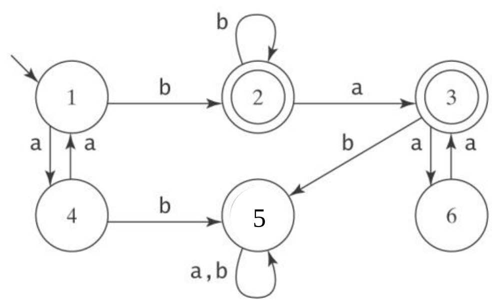
COMP2270/6270 – Theory of Computation
Third week

School of Electrical Engineering & Computing
The University of Newcastle

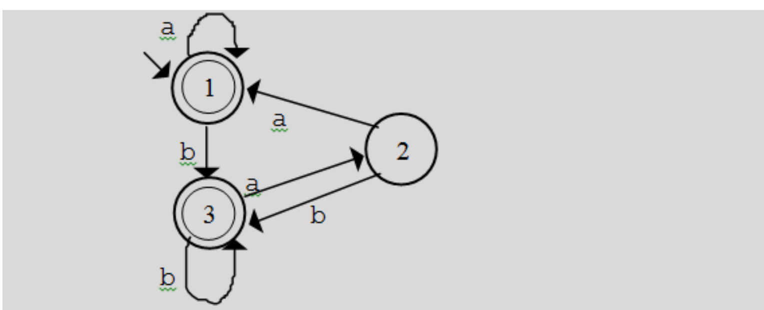
Exercise 1) (Chapter 5, Exercise 1 of Ref. [1])
Give a clear English description of the language accepted by the following FSM:



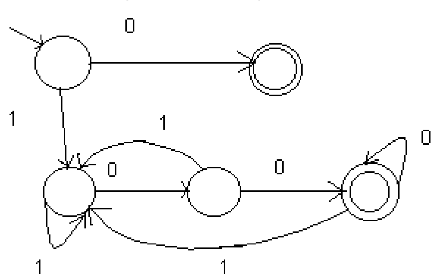
All strings of `a`'s and `b`'s consisting of an even number of `a`'s, followed by at least one `b`, followed by zero or an odd number of `a`'s.

Exercise 2) (Chapter 5, Selected cases of Exercise 2 of Ref. [1])
Build a deterministic FSM for each of the following languages

a) L = {*w* ∈ {a, b}* : *w* does not end in ba}.

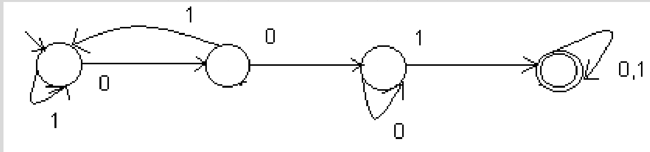


b) {w ∈ {0, 1}* : w corresponds to the binary encoding, without leading 0's, of natural numbers that are evenly divisible by 4}.

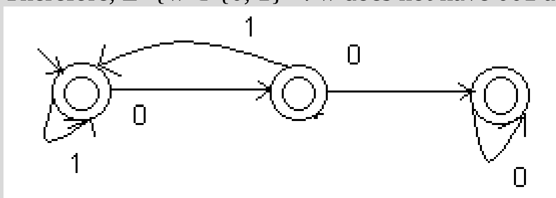


c) L= {w ∈ {0, 1}* : w does not have 001 as a substring}

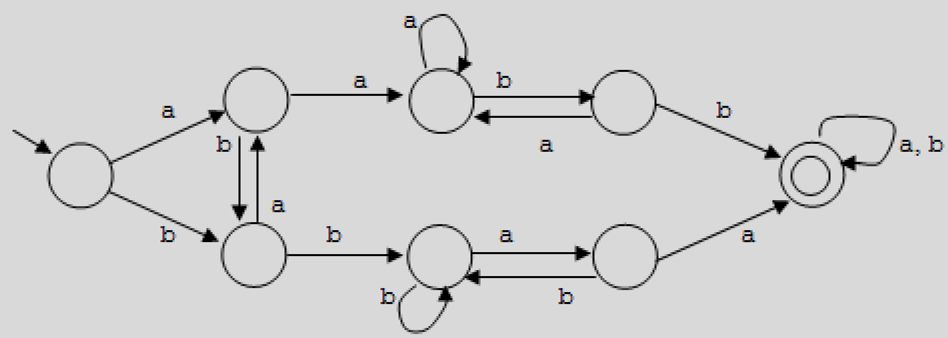
At first lets construct, L ={ w ∈ {0, 1}* : w has 001 as a substring}



Therefore, L={w ∈ {0, 1}* : w does not have 001 as a substring} is



d) {w ∈ {a, b}* : w has both aa and bb as a substrings}.



e) attempt to solve other exercises from the list of Chapter 5, Exercise 2 of Ref. [1]), (if you are studying with a colleague, attempt different exercises).

