Problem Set 1

Fall 11

Due: Tuesday, September 13th, in class, before class begins.

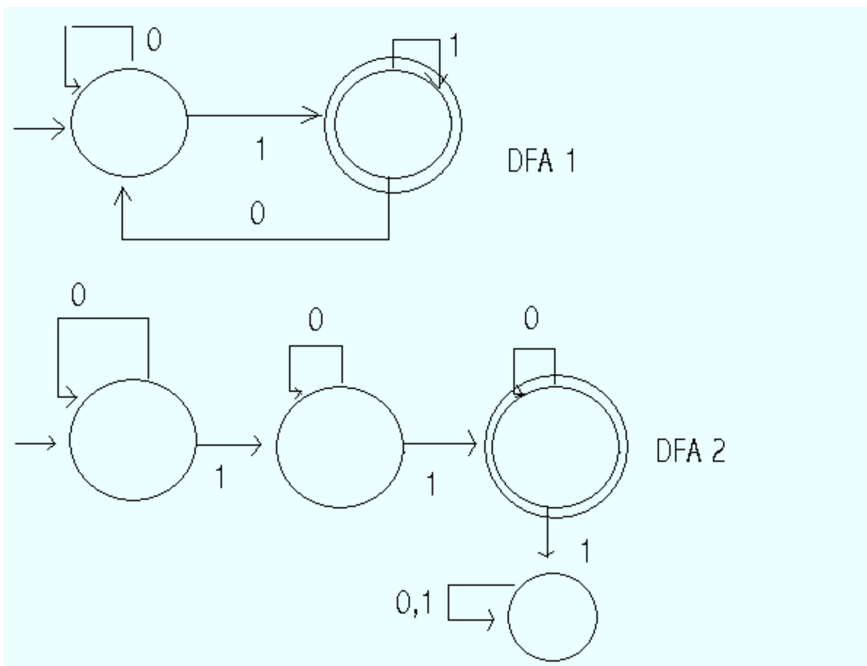
NOTE: Problem 6 is optional for EXTRA CREDIT. However, people who take the honor section MUST solve this problem.

Please <u>follow</u> the homework format guidelines posted on the class web page:

`http://www.cs.uiuc.edu/class/fa11/cs373/`

1. [**Category**: Understanding DFAs, **Points**: 10]

   Given the alphabet {0,1} and the following DFAs:



   (a) What is the language accepted by DFA 1?

   (b) What is the language accepted by DFA 2?

   (c) What is the language of their intersection?

Solution:

The language accepted by DFA 1 is $\{w1^n \mid w \in \{0,1\}^*, n > 0\}$, i.e. the set of all strings ending with a 1, and the language accepted by DFA 2 is $w \mid w$ contains exactly two$1s$. The language accepted by their intersection is $\{w11 \mid w \in \{0,1\}^*\}$, i.e. the set of all strings ending in 11.

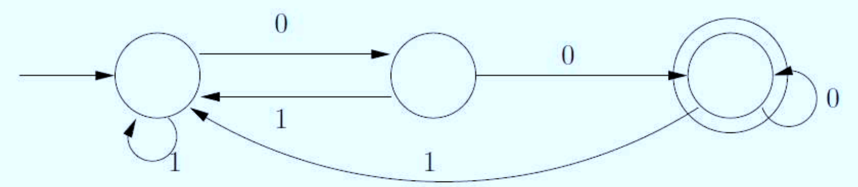
2. [**Category**: Language to DFA, **Points**: 10]

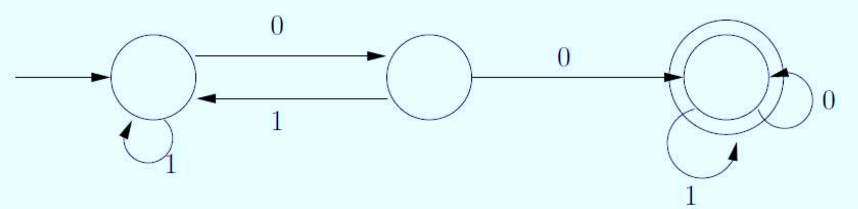
Given the following languages over $\Sigma = \{0, 1\}$ give a DFA recognizing the language in the form of a state diagram.

