Chair of Foundations of Software Reliability and Theoretical Computer Science
School of Computation, Information and Technology
Technical University of Munich

Eexam
Place student sticker here

Note:

- During the attendance check a sticker containing a unique code will be put on this exam.
- This code contains a unique number that associates this exam with your registration number.
- This number is printed both next to the code and to the signature field in the attendance check list.

Petrinetze

Exam:	IN2052 / Retake	**Date:**	Thursday 1st October, 2026
Examiner:	Prof. Dr. Javier Esparza	**Time:**	12:00 – 13:15

Sample Solution

Working instructions

- This exam consists of **8 pages** with a total of **4 problems**.
  Please make sure now that you received a complete copy of the exam.
- The total amount of achievable credits in this exam is **44 credits**, including **4 bonus credits**.
- To pass the exam, 17 credits are *sufficient*.
- Detaching pages from the exam is prohibited.
- Allowed resources:
  - one **analog dictionary** English ↔ native language
- **Answers are only accepted if the solution approach is documented.** Give a reason for each answer unless explicitly stated otherwise in the respective subproblem.
- Do not write with red or green colors nor use pencils.
- Physically turn off all electronic devices, put them into your bag and close the bag.

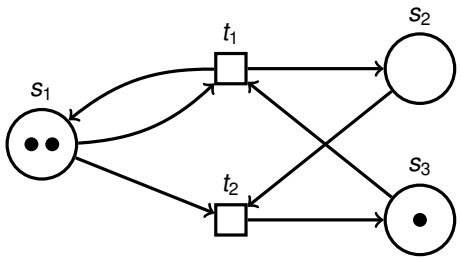
Left room from _____ to _____ / Early submission at _____

In all problems, Petri nets do not have arc weights.

Problem 1 Coverability and well-quasi-orders (8 credits)

0 1 2 3

a) Perform a step of the backwards coverability algorithm starting with $m_0 = \{(2,0,1)\}$. Recall that this requires computing $pre(R[t_1] \wedge m_0, t_1)$ and $pre(R[t_2] \wedge m_0, t_2)$, and then computing $m_1 = \min(m_0 \cup \bigcup_{t \in T} pre(R[t] \wedge m_0, t))$. You only need to give the results.



$pre(R[t_1] \wedge m_0, t_1)$ = {(2,0,2)} $pre(R[t_2] \wedge m_0, t_2)$ = {(3,1,0)} m_1 = {(2,0,1),(3,1,0)}

0 1 2 3 4 5

b) Let $A_1, A_2, ...$ be an infinite sequence of **upward-closed** sets of markings of a net N such that $A_1 \subseteq A_2 \subseteq ....$ Using Dickson's lemma, show that this sequence stabilizes, i.e. that there exists $n_0 \in \mathbb{N}$ such that for all $n \geq n_0$, we have $A_n = A_{n_0}$.
Dickson's lemma: Let $M_1, M_2, ...$ be an infinite sequence of markings of a net N. Then there exist two indices $i < j$ such that $M_i \leq M_j$.

We prove the statement by contraposition. If the sequence does not stabilize, then there exists a subsequence $A_{i_1}, A_{i_2}, ...$ which is strictly increasing. For simplicity, we rename the elements of that subsequence to $A_1, A_2, ...$ Now let M_n be some marking in $A_n \setminus A_{n-1}$ and let $i < j$ be arbitrary indices. Since A_i is upward-closed, $M_i \in A_i$ and $M_j \notin A_i$, we have $M_i \not\leq M_j$. Since i and j were arbitrary, the sequence $M_1, M_2, ...$ violates Dickson's lemma.

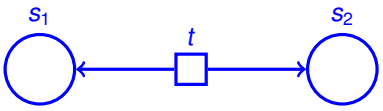
Sample Solution

Problem 2 Construction (15 credits)

For this exercise, it is sufficient to draw the Petri net. You are welcome to add explanations, but a correct Petri net already gives full points.

a) Construct an unbounded Petri net (N, M_0) with an S-invariant I such that $I(s) \neq 0$ for every place s of N.

