1 Modeling Using Petri Nets

Exercise 1 (Traffic Lights). Consider again the traffic lights example from the lecture notes:

1. How can you correct this Petri net to avert unwanted behaviours (like $r \rightarrow ry \rightarrow rr$) in a 1-safe manner?
2. Extend your Petri net to model two traffic lights handling a street intersection.

Exercise 2 (Producer/Consumer). A producer/consumer system gathers two types of processes:

- **producers** who can make the actions *produce* ($p$) or *deliver* ($d$), and
- **consumers** with the actions *receive* ($r$) and *consume* ($c$).

All the producers and consumers communicate through a single unordered channel.

1. Model a producer/consumer system with two producers and three consumers. How can you modify this system to enforce a maximal capacity of ten simultaneous items in the channel?
2. An *inhibitor arc* between a place $p$ and a transition $t$ makes $t$ firable only if the current marking at $p$ is zero. In the following example, there is such an inhibitor arc between $p_1$ and $t$. A marking $(0, 2, 1)$ allows to fire $t$ to reach $(0, 1, 2)$, but $(1, 1, 1)$ does not allow to fire $t$. 

1
Using inhibitor arcs, enforce a priority for the first producer and the first consumer on the channel: the other processes can use the channel only if it is not currently used by the first producer and the first consumer.

2 Model Checking Petri Nets

Exercise 3 (Upper Bounds). Let us fix a Petri net $\mathcal{N} = \langle P, T, F, W, m_0 \rangle$. We consider as usual propositional LTL, with a set of atomic propositions $AP$ equal to $P$ the set of places of the Petri net. We define proposition $p$ to hold in a marking $m$ in $\mathbb{N}^P$ if $m(p) > 0$.

The models of our LTL formulae are computations $m_0m_1 \cdots$ in $(\mathbb{N}^P)^\omega$ such that, for all $i \in \mathbb{N}$, $m_i \rightarrow_{\mathcal{N}} m_{i+1}$ is a transition step of the Petri net $\mathcal{N}$.

1. We want to prove that state-based LTL model checking can be performed in polynomial space for 1-safe Petri nets. For this, prove that one can construct an exponential-sized B"uchi automaton $B_{\mathcal{N}}$ from a 1-safe Petri net that recognizes all the infinite computations of $\mathcal{N}$ starting in $m_0$.

2. In the general case, state-based LTL model checking is undecidable. Prove it for Petri nets with at least two unbounded places, by a reduction from the halting problem for 2-counter Minsky machines.

3. We consider now a different set of atomic propositions, such that $\Sigma = 2^{AP}$, and a labeled Petri net, with a labeling homomorphism $\lambda : T \rightarrow \Sigma$. The models of our LTL formulae are infinite words $a_0a_1 \cdots$ in $\Sigma^\omega$ such that $m_0 \xrightarrow{t_0}_{\mathcal{N}} m_1 \xrightarrow{t_1}_{\mathcal{N}} m_2 \cdots$ is an execution of $\mathcal{N}$ and $\lambda(t_i) = a_i$ for all $i$.

Prove that action-based LTL model checking can be performed in polynomial space for labeled 1-safe Petri nets.

3 Coverability Graphs

Exercise 4 (Dickson’s Lemma). A quasi-order $(A, \leq)$ is a set $A$ endowed with a reflexive and transitive ordering relation $\leq$. A well quasi order (wqo) is a quasi order $(A, \leq)$ s.t., for any infinite sequence $a_0a_1 \cdots$ in $A^{\omega}$, there exist indices $i < j$ with $a_i \leq a_j$.

1. Let $(A, \leq)$ be a wqo and $B \subseteq A$. Show that $(B, \leq)$ is a wqo.
2. Show that \((\mathbb{N} \cup \{\omega\}, \leq)\) is a wqo.

3. Let \((A, \leq)\) be a wqo. Show that any infinite sequence \(a_0 a_1 \cdots \) in \(A^\omega\) embeds an infinite increasing subsequence \(a_{i_0} \leq a_{i_1} \leq a_{i_2} \leq \cdots\) with \(i_0 < i_1 < i_2 < \cdots\).

4. Let \((A, \leq_A)\) and \((B, \leq_B)\) be two wqo's. Show that the cartesian product \((A \times B, \leq)\), where the product ordering is defined by \((a, b) \leq (a', b')\) iff \(a \leq_A a'\) and \(b \leq_B b'\), is a wqo.

Exercise 5 (Coverability Graph). The coverability problem for Petri nets is the following decision problem:

**Instance:** A Petri net \(\mathcal{N} = (P, T, F, W, m_0)\) and a marking \(m_1\) in \(\mathbb{N}^P\).

**Question:** Does there exist \(m_2\) in \(\text{reach}_\mathcal{N}(m_0)\) such that \(m_1 \leq m_2\)?

For 1-safe Petri nets, coverability coincides with reachability, and is thus PSPACE-complete.

