Concurrency Theory

11. Lecture: Petri Nets – Problems and Decision Procedures

Dr. Stephan Mennicke

Institute for Theoretical Computer Science
Knowledge-Based Systems Group

June 30, 2025

TECHNISCHE UNIVERSITÄT DRESDEN

International Center for Computational Logic

Last Lecture

- an introduction to Petri nets
  - syntax (nets)
  - semantics (markings, enabledness, firing rule)
  - monotonicity of the firing rule
  - (non-)extensions (place capacities, arc weights, inhibitor arcs)

Today

- Petri net properties and problems
- decision procedures

Next Week

- Petri net languages (**promise**)
- Petri net complexity results

Petri Nets – System Properties

Let $N = (P, T, F, m_0)$ be a Petri net with at least one place and one transition.

N is *deadlock-free* if every reachable marking enables at least one transition (i.e., no dead marking is reachable in N).

N is *live* if for every reachable marking m and every transition t, there is a marking m' reachable from m that enables t.

N is *bounded* if for every place $p \in P$ there is a number $b \geq 0$ such that $m(p) \leq b$ for everey reachable marking m.

m_0 is a *bounded marking of* N if N is bounded.

Petri Nets – System Properties

The *bound* of a place p is the number

$$\max\{m(s) \mid m \text{ is reachable in } N\}.$$

N is b-bounded if every place has a bound at most b.

It is easy to formulate respective decision problems. Two additional ones are with mentioning.

Reachability Given a Petri net N and a marking m, is m reachable?

Coverability Given a Petri net N and a marking m, is there a reachable marking m' of N such that $m \leq m'$?

Petri Nets – System Properties

Proposition 11.1

1. Liveness implies deadlock-freedom.
2. If N is bounded, then there is a number b such that N is b-bounded.
3. If N is bounded, then it has finitely many reachable markings.

Decision Procedures for Bounded Petri Nets

If a Petri net is bounded, it has finitely many reachable markings.
Hence, the reachability graph can effectively be computed and stored in memory for analysis purposes.

If the reachability graph is available, the following problems have straightforward decision procedures.

b-boundedness

Coverability

Reachability

Deadlock-freedom

What about **liveness**?

Deciding Liveness for Bounded Petri Nets

Let $G = (V, E)$ be the reachability graph of a Petri net N.

Define the equivalence relation $m \Leftrightarrow m'$ between markings $m, m' \in V$ if $m \xrightarrow{\star} m'$ and $m' \xrightarrow{\star} m$.

A *strongly connected component* (SCC) of G is a graph (V', E') where $V' \subseteq V$ is an equivalence class of $\Leftrightarrow$ and $E' = E \cap (V' \times V')$.

SCCs are partially ordered by the following relation $<$:

$$(V', E') < (V'', E'') \text{ iff } V' \neq V'' \text{ and } \forall m' \in V', m'' \in V'' : m' \xrightarrow{\star} m''$$

The *bottom* SCCs of G are the maximal SCCs of G w.r.t. $<$.

Deciding Liveness for Bounded Petri Nets

Theorem 11.2 A bounded Petri net N is live if and only if for every bottom SCC of the reachability graph of N and every transition t, some marking of the SCC enables t.

Algorithms depending on the construction of the reachability graph have exponential worst-case runtime.

Using Savitch's Theorem, complexity of reachability comes down to PSpace-completeness for 1-bounded Petri nets.

But we need *boundedness* in the first place: can we decide boundedness for general Petri nets?

A Simpler Case: b-Boundedness

What about it?

The General Case: Boundedness

Constructing the reachability graph is a semi-decision procedure.

Need a way to detect the unboundedness case by finite means.

Recall

1. every Petri net is a finite object — set of transitions T is finite
2. markings enable at most $|T|$ transitions — $\leq |T|$ successor markings
3. the reachability graph is finitely branching / has finite degree
   - calls for König's Lemma — *again* 😀
4. sequences of markings are sequences of $|P|$-dimensional $\mathbb{N}$-vectors
   - calls for Dickson's Lemma — *again* 😀

Towards a Decision Procedure for Boundedness

Theorem 11.3 A Petri net N is unbounded if and only if there are markings m, ℓ of N such that $\ell \neq \mathbf{0}$ and $m_0 \xrightarrow{\star} m \xrightarrow{\star} m + \ell$.

A Decision Procedure for Boundedness

Theorem 11.4 Boundedness for Petri nets is decidable.

1. Start constructing the reachability graph of N using breadth-first-search.
2. After adding a new marking m' such that there is a marking m already in the graph such that (1) $m \xrightarrow{\star} m'$, (2) $m \leq m'$, and (3) $m \neq m'$, stop the computation and report *UNBOUNDED*
3. Otherwise, the reachability graph construction will eventually terminate. Thus, report *BOUNDED*.

The coverability problem can be solved by similar means (see exercise).

Decision Problems based on Reachability

The reachability problem was open for quite some time (unti Mayr, 1980).

