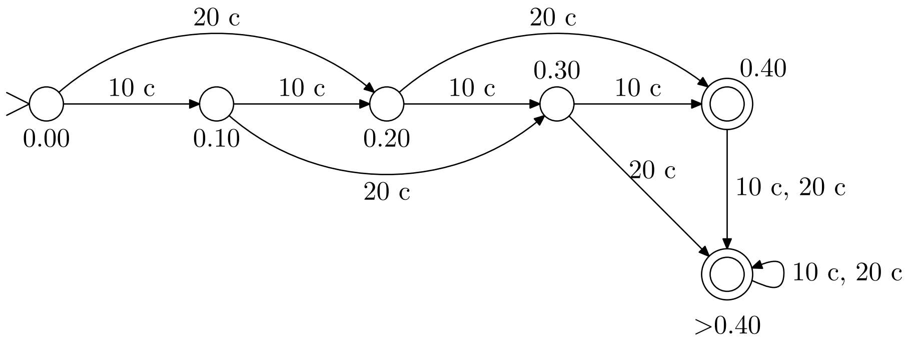
T-79.1001/2 **Syksy 2006**
Introduction to Theoretical Computer Science (T/Y)
Session 2
Answers to demonstration exercises

4. **Problem**: Formulate the model of a simple coffee machine presented in class (lecture notes p. 15) precisely according to the mathematical definition of a finite automaton (Definition 2.1). What is the formal language recognized by this automaton?

   Solution: The problem was to give formal representation to the finite automaton given below:



Formally, a deterministic state machine (finite automaton) is a tuple $M = (Q, \Sigma, \delta, q_0, F)$, where Q is finite set of states, Σ finite alphabet, δ function $Q \times \Sigma \to Q$, $q_0 \in K$ initial state and $F \subseteq Q$ set of accepting states.

In the Coffee Machine the parts are defined as follows:

$$\begin{aligned}
Q &= \{0.00, 0.10, 0.20, 0.30, 0.40, {>}0.40\}\\
\Sigma &= \{10\text{c}, 20\text{c}\}\\
q_0 &= \{0.00\}\\
F &= \{0.40, {>}0.40\}
\end{aligned}$$

The easiest way to show the transition function δ is a table.

q	$\delta(q, 10\text{c})$	$\delta(q, 20\text{c})$
0.00	0.10	0.20
0.10	0.20	0.30
0.20	0.30	0.50
0.30	0.40	>0.40
0.40	>0.40	>0.40

A configuration $c \in Q \times \Sigma^*$ of a finite automaton contains the current state and input. The automaton reads input one symbol at a time and then moves to a new state according to the symbol and current state. If the automaton is in an accepting state when the word ends, the word is accepted. Otherwise the word is rejected. The language $L(M)$ accepted by a finite automaton M is the set of all accepted words. For the automaton in question it is:

$$L(M) = \{x_1 x_2 \ldots x_n \mid x_i \in \Sigma \text{ for all } 1 \leq i \leq n \text{ and } \sum_{i=1}^{n} x_i \geq 40\ c\}$$

So the machine accepts all the strings in which the sum of given coins is 40 c or more. Let us see through a few inputs and how the machines operates with them.

- $w = 0.10\text{c } 0.10\text{c } 0.20\text{c}$:

$$\begin{aligned}(0.00, 0.10\text{c } 0.10\text{c } 0.20\text{c}) &\vdash_M (0.10, 0.10\text{c } 0.20\text{c})\\ &\vdash_M (0.20, 0.20\text{c}) \vdash_M (0.40, \varepsilon)\end{aligned}$$

  Because $0.40 \in F$ the word is accepted. Here the $\vdash_M$ means that machine M proceeds one step.

- $w = 0.20\text{c } 0.10\text{c}$:

$$(0.00, 0.20\text{c } 0.10\text{c}) \vdash_M (0.20, 0.10\text{c }) \vdash_M (0.30, \varepsilon)$$

  Because $0.30 \notin F$, the word is rejected.

- $w = 0.20\text{c } 0.20\text{c } 0.20\text{c}$:

$$(0.00, 0.20\text{c } 0.20\text{c } 0.20\text{c}) \vdash_M^* (>0.40, \varepsilon)$$

  Word is accepted. Marking $\vdash_M^*$ denotes that machine M proceeds zero or more steps.

