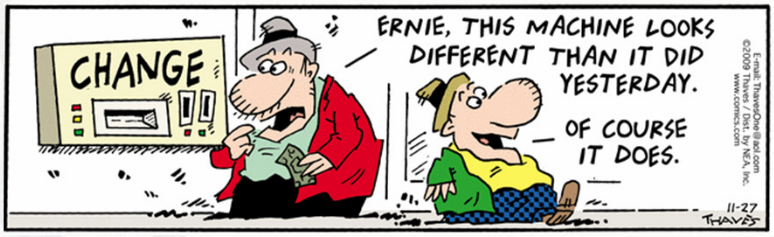
MA/CSSE 474

Theory of Computation



More FSMs

NFSMs

Exam 1: Session 12 (Dec 16)

- Resources allowed:
  - A double-sided 8.5 x 11 sheet of paper with whatever you want on it.
  - No books or electronic devices.
- I'll tell you by Tuesday how far the exam will cover.
- HW 5 will be due the following Tuesday..

Recap: Definition of a DFSM

The D is for Deterministic

$M = (K, \Sigma, \delta, s, A)$, where:

- K is a finite ***set of states***
- Σ is a (finite) ***alphabet***
- $s \in K$ is the ***initial state*** (a.k.a. start state)
- $A \subseteq K$ is the ***set of accepting states***
- $\delta: (K \times \Sigma) \to K$ is the ***transition function***

Sometimes we will put an M subscript on K, Σ, δ, s, or A (for example, s_M), to indicate that this component is part of machine M.

Q1

Recap: Configurations of a DFSM

A ***configuration*** of a DFSM M is an element of:

$$K \times \Sigma^*$$

It captures the two things that affect M's future behavior:

- its current state
- the remaining input to be read.

The ***initial configuration*** of a DFSM M, on input w, is:

$$(s_M, w)$$

Q2

Recap: The "Yields" Relations

The *yields-in-one-step* relation: $|-_M$:

$(q, w)\ |-_M\ (q', w')$ iff

- $w = a\, w'$ for some symbol $a \in \Sigma$, and
- $\delta\,(q, a) = q'$

In a context where there is only one machine under consideration, we may sometimes omit the M and simply write |-

The *yields-in-zero-or-more-steps* relation: $|-_M{}^*$

$|-_M$ * is the reflexive, transitive closure of $|-_M$.

The *yields-in-exactly-n-steps* relation: $|-_M{}^n$

"Yields in exactly n steps", where $n \geq 0$

Q3

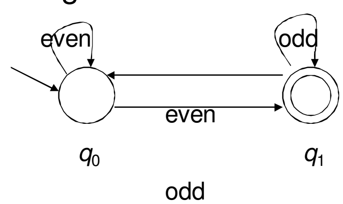
Recap: Computations Using FSMs

A ***computation*** by M is a finite sequence of configurations $C_0, C_1, \ldots, C_n$ for some $n \geq 0$ such that:

- C_0 is an initial configuration,
- C_n is of the form (q, ε), for some state $q \in K_M$,
- $\forall i \in \{0, 1, \ldots, n\text{-}1\}\ (C_i\ |-_M\ C_{i+1})$

Recap: An Example Computation

An FSM M that accepts decimal representations of odd integers:



On input 235, the configurations are:

$(q_0, 235)$ $|-_M$ $(q_0, 35)$
$|-_M$
$|-_M$

Thus $(q_0, 235)$ $|-_M{}^*$ (q_1, ε)

Q4

Recap: Accepting and Rejecting

A DFSM *M* ***accepts*** a string *w* iff:

(s_M, w) $|-_M{}^*$ (q, ε), for some $q \in A_M$

A DFSM *M* ***rejects*** a string *w* iff:

(s_M, w) $|-_M{}^*$ (q, ε), for some $q \notin A_M$

The ***language accepted by*** *M*, denoted *L*(*M*), is the set of *all* strings accepted by *M*.

What is L(M) for the machine on the in-class quiz?

Theorem: Every DFSM *M*, in configuration (q, w), halts in exactly $|w|$ steps.

