**Exercises: Plain Petri nets (session 1)**

**Exercise 1 (Car Assembly) – taken from [www.workflowcourse.com](http://www.workflowcourse.com)**
Model as a Petri net the production process shown in the following Bill-Of-Materials.

```
subassembly2  car
2

chair  subassembly1

4
chassis

wheel
```

Model the chemical reaction $\text{C}_2\text{H}_5\text{OH} + 3 * \text{O}_2 \rightarrow 2 * \text{CO}_2 + 3 * \text{H}_2\text{O}$ as a Petri net. Assume that there are two steps: first each molecule is disassembled into its atoms and then these atoms are assembled into other molecules.

**Exercise 3 (Vending machine)**
We consider a vending machine that sells chocolate bars. The machine sells small bars for 15 cents and large bars for 20 cents. The machine accepts coins of 5 cents, 10 cents and 20 cents. The machine is not able to return coins. Accordingly, the machine never allows a user to insert more than 20 cents. Once the user has put 20 cents, the machine will not accept any more coins. Instead, it will only allow the user to push the button and get a large bar. If the user has inserted 15 cents, he/she may opt to get a small bar, or put an additional 5 cents to get a large bar.

Capture this process as a state machine and as a Petri net.

A circular rail network consists of four tracks. Each track is in one of the following states:

- Busy, i.e., there is a train on the track.
- Claimed, i.e., a train has successfully requested access to the track.
- Free, i.e., neither busy nor claimed.

There are two trains driving on the circular track. Initially, one train is in track 1 while the other is in track 3. Trains can move from track 1 to 2, from track 2 to 3, from 3 to 4 and from 4 back to 1.
The track where a train resides is "busy". To move to the next track a train first claims the next track. Only free tracks can be claimed. Busy tracks are released the moment the train moves to another track. One can abstract from the identity of trains only the state of the rail network is considered. Model the dynamic behavior of the rail network in terms of a Petri net.

Exercise 5. Milner’s Scheduler

Let’s specify a scheduler for a set of $n$ agents $P_1,...,P_n$. Each agent $P_i$ performs a task repeatedly. The scheduler must ensure that they begin the task in cyclic order starting with $P_1$. The execution of the tasks can occur overlap in time—for example $P_2$ can begin before $P_1$ finishes—but the scheduler is required to ensure that each agent finishes one performance before it begins another.

We assume that $P_i$ requests task initiation by means of an action $a_i$ and it signals completion of the task by an action $b_i$. The scheduler can then be specified by requiring that:

1. It must perform $a_1, \ldots, a_n$ cyclically, starting with $a_1$.
2. It must perform $a_i$ and $b_i$ alternately, for each $i$.

However, a scheduler which imposes a fixed sequence, say $a_1b_1a_2b_2 \ldots$, is not good enough, the scheduler must allow any sequence of actions compatible with the conditions (1) and (2) above. For example, for $n = 2$, the sequences $a_1a_2b_1b_2a_1$ and $a_1b_1a_2b_1a_1$ are compatible with the specification, but the sequences $a_1b_1a_1$ and $a_1b_1a_2a_2$ are not.

For $n = 2$ agents, give a Petri net $(N, M_0)$ which models a scheduler satisfying the given requirements. The Petri net must have transitions $\{a_1, a_2, b_1, b_2\} \subseteq T$. Firing one of these transitions is equivalent to the execution of the corresponding action. Further, for any firing sequence $\sigma$ enabled at $M_0$, the requirements above must hold when interpreting the firing sequence as a sequence of actions. The Petri net may also have additional transitions, which do not correspond to an action and may occur at any time.

Acknowledgment: Exercise designed by Javier Esparza (TU Munich)
**Additional Exercise. (optional)**

Consider a circuit for traffic lights, which controls two traffic lights at a pedestrian crossing: one traffic light for the car traffic, which has four states (green, yellow, red, red-yellow – in this order), and one for the pedestrians, which has two states (green and red). In addition, there is a button, which can be pressed by the pedestrians. The traffic light circuit must have the following properties: the traffic light for pedestrians is only green, when the traffic light for cars is red; the traffic light for cars is only green, when the traffic light for pedestrians is red; the traffic light for cars only switches to red, when the button is pressed (the button stays pressed, until the pedestrian light switches to red again).

Model the above scenario with a Petri net containing at least the following places: Gr (green for cars), Y (yellow), R (red for cars), RY (red-yellow), PR (red for pedestrians), PG (green for pedestrians). Furthermore, there should be a place B which indicates whether the button was pressed. In addition, you may need to add more places.

*Acknowledgment: Exercise designed by Barbara König (University of Duisburg-Essen)*