Exercise 3) (Chapter 5, Exercise 3 of Ref. [1])
Consider the children's game Rock, Paper, Scissors. We'll say that the first player to win two rounds wins the game. Call the two players A and B.

a) Define an alphabet Σ and describe a technique for encoding Rock, Paper, Scissors games as strings over Σ (Hint: each symbol in Σ should correspond to an ordered pair that describes the simultaneous actions of A and B.)

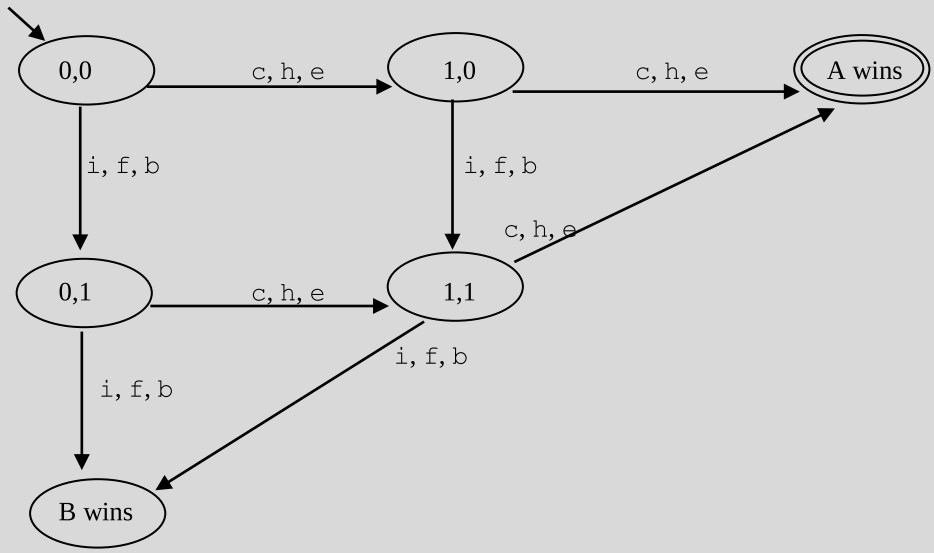
Let Σ have 9 characters. We'll use the symbols `a` - `i` to correspond to the following events. Let the first element of each pair be A's move. The second element of each pair will be B's move.

a	b	c	d	e	f	g	h	i
R, R	R, P	R, S	P, P	P, R	P, S	S, S	S, P	S, R

A Rock, Paper, Scissors game is a string of the symbols `a` - `i`. We'll allow strings of arbitrary length, but once one player as won two turns, no further events affect the outcome of the match.

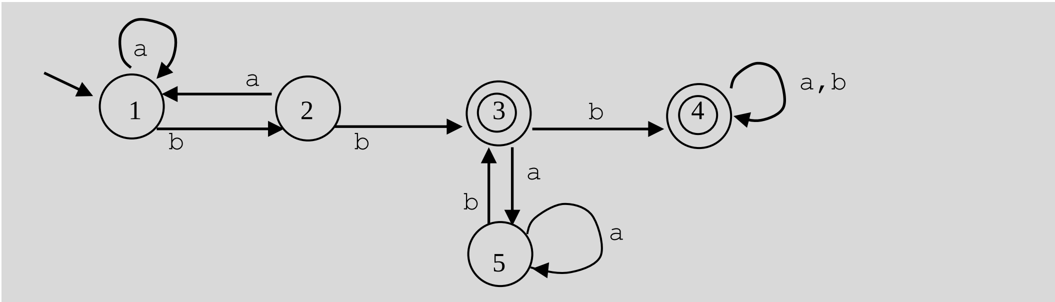
b) Let L_{RPS} be the language of Rock, Paper, Scissors games, encoded as strings as described in part (a), that correspond to wins for player *A*. Show a DFSM that accepts L_{RPS}.

In the following diagram, a state with the name (*n*, *m*) corresponds to the case where player *A* has won *n* games and player *B* has won *m* games.



In addition, from every state, there is a transition back to itself labeled `a`, `d`, `g` (since the match status is unchanged if both players make the same move). And, from the two winning states, there is a transition back to that same state with all other labels (since, once someone has won, future events don't matter).

Exercise 4) You have been given the diagram below and you have been told that it is the diagram of a FSM that recognizes a language L. Give a simple English description for a language that is accepted by this DFSM and an interpretation (also an English description) for each of the states in the DFSM.