0 1 2 3



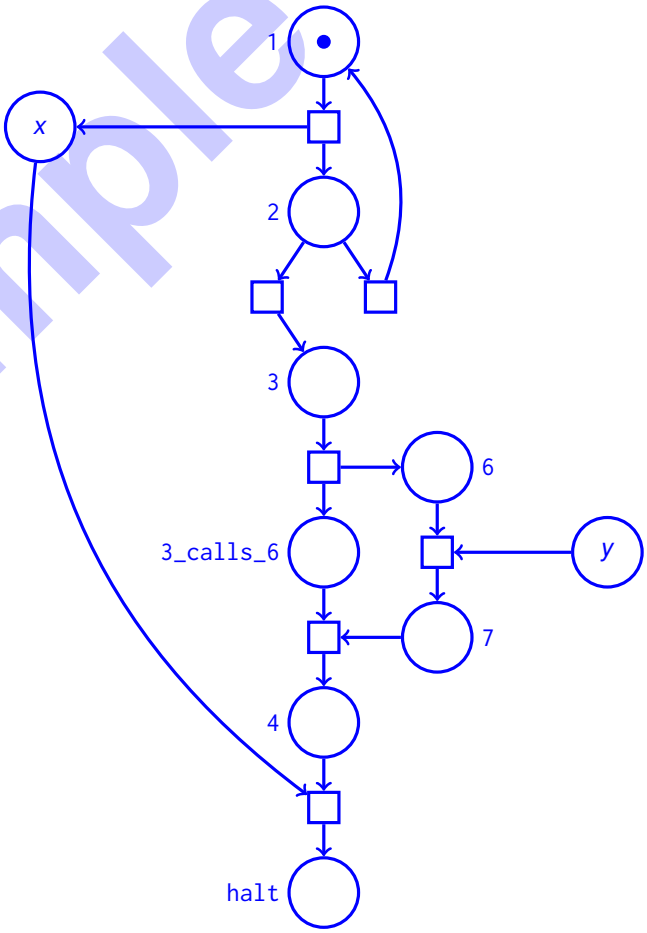
$I = (1, -1)$ is an S-invariant.

b) **Bonus points:** Construct the Petri net corresponding to the net program

1: $x := x + 1$**;**
2: goto 1 or goto 3
3: gosub 6;
4: $x := x - 1$**;**
5: halt
6: $y := y - 1$**;**
7: return

0 1 2 3 4

This is the Petri net obtained by applying the translation described in pages 131-133 of the lecture notes.

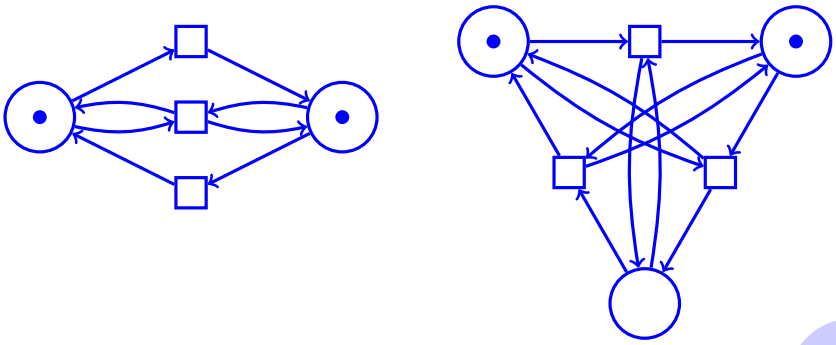


Sample Solution

0 1 2 3 4

c) Construct a live and bounded Petri net such that there is a place which does not belong to any S-component.

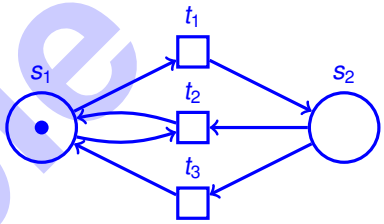
Here are two examples:



In both of these nets, taking a single place does not yield an S-component because it will include transitions with no output places, and taking two (or three) places does not yield an S-component because it will include transitions with two output places.

0 1 2 3 4

d) Construct a non-live Petri net in which every proper siphon contains an initially marked trap.



$\{s_1\}$ and $\{s_2\}$ are not siphons because of t_3 and t_1, respectively. $\{s_1, s_2\}$ is both a siphon and a marked trap. The Petri net is not live because t_2 is dead.