One way to decide the general coverability problem is to use Karp and Miller's coverability graph (see the lecture notes). Indeed, we have the equivalence between the two statements:

i. there exists \(m_2\) in \(\text{reach}_\mathcal{N}(m_0)\) such that \(m_1 \leq m_2\), and

ii. there exists \(m_3\) in \(\text{CoverabilityGraph}_\mathcal{N}(m_0)\) such that \(m_1 \leq m_3\).

1. In order to prove that [i] implies [ii], we will prove a stronger statement: for a marking \(m\) in \((\mathbb{N} \cup \{\omega\})^P\), write \(\Omega(m) = \{p \in P \mid m(p) = \omega\}\) be the set of \(\omega\)-places of \(m\).

   Show that, if \(m_0 \xrightarrow{u} m_2\) in the Petri net \(\mathcal{N}\) for some \(u\) in \(T^*\), then there exists \(m_3\) in \((\mathbb{N} \cup \{\omega\})^P\) such that \(m_2(p) = m_3(p)\) for all \(p\) in \(P \setminus \Omega(m_3)\) and \(m_0 \xrightarrow{u'} G\ m_3\) in the coverability graph.

2. Let us prove that [ii] implies [i]. The idea is that we can find reachable markings that agree with \(m_3\) on its finite places, and that can be made arbitrarily high on its \(\omega\)-places. For this, we need to identify the graph nodes where new \(\omega\) values were introduced, which we call \(\omega\)-nodes.

   (a) The threshold \(\Theta(u)\) of a transition sequence \(u\) in \(T^*\) is the minimal marking \(m\) in \(\mathbb{N}^P\) s.t. \(u\) is enabled from \(m\). Show how to compute \(\Theta(u)\). Show that \(\Theta(u \cdot v) \leq \Theta(u) + \Theta(v)\) for all \(u, v\) in \(T^*\).

   (b) Recall that an \(\omega\) value is introduced in the coverability graph thanks to Algorithm [i].

   Let \(\{v_1, \ldots, v_k\}\) be the set of “\(v\)” sequences found on line [i] of the algorithm that resulted in adding at least one \(\omega\) value to \(m'\) on line [i] during a single call.
repeat
  saved ← m'
foreach m'' ∈ V s.t. ∃v ∈ T⁺, m'' ↠₅ G m do
  if m'' < m' then
    m' ← m' + ((m' − m'') · ω)
until saved = m'
return m'

Algorithm 1: ADDOMEGAS(m, m', V)

to ADDOMEGAS(m, m', V) on line 8 of the COVERABILITYGRAPH algorithm from the course notes. Let w = v₁ · · · vₗ. Show that, for any k in N, the marking ν_k defined by
\[ ν_k(p) = \begin{cases} m'(p) & \text{if } p ∈ P \setminus Ω(m) \\ Θ(w^k)(p) & \text{if } p ∈ Ω(m) \end{cases} \]

allows to fire w^k. How does the marking ν'_k with ν_k w^k − − − − − − − − − − − − − − → G ν'_k compare to ν_k?

(c) Prove that, if m₀ ↠₅ G m₃ for some u in T* in the coverability graph and m' in NΩ(m₃) is a partial marking on the places of Ω(m₃), then there are
• n in N,
• a decomposition u = u₁ u₂ · · · uₙ₊₁ with each uᵢ in T* (where the markings µᵢ reached by m ↠₅ G µᵢ for i ≤ n have new ω values),
• sequences w₁, . . . , wₙ in T⁺,
• numbers k₁, . . . , kₙ in N,
such that m₀ u₁ w₁ k₁ u₂ w₂ k₂ · · · uₙ wₙ kₙ uₙ₊₁ − − − − − − − − − − − − − − → G m₂ with m₂(p) = m₃(p) for all p in P \ Ω(m₃) and m₂(p) ≥ m'(p) for all p in Ω(m₃).

Exercise 6 (Decidability of Model-checking Action-based LTL).

1. Let N be Petri net, G its coverability graph, and m some marking in Nᵐ. An infinite computation is a sequence m₀m₁ · · · in (Nᵐ)ᵐ where for all i ∈ N, mᵢ ↠₅ mᵢ₊₁ is a transition step. The effect Δ(u) of a transition sequence u in T* is defined by Δ(ε) = 0P and Δ(ut) = Δ(u) − W(P, t) + W(t, P).

Show that there exists an infinite computation s.t. m ≤ mᵢ for infinitely many indices i iff there exists an accessible loop m' ↠₅ G m' in G s.t. m ≤ m' and Δ(v) ≥ 0P.

2. Show that action-based LTL model-checking is decidable for labeled Petri nets.