA decides whether or not a string has different start and end character for a binary alphabet

Explanation of States:

* START – we have not read anything yet
* 0X0 – the string starts and ends with 0
* 0X1 – the string starts with 0 and ends with 1
* 1X1 – the string starts with 1 and ends with 1
* 1X0 – the string starts with 1 and ends with 0

0

1

1

1

1

1

0

0

0

0

**(b)** All strings that do not contain the substring 101

Explanation of States:

* START – we have read nothing or what we have read so far does not contain a 101
* 1 – We have read an initial 1
* 10 – We have read an initial 1 followed by a 0
* 101 – We have read a consecutive 1-0-1 and have located our substring

1

1

1

0

0

0



**(c)** All strings that do not contain the *subsequence* 101

Explanation of States:

* START – We have not read anything, or we have not read a 1 yet
* 1 – we have read a single 1
* 10 – we have read a 1 and a 0
* 101 – we have read a 1, 0 and another 1

1

1

1

0

0

0



**Problem 3**

L(M­­1) = Language of all strings containing the substring ‘10’

L(M2) = Language of all strings containing the substring ‘101’

L(M3) = = Language of all strings containing the substring ‘101’

M1

1

0

1

0

0

M2, M3 (Note:

1

0

1

1

0

0

0

M1 Explanation of States:

* START – we have not read anything, or we have not begun reading a ’10’ substring
* 1 – we have read an initial 1
* 10 – we have read a 1 followed immediately by a 0, finding our substring

M2 and M3 Explanation of States:

* START – we have not read anything, or we have not began reading our ‘101’ substring
* 1 – we have read an initial 1
* 10 – we have read a 1, followed by a 0
* 101 – we have read a 1, followed by a 0, followed by a 1, thus finding our substring ‘101’

Justification:

1. M1 has 3 states and M2 has 4 states as shown by above. This satisfies the condition of
2. Both M1 and M­2 use the minimal states, the languages can no be expressed by a DFA of less states
3. L(M1) and L(M2) describe different languages. See the initial description for what each language / DFA describes, the recognize different things.
4. L(M1) recognizes strings that contain ‘10’. L(M2) recognizes strings that contains ‘101’. Because ‘10’ is a substring of ‘101’, any language that contains the substring ‘101’ then also contains the substring ‘10’. Because of this, L(M2) is a subset of L(M­1), which means that:
5. L(M3) is the concatenation of and ‘101’, and because , this means that