Q5,6,7

Regular Languages

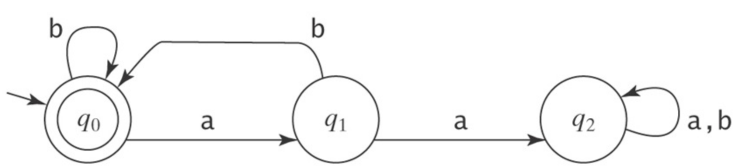
Definition:
A language L is ***regular***
iff
L = L(M) for some DFSM M.

Q8

Example

$L = \{w \in \{\mathtt{a}, \mathtt{b}\}^* :$

every a is immediately followed by a b}.



δ can also be represented as a **transition table**:

state ↓ input →	a	b
q_0	q_1	q_0
q_1	q_2	q_0
q_2	q_2	q_2

q_2 is a **dead state**.

Parity Checking

$L = \{w \in \{\texttt{0}, \texttt{1}\}^* : w \text{ has odd parity}\}$.
I.e. an odd number of 1's.

Q9

Even a Regions

$L = \{w \in \{\texttt{a}, \texttt{b}\}^* : \text{every } \texttt{a} \text{ region in } w \text{ has even length}\}$.

Note that by "a region", we mean a maximal sequence of consecutive as.

This would be a good example for practice later

Checking Consecutive Characters

$L = \{w \in \{\texttt{a}, \texttt{b}\}^*$:

no two consecutive characters are the same$\}$.

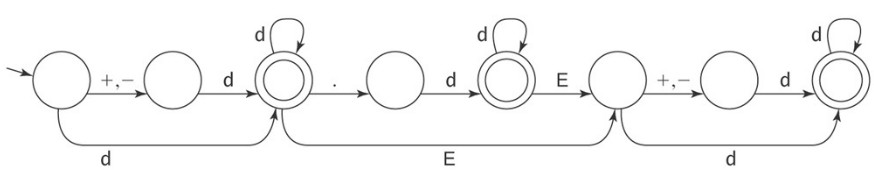
Q10

The Language of Floating Point Numbers is Regular

Example strings:

`+3.0, 3.0, 0.3E1, 0.3E+1, -0.3E+1, -3E8`

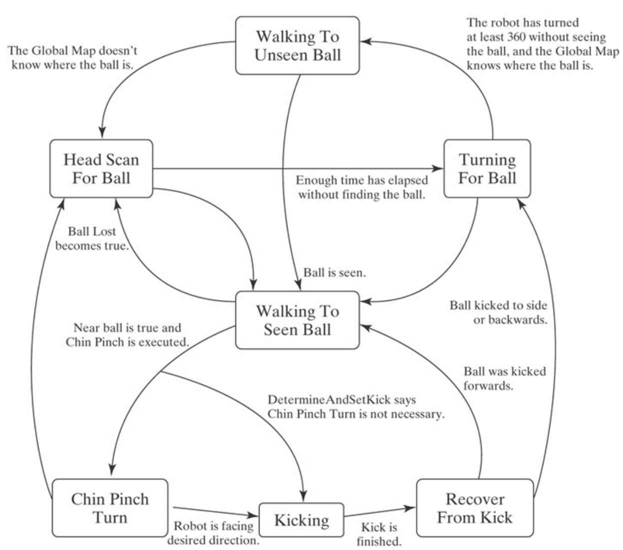
The language is accepted by the DFSM:



We will now take a very quick look at a few DFSM examples

Controlling a Soccer-Playing Robot

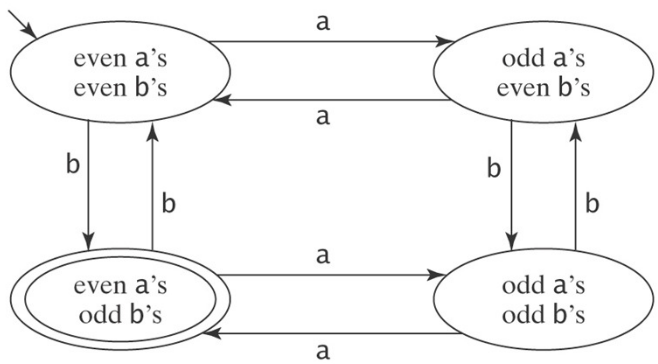
A Simple Controller



Programming FSMs

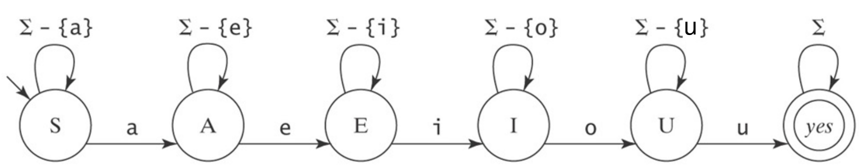
Cluster strings that share a "future".

