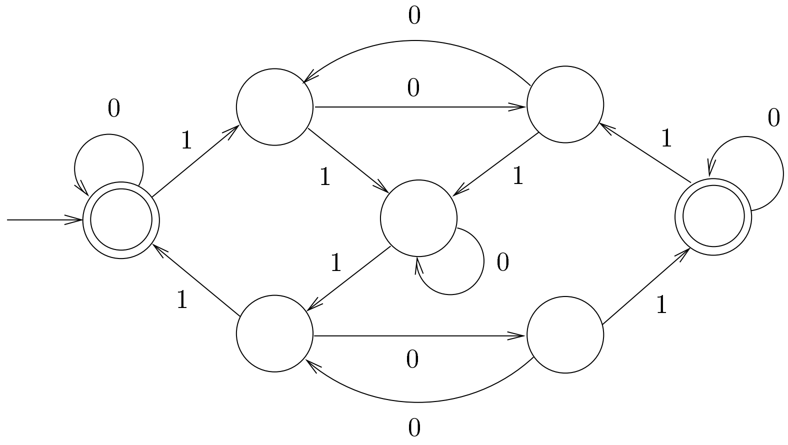
T-79.148 **Spring 2003**
Introduction to Theoretical Computer Science
Tutorial 4, 10–12 February
Problems

Homework problems:

1. Construct the minimal automaton corresponding to the following deterministic finite automaton:



2. Construct a nondeterministic finite automaton that tests whether a given binary input sequence contains either 0010 or 1101 (or both) as a subsequence. Make the automaton deterministic using the subset construction. (You do not need to list explicitly all the mutually equivalent accepting final states.)

3. Show that if a language $L \subseteq \{0,1\}^*$ is recognised by some finite automaton, then so are the following languages, consisting of all the prefixes and suffixes of the words in L:

$$\begin{aligned} \mathrm{Pref}_L &= \{x \in \{0,1\}^* \mid xy \in L \text{ for some } y \in \{0,1\}^*\}, \\ \mathrm{Suff}_L &= \{y \in \{0,1\}^* \mid xy \in L \text{ for some } x \in \{0,1\}^*\}. \end{aligned}$$

Demonstration problems:

4. Construct a nondeterministic finite automaton that tests whether in a given binary input sequence the third-to-last bit is a 1. Make the automaton deterministic using the subset construction.

5. Show that if languages A and B over the alphabet $\Sigma = \{a, b\}$ are recognised by some finite automata, then so are the languages $\bar{A} = \Sigma^* - A$, $A \cup B$, and $A \cap B$.

PLEASE TURN OVER

6. *(Application.)* Many methods for analysing data transfer protocols construct the global state space of the system, which can then be examined to find out the possibilities for undesirable events, e.g. deadlocks. One way of constructing the global state space is to model each participant of the protocol as a finite automaton, and join all of these together into one big state machine.

   Let us study specifically the case of two interacting nondeterministic automata $M_1 = (K_1, \Sigma_1, \Delta_1, s_1, \emptyset)$ and $M_2 = (K_2, \Sigma_2, \Delta_2, s_2, \emptyset)$.[1] The joint state machine $M = (K, \Sigma, \Delta, s, \emptyset)$ is then constructed in the following way:

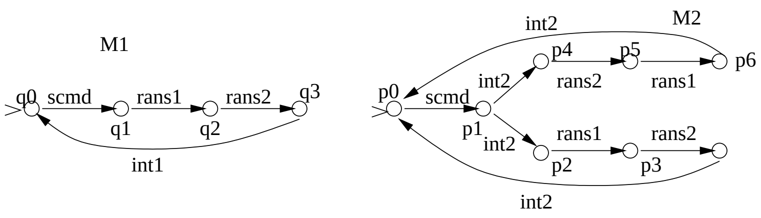
   - $K = K_1 \times K_2$
   - $\Sigma = \Sigma_1 \cup \Sigma_2$
   - $s = (s_1, s_2)$
   - The transition $(p_1, p_2) \xrightarrow{a} (q_1, q_2)$ is in the relation Δ if any of the following conditions hold:

     (a) $a \in \Sigma_1 \cap \Sigma_2$, $(p_1, a, q_1) \in \Delta_1$ and $(p_2, a, q_2) \in \Delta_2$.

     (b) $a \in \Sigma_1$, $a \notin \Sigma_2$, $(p_1, a, q_1) \in \Delta_1$ and $p_2 = q_2$.

     (c) $a \notin \Sigma_1$, $a \in \Sigma_2$, $(p_2, a, q_2) \in \Delta_2$ and $p_1 = q_1$.

   Let M_1 and M_2 be as below. Construct the joint state machine M and show that the system contains a reachable deadlock (i.e. a reachable state from which there are no further transitions).



[1] The automata notation used in this problem differs slightly from that used for the rest of the course. The state set of an automaton is denoted by K rather than Q, the initial state is denoted by s rather than q_0, and the set-valued transition function of a nondeterministic automaton, $\delta\colon K \times \Sigma \to \mathcal{P}(K)$ is represented as a relation $\Delta \subseteq K \times \Sigma \times K$, where $(p, a, q) \in \Delta$ if and only if $q \in \delta(p, a)$.