Sample Solution

Problem 3 Reductions (10 credits)

a) Given a net N and two markings M_1, M_2, show how to construct a Petri net (N', M_0') and a marking M' such that

(N, M_1) and (N, M_2) have the same set of reachable markings **iff** M' is reachable in (N', M_0').

You do not need to give a formal definition of N' and M_0', but describe them sufficiently. Sketch a proof of correctness of the construction. Your construction should be implementable as a program that runs in polynomial time in the size of N, M_1 and M_2.

0 1 2 3 4 5

We define N' as two copies of N side by side. (To connect them, we can e.g. add a transition with all places as input and output places.) Let M_0' be M_1 in the left copy and M_2 in the right copy, and let M' be M_2 in the left copy and M_1 in the right copy. We have

M' is reachable from M_0' in N'
iff M_1 is reachable from M_2 and M_2 is reachable from M_1 in N
iff (N, M_1) and (N, M_2) have the same set of reachable markings.

b) Given a Petri net (N, M_0) with $N = (S, T, F)$, show how to construct a Petri net (N', M_0') and a transition t' such that

(N, M_0) has an infinite firing sequence **iff** (N', M_0') has a firing sequence which contains t' infinitely often.

Indicate clearly which are the sets of places, transitions and arcs of N', and which is the marking M_0'. Sketch a proof of correctness of the construction. Your construction should be implementable as a program that runs in polynomial time in the size of (N, M_0).

0 1 2 3 4 5

$N' = (S', T', F')$ is defined by $S' = S \cup \{s, s'\}$, $T' = T \cup \{t'\}$, and

$$F' = F \cup \{(s, t), (t, s') \mid t \in T\} \cup \{(s', t'), (t', s)\}.$$

M_0' is the marking that coincides with M_0 on S and puts one token into s and no tokens into s'.

Every original transition now also moves a token from s to s', and t' puts it back. Therefore, every firing sequence σ of (N, M_0) corresponds to a firing sequence of (N', M_0') which contains t' $|\sigma|$ times. It follows that (N, M_0) has an infinite firing sequence iff (N', M_0') has a firing sequence which contains t infinitely often.

Sample Solution

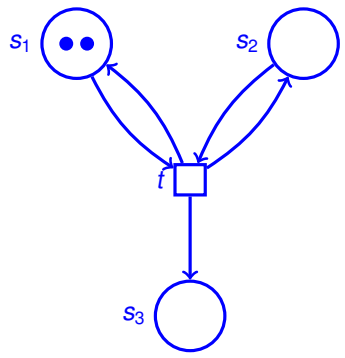
Problem 4 True or false (11 credits)

For each of the following statements, decide whether the statement is true. If it is true, give a proof. If it is false, give a counterexample.

0 1 2 3

a) Let N be a net and let M_0, M_0' be markings such that (N, M_0) and (N, M_0') are bounded and $M := \frac{M_0 + M_0'}{2}$ is a marking. Then (N, M) is bounded.

False. Counterexample:



This Petri net is bounded at the markings $(2, 0, 0)$ and $(0, 2, 0)$, but not at the marking $(1, 1, 0)$.

0 1 2 3 4

b) Let (N, M_0) be a Petri net such that the zero marking is reachable, and M_0 is not the zero marking. Then there exist infinitely many markings M such that the zero marking is reachable in (N, M).

True. We prove by induction that the zero marking is reachable from nM for all $n \in \mathbb{N}$. For $n = 0$, this is trivial. For the induction step, consider $(n+1)M$. Since the zero marking is reachable from M, nM is reachable from $(n+1)M$ by the monotonicity lemma. By the induction hypothesis, the zero marking is reachable from nM, and therefore also from $(n+1)M$.

Sample Solution

c) Let (N, M_0) be a bounded Petri net. If an infinite firing sequence is enabled at M_0, then N has a semi-positive T-invariant.

0 1 2 3 4

True. Since the Petri net is bounded, only finitely many markings are reachable from M, so firing the infinite firing sequence will lead to the same marking, say M, twice. That is, there exists a nonempty firing sequence σ such that $M \xrightarrow{\sigma} M$. By the fundamental property of T-invariants, $\vec{\sigma}$ is a T-invariant of N. It is semi-positive because it is the Parikh vector of a nonempty firing sequence.

Additional space for solutions–clearly mark the (sub)problem your answers are related to and strike out invalid solutions.

Sample Solution

Sample Solution