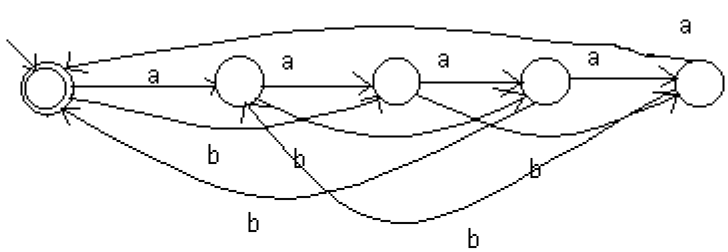


$L = \{w \in \{\mathtt{a}, \mathtt{b}\}^* :$ w contains at least two b's that are not immediately followed by a's$\}$.

This is problem is a bit tricker than it looks. Define a b to be "clear" iff it is not followed by an a. A b is "definitely clear" iff it is already known to be clear, regardless of what comes next (i.e., it was followed by another b). A b is "possibly clear" iff it is the mostly recently read symbol. So it will turn out to be clear iff either it is the last input symbol or it is followed by another b. Then states in the following solution correspond to the following situations:

1. No definitely or possibly clear b's.
2. No definitely clear b's. One possibly clear b.
3. One definitely clear b. One possibly clear b.
4. At least two definitely clear b's.
5. One definitely clear b. No possibly clear b.

Exercise 5) Give a simple English description for a language that is accepted by this DFSM



$\{w \in \{\mathtt{a}, \mathtt{b}\}^*: (\#_{\mathtt{a}}(w) + 2 \cdot \#_{\mathtt{b}}(w)) \equiv_5 0\}$. ($\#_{\mathtt{a}}w$ is the number of a's in w).

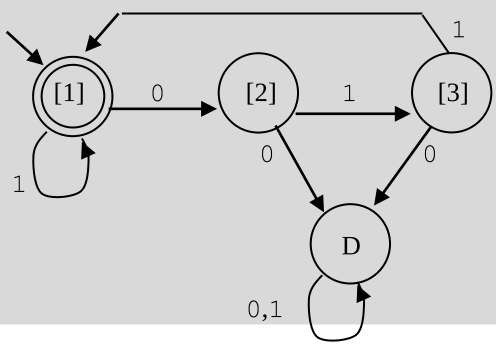
Exercise 6) For following languages L and L'

(i) Describe the equivalence classes of $\approx_L$.

(ii) If the number of equivalence classes of $\approx_L$ is finite, construct the minimal DFSM that accepts L.

$L = \{w \in \{0, 1\}^* :$ every 0 in w is immediately followed by the string 11$\}$.
$L' = \{ww^R : w \in \{a, b\}^*\}$.

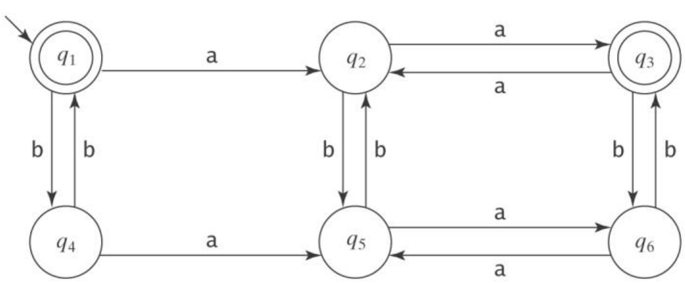
[1] {in L}
[2] {otherwise in L except ends in 0}
[3] {otherwise in L except ends in 01}
[D] {corresponds to the Dead state: string contains at least one instance of 00 or 010}



[1] {ε} in L'
[2] {a}
[3] {b}
[4] {aa} in L'
[5] {ab}

And so forth. Every string is in a different equivalence class because each could become in L' if followed by the reverse of itself but not if followed by most other strings. This language is not regular!!!

Exercise 7) Let M be the following DFSM. Use *minDFSM* to minimize M.



Initially, *classes* = {[1, 3], [2, 4, 5, 6]}.

At step 1:
((1, a), [2, 4, 5, 6]) ((3, a), [2, 4, 5, 6]) No splitting required here.
((1, b), [2, 4, 5, 6]) ((3, b), [2, 4, 5, 6])

((2, a), [1, 3]) ((4, a), [2, 4, 5, 6]) ((5, a), [2, 4, 5, 6]) ((6, a), [2, 4, 5, 6])
((2, b), [2, 4, 5, 6]) ((4, b), [1, 3]) ((5, b), [2, 4, 5, 6]) ((6, b), [1, 3])

These split into three groups: [2], [4, 6], and [5]. So classes is now {[1, 3], [2], [4, 6], [5]}.