5. **Problem**: Design finite automata that recognise the following languages:

   (a) $\{a^m b^n \mid m = n \mod 3\}$;

   (b) $\{w \in \{a, b\}^* \mid w$ contains equally many a's and b's, modulo 3$\}$.

   (The notation "$m = n \mod 3$" means that the numbers m and n yield the same remainder when divided by three.)

   Solution:

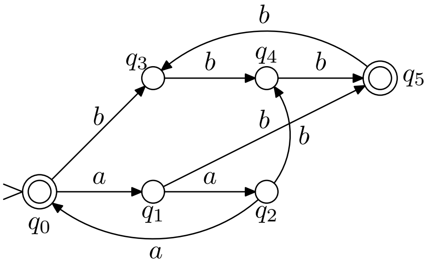
   a) The language $L = \{a^m n^n \mid m = n \mod 3\}$ can be recognized by the finite automaton:

$$\begin{aligned}M &= (Q, \Sigma, \delta, q_0, F)\\ Q &= \{q_0, q_1, q_2, q_3, q_4, q_5, q_6\}\\ \Sigma &= \{a, b\}\\ F &= \{q_0, q_5\}\end{aligned}$$

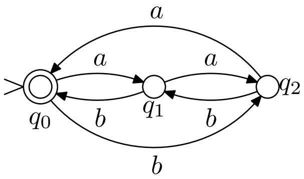
   The state transition function δ is:

q	$\delta(q, a)$	$\delta(q, b)$
q_0	q_1	q_3
q_1	q_2	q_5
q_2	q_0	q_4
q_3	q_6	q_4
q_4	q_6	q_5
q_5	q_6	q_3
q_6	q_6	q_6

   The state q_6 is used as a rejecting state. It means that the automaton moves into that state as soon as it is clear that the word cannot belong to the language (when ba substring is found). The machine then stays in that state until the end of the string. These states are often left out when an automaton is represented as a state diagram. This is also the case for the diagram below:

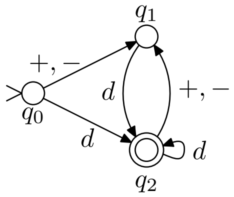


b) Language $L = \{w \in \{a,b\}^* \mid w$ contains as many a and b letters modulo $3\}$ is recognized by the following finite automaton:



6. **Problem**: Design a finite automaton that recognizes sequences of integers separated by plus and minus signs (e.g. $11+20-9$, $-5+8$). Implement your automaton as a computer program that also calculates the numerical value of the input expression.

   Solution: The plus and minus operations can be recognized with the following automaton:



Here d is a shorthand notation that means any number from set $\{0, \ldots, 9\}$.

Below is a C-program that reads user input and does the required addition. It has been modified from the C-program presented in the lecture that can identify a number and it's size.

```
#include <stdio.h>
#include <ctype.h>

int main (void)
{
  int q;   /* state */
  int c;   /* input symbol */
  int sgn, val, sum;
  sgn=1; val = 0, sum = 0;
  q = 0;

  while ((c = getchar()) !='\n') {
    switch (q) {
    case 0:
      if (c == '+') q = 1;
      else if (c == '-') {
        sgn = -1;
        q = 1;
      }
      else if (isdigit(c)) {
        val = c -'0';
        q = 2;
      }
      else q = 99;
      break;
    case 1:
      if (isdigit(c)) {
        val = c -’0’;
        q = 2;
      }
      else q = 99;
      break;

    case 2:
      if (isdigit(c)) {
        val = 10 * val + (c - ’0’);
        q = 2;
      }
      else if (c == ’+’)
        {
          sum = sum + val*sgn;
          val = 0;
          sgn = 1;
          q = 1;
        }
      else if (c == ’-’)
        {
          sum = sum + val*sgn;
          val = 0;
          sgn = -1;
          q = 1;
        }
      else q = 99;
      break;

    case 99:
      break;
    }
  }
  sum = sum + sgn*val;
  if (q == 2)
    printf("SUM IS %d.\n", sum);
  else
    printf("INVALID INPUT.\n");
  return;
}
```