T-79.148 **Spring 2005**
Introduction to Theoretical Computer Science
Tutorial 3, 8–11 February
Problems

Remember to enroll for the course using the TOPI registration system by 4 February. Registration is compulsory.

Homework problems:

1. Design finite automata that recognise the following languages:
   (a) $\{w \in \{a, b\}^* \mid w \text{ contains } abb \text{ as a substring}\}$;
   (b) $\{w \in \{a, b\}^* \mid w \text{ does not contain } abb \text{ as a substring}\}$;
   (c) $\{w \in \{a, b\}^* \mid ab \text{ occurs exactly twice as a substring in } w\}$;
   (d) $\{w \in \{0, 1\}^* \mid w \text{ contains an even number (possibly zero) of 0's}\}$;
   (e) $\{w \in \{0, 1\}^* \mid \text{the number of 1's in } w \text{ is divisible by three (or possibly zero)}\}$;
   (f) $\{w \in \{0, 1\}^* \mid w \text{ begins and ends with different symbols}\}$.
   (g) $\{w \in \{0, 1\}^* \mid w \text{ contains an even number of 0's or ends with 1}\}$.

2. Design a finite automaton that accepts precisely those binary strings that contain an even number of 0's and the number of 1's is divisible by three (e.g. 00111, 0000 and 10101, but not 0011 or 11). [NB. In this and similar problems in the future, it is for simplicity always assumed that zero is an even number, divisible by three, etc., unless otherwise indicated.]

3. Design a finite automaton to control the traffic lights at an extremely low-traffic intersection. Two streets meet at the intersection, and the traffic light on each street can display either red, yellow, or green. The traffic situation is indicated by three mutually exclusive input signals: 'car arriving on street 1', 'car arriving on street 2', and 'no arriving traffic'. The automaton must ensure that each car arriving at the intersection gets to continue, and that if the traffic light in one direction is green or yellow, then the light in the intersecting direction is red. The automaton does not need to have any distinct start or final states.

Demonstration problems:

4. Formulate the model of a simple coffee machine presented in class (lecture notes p. 17) precisely according to the mathematical definition of a finite automaton (Definition 2.1). What is the formal language recognised by this automaton?

5. Design finite automata that recognise the following languages:
   (a) $\{a^m b^n \mid m = n \mod 3\}$;
   (b) $\{w \in \{a, b\}^* \mid w \text{ contains equally many } a\text{'s and } b\text{'s, modulo 3}\}$.

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

6. Design a finite automaton that recognises 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.