Assignment 10: Petri Nets

ETH Zurich

1 Modelling Systems as Petri Nets

1.1 Background

These tasks have been adapted from [1] and are about modelling concurrent systems as Petri nets. We will use the elementary Petri net notation as given in the lecture slides:


1.2 Task

1. Consider the cookie vending machine Petri net we constructed in the lecture:

   ![Petri Net Diagram](image)

   Extend the Petri net such that:
   - at most one token can be in the coin slot place at any time; and
   - at most one token can be in the signal place at any time.
2. Consider the Petri net we constructed for mutual exclusion:

```
waiting_1

semaphore

waiting_2

local_1

local_2
```

For each process $i$ add a place noncritical$_i$ that holds a token if and only if that process $i$ is not in its critical region.

3. Model as an elementary Petri net a gambling machine that has the following characteristics:
   - a player can insert a coin, which should reach a “cash box”;
   - at this stage, the machine enters a state in which it pays out a coin from the (same) cash box an arbitrary number of times (including zero); and
   - eventually, the machine stops giving out coins and becomes ready for another game.

### 1.3 Sample Solutions

1. We add two new places which both contain a token in the initial marking:
2. A possible solution:

3. Below is a possible solution. The specification does not specify an initial number of coins, so we arbitrarily chose 7:
2 Reachability Graphs and Unfoldings

2.1 Background

These tasks have been partly adapted from [2], and are about the two semantics we assigned to Petri nets in the lecture: first, the semantics based on interleaving; second, the semantics based on true concurrency.

2.2 Task

1. Consider the Petri net below that models a producer-consumer scenario for a bounded buffer of capacity 1:

Construct a reachability graph for the Petri net, and prove that the buffer is never both full and empty.
2. For the Petri net below, iteratively construct its unfolding until there are 9 transitions:

![Petri Net Diagram]

2.3 Sample Solutions

1. Let a marking $M$ of the Petri net be expressed by the vector:

$$( M(\text{produce}) \ M(\text{wait}_p) \ M(\text{buffer\_empty}) \ M(\text{buffer\_full}) \ M(\text{consume}) \ M(\text{wait}_c) ) .$$

For such an encoding, we get the following reachability graph:

![Reachability Graph Diagram]

The buffer is never both full and empty because the graph contains no marking with $M(\text{buffer\_empty}) = M(\text{buffer\_full}) = 1.$
2. The unfolding, cut off after nine iterations, is as below:

(Note that there are other solutions, e.g. if reachable markings are searched for in a “depth-first” manner.)

References
