1. *Dining Philosophers:* Five philosophers sit around a plate of noodles. Between each philosopher there is a single chopstick (so there are five chopsticks in total). A philosopher can only eat if he holds the chopsticks at his left and at his right. When a philosopher is done eating he puts down the chopsticks.

   (a) Give a μCRL specification of the dining philosophers in which no philosopher is ever permanently excluded from the meal. (You may assume the data type *Bool* and its standard operations as given.) Represent picking up and putting down a pair of chopsticks i and j by actions $u(i,j)$ and $d(i,j)$, respectively.

   (b) Draw the process graph belonging to your μCRL specification.

2. Let $even : Nat \to Bool$ be defined such that $even(n) = \mathsf{T}$ if and only if n is an even number:

$$\begin{array}{lcl} even(0) & = & \mathsf{T} \\ even(S(n)) & = & \neg even(n) \end{array}$$

   (a) Consider the following LPE:

$$\begin{array}{lcl} X(n : Nat) & = & \tau{\cdot}X(n) \lhd even(n) \rhd \delta \\ & + & a(n){\cdot}X(S(S(n))) \lhd \mathsf{T} \rhd \delta \end{array}$$

   Prove that the confluence formula for the τ-summand is true, meaning that the τ-transitions generated by this summand are confluent.

   (b) Consider the following LPE:

$$\begin{array}{lcl} X(n : Nat) & = & \tau{\cdot}X(S(n)) \lhd even(n) \rhd \delta \\ & + & a(n){\cdot}X(S(S(n))) \lhd \mathsf{T} \rhd \delta \end{array}$$

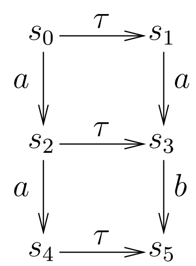
   Prove that in this case the confluence formula regarding the τ-summand is false. Also give an example consisting of transitions of $X(0)$ to show that the τ-summand does not give rise to confluent τ-transitions.

   (c) Consider the following LPE:

$$\begin{array}{lcl} X(n : Nat) & = & \tau{\cdot}X(S(n)) \lhd \mathsf{T} \rhd \delta \\ & + & a{\cdot}X(S(S(n))) \lhd even(n) \rhd \delta \end{array}$$

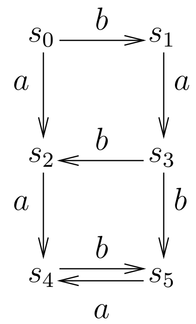
   Prove that in this case the confluence formula regarding the τ-summand is false. Also give an example consisting of transitions of $X(0)$ to show that the τ-summand does not give rise to confluent τ-transitions.

3. (a) Apply the minimisation algorithm modulo branching bisimulation to the process graph below. Describe the subsequent splits that you perform, and the results of those splits.



(b) What is the maximal collection of confluent τ-transitions in the process graph of (a)? Explain your answer.

(c) Describe in detail how the model-checking algorithm computes for which states in the process graph below the ACTL formule $\mathtt{E}\,(\mathtt{EX}_b\mathsf{T})\,\mathtt{U}\,(\mathtt{EX}_a\mathtt{EX}_b\mathsf{T})$ is true.



(d) For each state in the process graph in (c) there exists an ACTL formule that is only true in this state. Give such a characterizing ACTL formula for each of the states.