

Solution

Petri nets – Homework 1

Discussed on Thursday 21st April, 2016.

For questions regarding the exercises, please send an email to meyerphi@in.tum.de or just drop by at room 03.11.042.

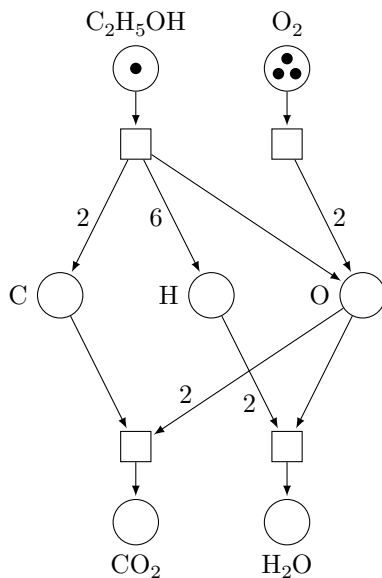
Exercise 1.1 Alcohol burning

Model the chemical reaction $C_2H_5OH + 3O_2 \rightarrow 2CO_2 + 3H_2O$ as a Petri net. Use weighted arcs as needed.

Assume that there are two steps: first each molecule is disassembled into its atoms and then these atoms are assembled into other molecules. The net should have places for each of the molecules, and the marking where there are two tokens in the place CO_2 and three tokens in the place H_2O should be reachable if and only if there are at least one token in the place C_2H_5OH and three tokens in the place O_2 in the initial marking.

Acknowledgement: Exercise taken from www.workflowcourse.com

Solution: (This solution is just an example. There are many possible solutions to any given modeling exercise)



Exercise 1.2 Manufacturing process

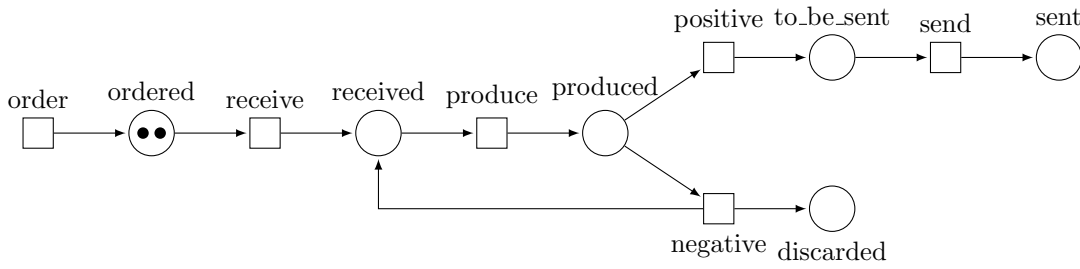
In a “just-in-time” manufacturing system, items can be ordered to be produced, which is signaled by an “order” event. An order is acknowledged by a “receive” event, and then manufactured by a “produce” event. After that, the item is checked by quality control and can either be ready to be sent (event “positive”) or be discarded (event “negative”). In the positive case, the item is sent (event “send”) to the customer. In the negative case, a replacement item is scheduled to be produced.

An item can be in any of the states “ordered”, “received”, “produced”, “to_be_sent”, “sent” “discarded”. Suppose that there are initially two orders.

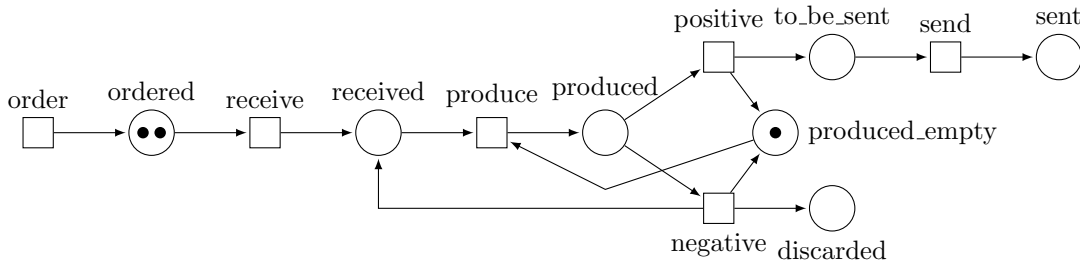
- Model the system as a Petri net, where there are transitions for each event and places for each possible state of an item. You may have additional places and transitions. Do not use weighted arcs.
- Now adapt your model such that there is at most one item in the “produced” state at any time, and further items have to wait for that item to be quality checked before they are produced.

Solution: (This solution is just an example. There are many possible solutions to any given modeling exercise)

(a)



(b) The capacity constraint can be enforced by introducing an additional place that has a token iff there is no token in “produced” and adding arcs from the transitions that affect “produced” accordingly.



Exercise 1.3 Nets and subnets

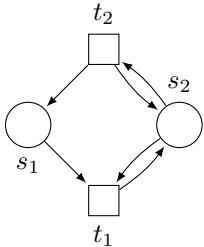
Consider the net $N = (S, T, F)$ defined by $S = \{s_1, s_2\}$, $T = \{t_1, t_2\}$ and $F = \{(s_1, t_1), (t_1, s_2), (s_2, t_1), (s_2, t_2), (t_2, s_2), (t_2, s_1)\}$.