(a) $L = \{w00 \mid w \in \{0,1\}^*\}$, i.e. the set of all strings ending in 00.

(b) $L = \{w_1 00 w_2 \mid w_1, w_2 \in \{0,1\}^*\}$

Solution:





3. [**Category**: Construction, **Points**: 20]

There are two rooms A and B with lights, but with a single control switch for both rooms. Hence lights in rooms are both on or both off at any point.

The goal is to build an automatic control system that manages this switch. There is a sensor that detects motion in the two rooms and sends data to the controller; the controller reads these two signals and then instructs whether the switch should be turned on or off. We would like the controller to turn the lights on when motion is detected in either room, and turns them off if both rooms are empty for two consecutive signals from the sensor. Assume that the system starts from the state when lights are off.

The sequence of events and actions is represented by the alphabet $\Sigma = \{\textit{yes}, \textit{no}, \textit{on}, \textit{off}\}$.

Sequences accepted are of the form: $r_1\ s_1\ t_1\ r_2\ s_2\ t_2\ \ldots\ r_n\ s_n\ t_n$

where each $r_i \in \{\textit{yes}, \textit{no}\}$ and stands for the signal coming from Room A, each $s_i \in \{\textit{yes}, \textit{no}\}$ stands for the signal coming from room B, and each $t_i \in \{\textit{on}, \textit{off}\}$ stands for the instruction the controller gives to the switch.

Design a DFA that accepts precisely the sequences that conform to the behavior of the controller.

Example of good sequences:

(a) *no*, *no*, *off*

(b) *yes*, *yes*, *on*, *yes*, *no*, *on*, *no*, *no*, *on*, *no*, *no*, *off*

(c) *no*, *yes*, *on*, *no*, *no*, *on*, *no*, *yes*, *on*

(d) *yes*, *no*, *on*, *no*, *no*, *on*, *no*, *yes*, *on*

Example of bad sequences:

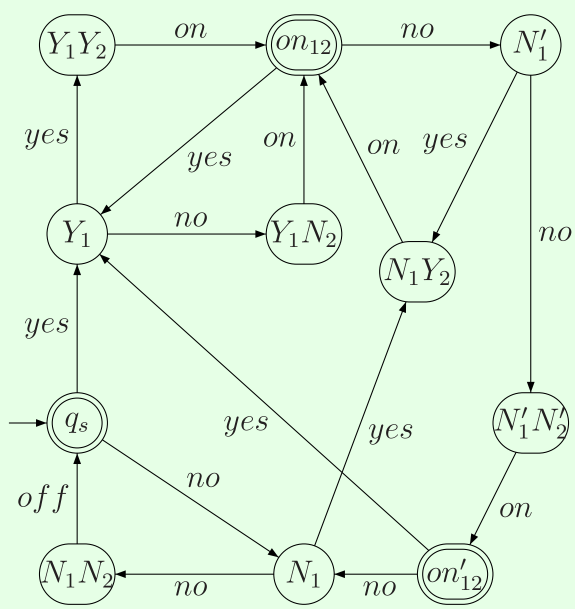
(a) *yes*, *no*, *on*, *no*

(b) *no*, *no*, *yes*

(c) *yes*, *yes*, *off*

Hint: You can assume the existence of a "trap" state, T, where automata goes if it finds any unexpected event (those transition not specified in your diagram) in the sequence.

