CS 512, Spring 2014
Assignment 8

Shan Sikdar

1 Problem 1

Let $\Sigma = \{a, b\}$. Let $L \subseteq \Sigma^*$ be a regular language.
Note: After taking the hints adivce and searching the internet, I used a Paper from Turku Center for Computer Science by Okhotin as a reference.

(a) Since L is regular we know that there exists a NFA $\mathbf{A}$ that recognizes the regular language L.
w.l.o.g $\mathbf{A} = \{S, \Sigma, \delta, Init, Final\}$
(S - set of states, Σ the alphabet, δ the transition function, Init - set if initial states, Final - set of final states).

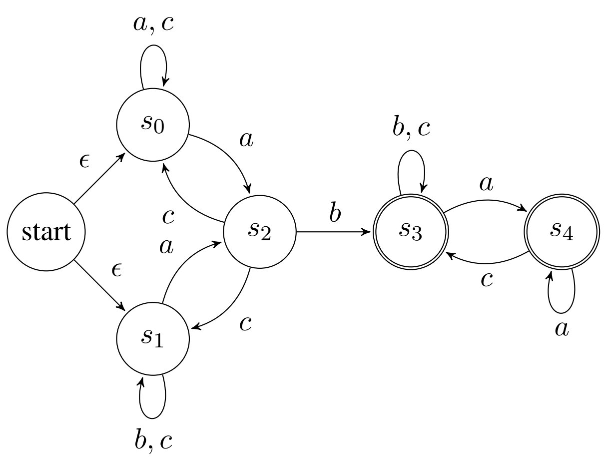
We can extend $\mathbf{A}$ to $\mathbf{A1}$ as follows:
Keep S - the set of states the same.
Keep Σ- the alphabet the same.
Change $Init$ to $\{\delta(q, w)|w \in L, q \in Init\}$
(Basically the new start states are the places you would go to from the original start states after you apply $w \in L$)
Change Final to $\{q' \in S|\exists w \in L : \delta(q', w) \in Final\}$
(Basically the new set of final states will be any state you get when you apply w to an element from the original set of final states)
Change δ to $\delta'(q, x) = \{\delta(q, wx)|w \in L, q \in S\}$
So the new transition state is :
$\mathbf{A1} = \{S, \Sigma, \{\delta(q, wx)|w \in L, q \in S\}, \{\delta(q, w)|w \in L, q \in Init\}, \{q' \in S|\exists w \in L\delta(q', w) \in Final\}\}$
Since this NFA recognizes $L_\triangleleft \triangleq \{u \in \Sigma^*|$there is a word $w \in L$ such that u $\triangleleft$ w$\}$, $L_\triangleleft$ is a regular language.

(b) Since L is regular we know that there exists a NFA $\mathbf{A}$ that recognizes the regular language L.
w.l.o.g $\mathbf{A} = \{S, \Sigma, \delta, Init, Final\}$
(S - set of states, Σ the alphabet, δ the transition function, Init - set if initial states, Final - set of final states).

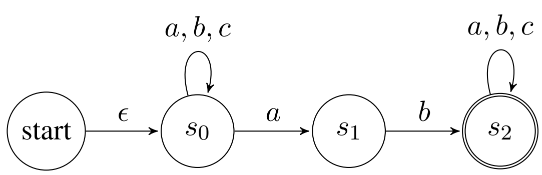
We can extend $\mathbf{A}$ to $\mathbf{A2}$ as follows:
Keep S - the set of states the same.
Keep Σ- the alphabet the same.
Keep $Init$
Keep Final
Change δ to $\delta'(q, x) = \{\delta(q, x), x\}$
(Basically extend the transition system to have the ability to go to the state it was going to go initially, or add a self loop to that state.)
So essentially the NFA for Language L has been extended to have self loops in the transition system. So the new transition state is :
$\mathbf{A2} = \{S, \Sigma, \{\delta(q, x), x\}, Init, Final\}\}$
Since this NFA recognizes $L_{\triangleright} \triangleq \{u \in \Sigma^* |$there is a word $w \in L$ such that w $\triangleleft$ u$\}$, $L_{\triangleright}$ is a regular language.

2 Problem 2

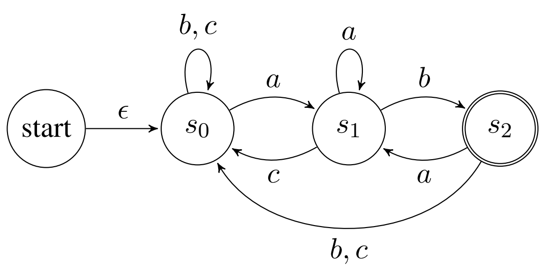
(a) w contains ab exactly once.