(a) Draw the net N .

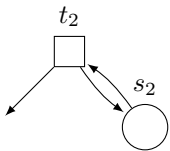
(b) Is the net $N' = (S', T', F')$ defined by $S' = \{s_2\}$, $T' = \{t_2\}$ and $F' = \{(s_2, t_2), (t_2, s_2), (t_2, s_1)\}$ a subnet of N ?

Solution:

(a)

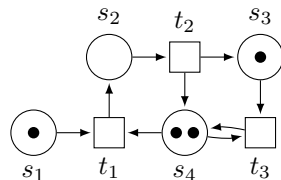


(b) No, N' is not even a net, as $(t_2, s_1) \in F' \not\subseteq (S' \times T') \cup (T' \times S')$.



Exercise 1.4 Reachable markings

Consider the following Petri net:



(a) Give the preset and postset of each place and each transition.

- (b) Which transitions are enabled at M_0 ?
- (c) Construct the reachability graph of the Petri net.
- (d) Does the number of reachable markings increase or decrease if we remove i) place s_1 and its adjacent arcs ii) place s_3 and its adjacent arcs?

Acknowledgment: Exercise designed by Wil van der Aalst et al. (TU Eindhoven).

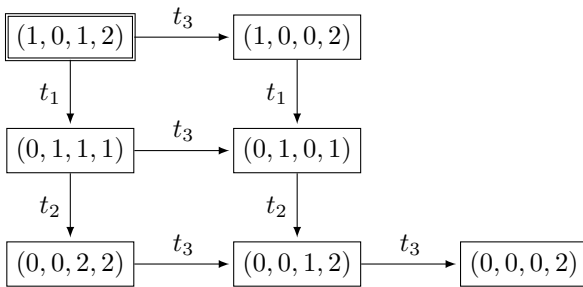
Solution:

(a)

$\bullet s_1 = \{ \}$	$s_1 \bullet = \{ t_1 \}$	$\bullet t_1 = \{ s_1, s_4 \}$	$t_1 \bullet = \{ s_2 \}$
$\bullet s_2 = \{ t_1 \}$	$s_2 \bullet = \{ t_2 \}$	$\bullet t_2 = \{ s_2 \}$	$t_2 \bullet = \{ s_3, s_4 \}$
$\bullet s_3 = \{ t_2 \}$	$s_3 \bullet = \{ t_3 \}$	$\bullet t_3 = \{ s_3, s_4 \}$	$t_3 \bullet = \{ s_4 \}$
$\bullet s_4 = \{ t_3 \}$	$s_4 \bullet = \{ t_1, t_3 \}$		

(b) $\text{enabled}(M_0) = \{ t_1, t_3 \}$.

(c) With the markings as a fixed vector $M = (s_1, s_2, s_3, s_4)$, the reachability graph is as follows. The initial marking is marked with a double border.

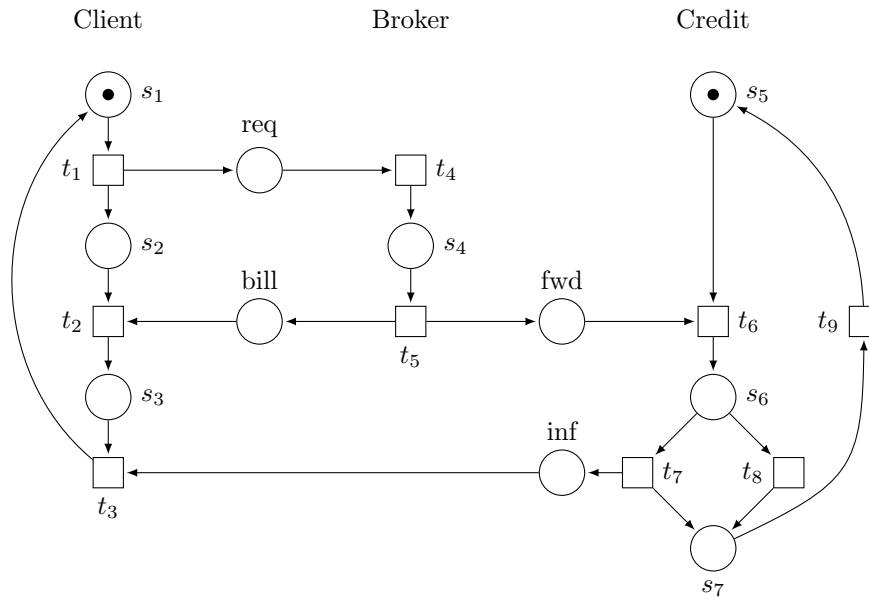


- (d) i) The number of reachable markings increases, as all previously reachable markings without the s_1 component are still reachable and all different, but now t_1 is enabled for instance at $M = (s_2, s_3, s_4) = (1, 0, 1)$, leading to the new marking $M' = (2, 0, 1)$.
- ii) The number of reachable markings decreases, as now firing t_3 does not change the marking, and the other transitions do not become enabled at any other marking.