Let $L = \{w \in \{\texttt{a}, \texttt{b}\}^* : w$ contains an even number of `a`'s and an odd number of `b`'s$\}$



Vowels in Alphabetical Order

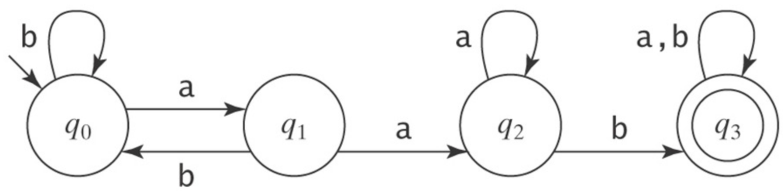
$L = \{w \in \{\texttt{a} - \texttt{z}\}^* :$ all five vowels, `a`, `e`, `i`, `o`, and `u`, occur in w in alphabetical order$\}$.



Programming FSMs

$L = \{w \in \{\texttt{a}, \texttt{b}\}^* : w$ does not contain the substring $\texttt{aab}\}$.

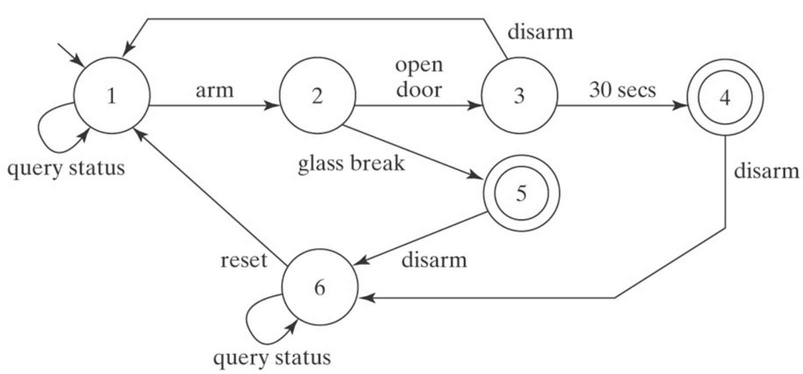
Start with a machine for $\neg L$:



How must it be changed?

A Building Security System

L = {event sequences such that the alarm should sound}



The Missing Letter Language

Let Σ = {a, b, c, d}.

Let $L_{Missing}$ =
{w : there is a symbol $a_i \in \Sigma$ not appearing in w}.

Try to make a DFSM for $L_{Missing}$:

Q7

The Missing Letter Language

Let Σ = {a, b, c, d}.

Let $L_{Missing}$ =
{w : there is a symbol $a_i \in \Sigma$ not appearing in w}.

Try to make a DFSM for $L_{Missing}$:

Do this on your own if you want extra practice.
For now, think about why it is hard.

NONDETERMINISTIC FSM

Definition of an NDFSM

$M = (K, \Sigma, \Delta, s, A)$, where:

K is a finite set of states

Σ is an alphabet

$s \in K$ is the initial state

$A \subseteq K$ is the set of accepting states, and

Δ is the **transition relation**. It is a finite subset of

$$(K \times (\Sigma \cup \{\varepsilon\})) \times K$$

Another way to present it: $\Delta : (K \times (\Sigma \cup \{\varepsilon\})) \rightarrow 2^K$

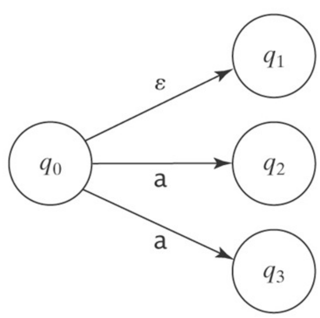
Q11

Accepting by an NDFSM

M **accepts** a string *w* iff there exists *some path* along which *w* drives *M* to some element of *A*.