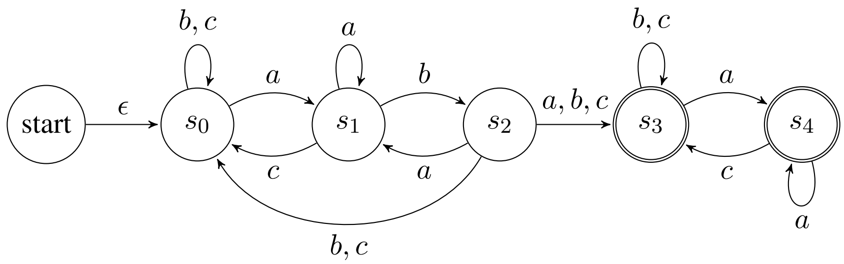
(b) w contains ab atleast once.



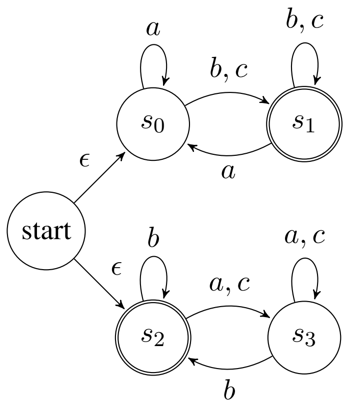
(c) w contains ab infinitely often.



(d) w contains ab finitely often only. In this case, we need to add accepting states that can parse all possible strings over alphabets except ab to the final state of (c). The states s_1 and s_2 take care of the finitely looping part of ab. Hence, our new NBA will look like :



(e) if w contains infinitely often as then it contains infinitely often bs.



3 Problem 3

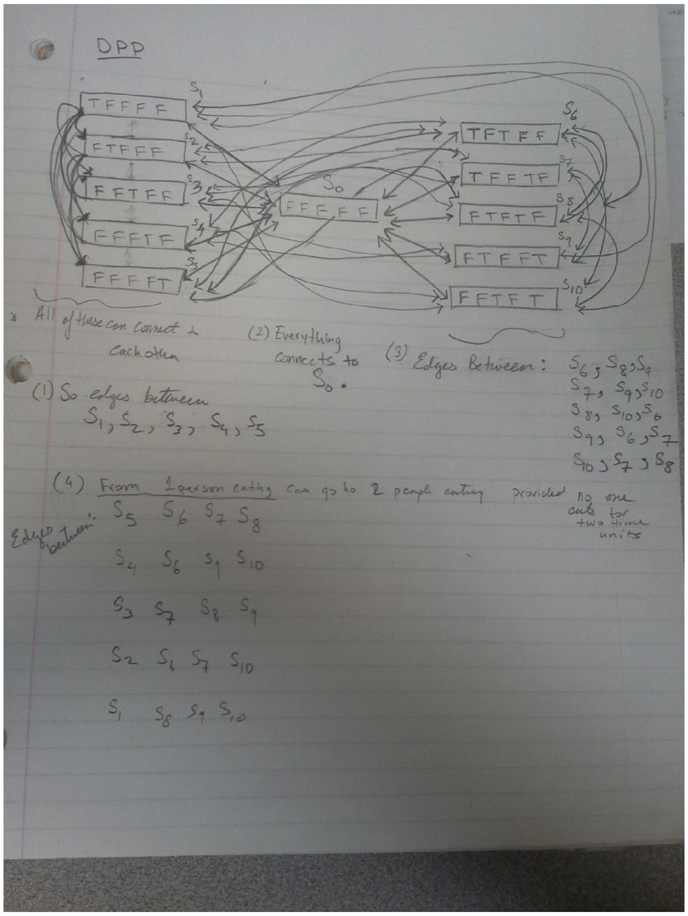


Figure 1: DPP Model

Bonus:

(1) $\mathbf{AG}\neg(\mathbf{e_1} \wedge \mathbf{e_4})$
False: look at path $s_0, s_10, ...$ the path falsifies it.

(2) $\mathbf{A}(\neg(e_1 \vee e_3 \vee e_4 \vee e_5)\mathbf{U}e_2)$
False: Look at path $s_0, s_1, ..,$ the path falsifies the wwf.

(3) True wff statisfied by the model

(4) False look at path $s_0, s_8, s_0, s_4, ...$ the path falsifies the wff.

4 Problem 4

The situations where deadlock occurs is a. when everyone chooses their right fork, b. when everyone chooses their left fork. The probability of everyone choosing same side fork is $\frac{1}{2^5}$. So the probability of this happening for both these situations is $\frac{1}{32} + \frac{1}{32} = \frac{1}{16}$. So in iteration of the transition system the probability of a deadlock is $\frac{1}{16} \leq \frac{1}{10}$ So it is deadlock free with probability greater than $\frac{9}{10}$ Not only that but the probability of continuosly getting into this situation is $lim_{n \to \infty} (1/16)^n = 0$ so definiently deadlock free.