Exercise 1.5 Model analysis (with tools)

For the Petri net below modelling a bank contract, check if the following properties hold. You may use tools such as **PIPE**, **APT** or **Lola** to check the properties. The input files for the net are given in appropriate format on the homepage. You may then use the tools with the integrated modules to check properties or construct the reachability graph.

- (a) **Boundedness:** Is the Petri net bounded, i.e., for every place s , is there a number $b \geq 0$ such that $M(s) \leq b$ for every reachable marking M ?
- (b) **Liveness:** Is the net live, i.e., for every reachable marking M and every transition t , is there a marking M' reachable from M that enables t ?
- (c) **Deadlock freedom:** Is the Petri net deadlock-free, i.e., is there a reachable marking M that enables no transitions?



Solution:

(a) **Boundedness:** The Petri net is bounded.

With **PIPE**: Launch PIPE and open the file `exercise-1.5.xml`. After starting the “State Space Analysis” module and clicking on “Analyse”, the result shows: “**Bounded:** true”. This can be verified by constructing the reachability graph with the “Reachability/Coverability Graph” module.

With **APT**: Execute `java -jar apt.jar bounded exercise-1.5.apt`. The result shows “bounded: Yes”.

With **Lola**: Lola can only check boundedness for a given place. Execute `lola --search=cover --encoder=full --formula="AG s < oo" exercise-1.5.lola` to show that place s is bounded. This check can be repeated for every place $s \in S$.

(b) **Liveness:** The Petri net is not live.

With **PIPE**: PIPE has no module for directly checking liveness, however the reachability graph can be inspected for bounded nets.

With **APT**: Execute `java -jar apt.jar strongly.live exercise-1.5.apt`. The result shows “strongly.live: No”. *Note:* APT can only check liveness for bounded nets.

With **Lola**: Lola can only check liveness for a given transition. Execute `lola --formula="AGEF FIREABLE(t1)" exercise-1.5.lola` to show that transition t is live. This check can be repeated for every transition $t \in T$.

(c) **Deadlock freedom:** The net is not deadlock free (it has a deadlock).

With **PIPE**: Launch PIPE and open the file `exercise-1.5.xml`. After starting the “State Space Analysis” module and clicking on “Analyse”, the result shows: “**Deadlock:** true”. The dead marking is shown in the reachability graph in red.

With **APT**: APT has no module to directly check for deadlocks.

With **Lola**: Execute `lola --formula="EF DEADLOCK" exercise-1.5.lola`. The result shows “yes”. A witness state and path can be produced by additionally passing the parameters `-s` and `-p`.

Exercise 1.6 Exchange Lemma

Let u and v be transitions of a net satisfying $\bullet u \cap \bullet v = \emptyset$. Show: If $M \xrightarrow{vu} M'$ then $M \xrightarrow{uv} M'$.

Solution: We prove the following: If $M \xrightarrow{v} L \xrightarrow{u} M'$ for arbitrary markings M, L, M' and transitions u, v with $\bullet u \cap \bullet v = \emptyset$, then $M \xrightarrow{v} L' \xrightarrow{u} M'$ for some marking L' .

We show:

- $M(s) \geq 1$ for every $s \in \bullet v$
- $M(s) \geq 1$ for every $s \in \bullet u$
- $M(s) \geq 2$ for every $s \in \bullet v \cap \bullet u$

Assume $s \in \bullet v$. Then $M(s) \geq 1$ because M enables v .

Assume $s \in \bullet u$. Then $L(s) \geq 1$ because L enables u . Since $\bullet u \cap v^\bullet = \emptyset$ we have $s \notin v^\bullet$. So the number of tokens on s does not increase by the occurrence of v , i.e. $M(s) \geq L(s)$. Therefore $M(s) \geq 1$.

Assume $s \in \bullet v \cap \bullet u$. Again, $L(s) \geq 1$ because L enables u and $s \in \bullet u$. Since $s \in \bullet v$ and $s \notin v^\bullet$ we get $L(s) = M(s) - 1$. So $M(s) \geq 2$.

We have that M enables u because $M(s) \geq 1$ for every place $s \in \bullet u$. Let $M \xrightarrow{u} L'$, we show that L' enables v . Let s be a place of $\bullet v$. If $s \notin \bullet u$ then $L'(s) \geq M(s) \geq 1$. If $s \in \bullet u$ then $L'(s) \geq M(s) - 1$ and $M(s) \geq 2$ whence $L'(s) \geq 1$. So L' marks every place in $\bullet v$, and therefore L' enables v .

The sequences uv and vu have the same effect on the number of tokens in each place, therefore both lead from M to the same marking, namely to M' . This completes the proof of the claim.