Solution:

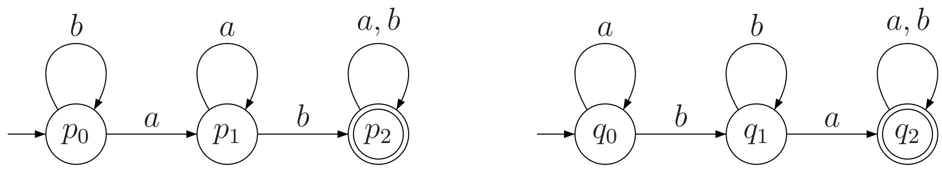


State name indicates either sensor value from each room (e.g., Y_1N_2 means that sensor detects something in room 1 but not in room 2) or condition of light in both rooms (e.g., *on* and *on′* means that light is on in both rooms and q_s is the initial state where light is off in both rooms).

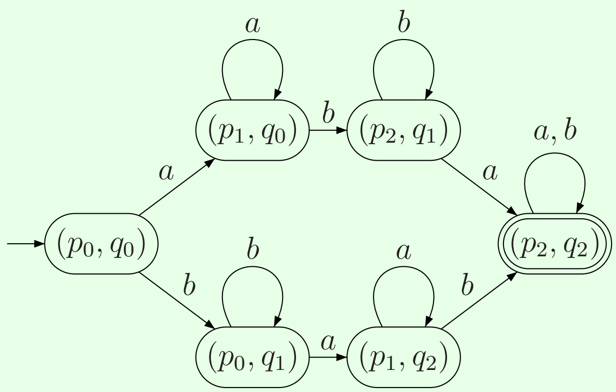
We need two final states to memorize the reason that the controller turned on the light. For state *on*, it memorizes that the controller turns on the light because something is detected in either or both room(s). For state *on′*, it memorizes that the controller turns on the light but there is nothing in both rooms and the controller is ready to turn the light off if nothing is detected in both rooms again.

4. [**Category**: Construction, **Points**: 20]

Construct the product of the following two DFAs that accepts the intersection of the languages of the two DFAs.



Solution:



5. [**Category**: Notation, **Points**: 20]

Write down the first DFA (that consists of states p_0, p_1, p_2) in the previous problem using formal notation (make sure to clearly describe all five important pieces of a DFA).

Solution:

It is the 5-tupe $(Q, \Sigma, \delta, p_0, F)$ where:

$$\Sigma = \{a, b\}$$
$$Q = \{p_0, p_1, p_2\}$$
$$F = \{p_2\}$$

δ	a	b
p_0	p_1	p_0
p_1	p_1	p_2
p_2	p_2	p_2

6. [**Category**: Extra credit, **Points**: 20]

Fix an arbitrary alphabet Σ.

A *pattern* is a string $p \in \Sigma^*$. For any pattern p, let the language $L_p = \{w \mid p \textit{ occurs in } w \textit{ as a subsequence}\}$.

When we say p occurs in w as a subsequence, we do not mean that p occurs as a consecutive subsequence. For example, ab occurs as a subsequence in the word $baccb$ (but not in ba). All we require the letters of p occur somewhere in w in the right order.

For any pattern p, show that L_p is regular. You must show this by constructing a DFA for L_p, in tuple mathematical notation, for *any* pattern p. No diagrams please. Your construction of the DFA should work for all patterns. You should also argue how/why this DFA works.

Solution:

Let $p \in \Sigma^*$ be a pattern.

Let $\mathit{Pre}(p)$ denote the set of all prefixes of p.

We construct the DFA accepting all words that have p as a subsequence as $A_p = (Q, \Sigma, \delta, q_0, F)$ where

- $Q = \mathit{Pre}(p)$
- $q_0 = \epsilon$
- $\delta(q, a) = qa$ if qa is a prefix of p, and $\delta(q, a) = q$, otherwise, for every $a \in \Sigma$.
- $F = \{p\}$

Intuitively, the DFA A_p has a state or every prefix of p (hence $|p| + 1$ states), and the automaton uses this state to remember the prefix of p that has been matched so far by the input. The initial state is ϵ (since no part of p has been matched) and the final state is p. The automaton moves to the next longer prefix whenever the correct letter is seen, and otherwise remains in the same state.

The language of A_p is hence the set of all words that have p as a subsequence.