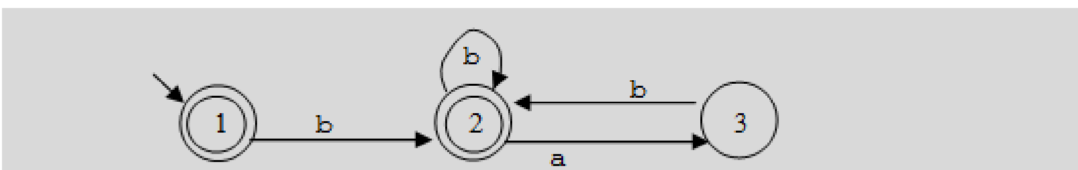
At step 2, we must consider [4, 6]:

((4, a), [5]) ((6, a), [5])
((4, b), [1]) ((6, b), [1])

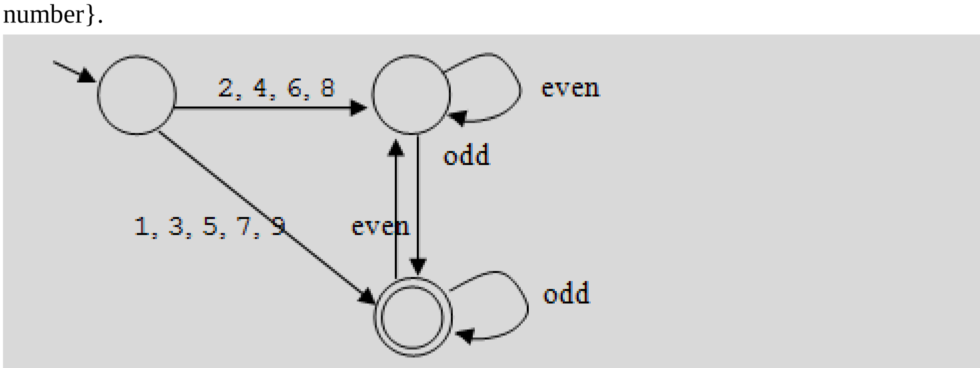
No further splitting is required. The minimal machine has the states: {[1, 3], [2], [4, 6], [5]}, with transitions as shown above.

EXTRAS FROM THE BOOK

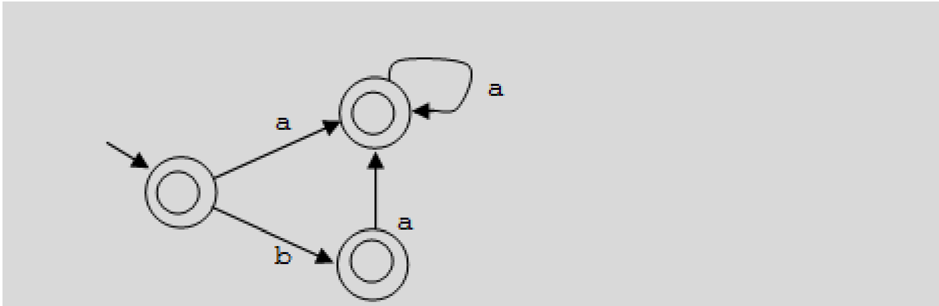
1) (Chapter 5, Selected cases of Exercise 2 of Ref. [1])
Build a deterministic FSM for each of the following languages:
   a) {$w \in$ {a, b}* : every a in w is immediately preceded and followed by b}.



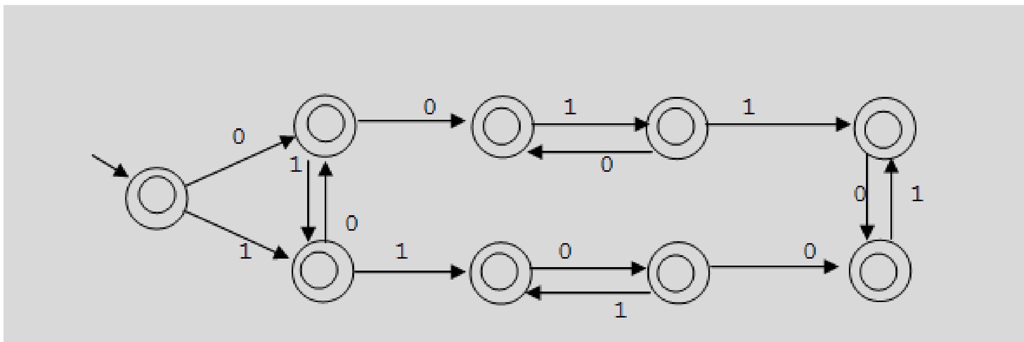
   b) {$w \in$ {0-9}* : w corresponds to the decimal encoding, without leading 0's, of an odd natural number}.



c) {$w \in$ {a, b}* : w has neither ab nor bb as a substring}.



d) The set of binary strings with at most one pair of consecutive 0's and at most one pair of consecutive 1's.



REFERENCES

[1] Elaine Rich, Automata Computability and Complexity: Theory and Applications, Pearson, Prentice Hall, 2008.