The language accepted by *M*, denoted *L*(*M*), is the set of all strings accepted by *M*.

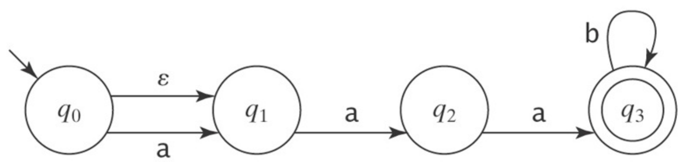
Sources of Nondeterminism



Q12-13

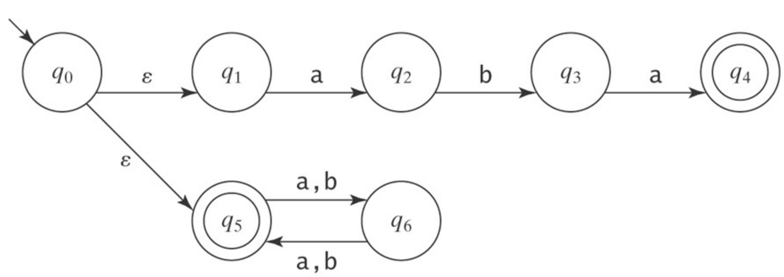
Optional Substrings

$L = \{w \in \{a, b\}^* : w$ is made up of an optional a followed by aa followed by zero or more b's$\}$.



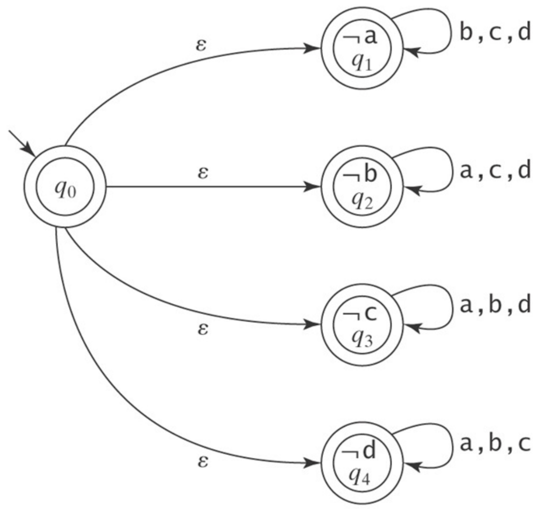
Multiple Sublanguages

$L = \{w \in \{a, b\}^* : w = aba$ or $|w|$ is even$\}$.



The Missing Letter Language

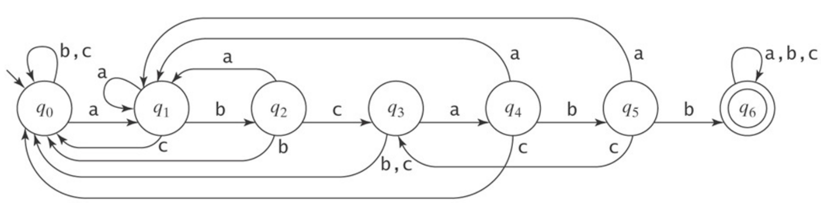
Let Σ = {a, b, c, d}. Let $L_{Missing}$ = {w : there is a symbol $a_i \in \Sigma$ that does not appear in w}



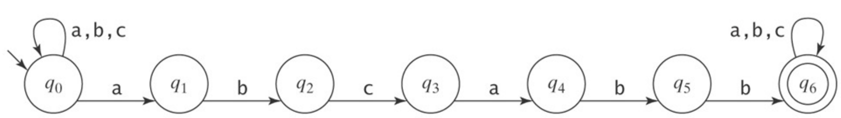
Pattern Matching

L = {$w \in$ {a, b, c}* : $\exists x, y \in$ {a, b, c}* ($w = x$ abcabb y)}.

A DFSM:



An NDFSM:



Pattern Matching: Multiple Keywords

$L = \{w \in \{\texttt{a}, \texttt{b}\}^* : \exists x, y \in \{\texttt{a}, \texttt{b}\}^*$

$((w = x\,\texttt{abbaa}\,y) \vee (w = x\,\texttt{baba}\,y))\}$.