Before 1980, there were decision procedures published based on the existence of a decision procedure for reachability, e.g., **deadlock-freedom**, liveness, or ***language equivalence***.

Route

1. decision procedure for deadlock-freedom
2. decision procedure for language equivalence
3. sketch how to solve reachability

Deciding Deadlock-Freedom

A two-step algorithm. We first reduce deadlock-freedom to another, auxiliary, problem $\boldsymbol{P}$.

Definition 11.5 For a Petri net $N = (P, T, F, m_0)$ and a subset $R \subseteq P$, we look for a reachable marking m such that $m(p) = 0$ for all $p \in R$. The set of all pairs (N, R) for which the property holds is defined as $\boldsymbol{P}$.

Deciding Deadlock-Freedom

Deadlock-freedom can be reduced to $\boldsymbol{P}$: For a Petri net N, define set $\mathcal{S} := \{R \subseteq P \mid \forall t \in T : {}^\bullet t \cap R \neq \emptyset\}$.

We have

1. $\mathcal{S}$ is finite.
2. a marking m of N is dead if and only if the set of places unmarked in m is an element of $\mathcal{S}$.

Decide deadlock-freedom of N as follows: For every $R \in \mathcal{S}$ use the algorithm for $\boldsymbol{P}$ on (N, R) to decide if there is a reachable marking m with $m(p) = 0$ for all $p \in R$.

If N is deadlock-free, the answer is negative in all cases of $\mathcal{S}$ and there are only finitely many cases to check.

Reduce P to Reachability

Let $N = (P, T, F, m_0)$ be a Petri net and $R \subseteq P$.

Construct $N' = (P', T', F', m_0)$ from N as follows:

1. add two new places p_0 and r_0 and put one token on p_0;
2. add transition t_0 together with arcs (p_0, t_0) and (t_0, r_0);
3. add arcs (p_0, t) and (t, p_0) for every transition $t \in T$;
4. for all $p \in P \setminus R$, add a transition t_p together with arcs (p, t_p), (r_0, t_p), and (t_p, r_0);

Reduce P to Reachability

Intuitively,

- t_0 freezes the current marking of N.
- transitions t_p *garbage-collect* the tokens from the set $p \in P \setminus R$.

Finally, $m_{r_0} = \{r_0 \mapsto 1\}$ is reachable if and only if there is a reachable marking m of N such that $m(p) = 0$ for all $p \in R$.

Deciding Language Equivalence

Recall, for Petri nets, bisimilarity is undecidable. In this proof, there was nondeterminism involved.

Definition 11.6 Let Σ be an alphabet. A *labeled Petri net* is a quintuple $N = (P, T, F, m_0, \ell)$ where (P, T, F, m_0) is a Petri net and $\ell : T \to \Sigma$ a labeling function.

We say that a Petri net $N = (P, T, F, m_0, \ell)$ is *(effectively) unlabled* (or deterministic) if ℓ is injective.

Recall next that bisimilarity for deterministic Petri nets is the same as language (i.e., trace) equivalence.

Deciding Language Equivalence

Let $N_1 = (P_1, T_1, F_1, m_1, \ell_1)$ and $N_2 = (P_2, T_2, F_2, m_2, \ell_2)$ be unlabeled Petri nets with disjoint sets of nodes.

Starting from the Petri net

$$N_1 + N_2 = (P_1 \cup P_2, T_1 \cup T_2, F_1 \cup F_2, m_1 + m_2, \ell_1 \cup \ell_2),$$

- add a duplicate transition t' for each transition $t \in T_1 \cup T_2$ with same label, pre-, and postset as t; sets denoted by T_1' and T_2'
- add a fresh place p with arcs $\{p\} \times (T_1 \cup T_2)$ and $(T_1' \cup T_2') \times \{p\}$;
- for each label $a \in \Sigma$, add places p_1^a, p_2^a and for $t \in T_i$ with $\ell_i(t) = a$, add arcs $\left(t, p_j^a\right), \left(p_j^a, u'\right)$ for transition duplicate $u' \in T_j$ with $\ell_j(u') = a$;
- add a transition t_ω with arcs $(p, t_\omega), (t_\omega, p)$

Finally, N_1 and N_2 produce the same sets of traces if and only if the constructed net is deadlock-free.

Deciding Reachability (Sketch)

The original alorithms for reachability (1980, 1982, 1992) are too complicated for presentation in this course.

Our presentation is based on the one presented by Leroux in 2012. The proof of correctness is quite complex, the algorithm itself is simple.

Deciding Reachability (Sketch)

Definition 11.7 A set $X \subseteq \mathbb{N}^k$ is *linear* if there is a *root* $r \in \mathbb{N}^k$ and a finite set $P \subseteq \mathbb{N}^k$ of *periods* such that

$$X = \left\{ r + \sum_{p \in P} \lambda_p \cdot p \,\middle|\, \forall p \in P : \lambda_p \in \mathbb{N} \right\}$$

A *semilinear set* is a finite union of linear sets.

Semilinear sets can be finitely represented as sets of pairs $\{(r_1, P_1), (r_2, P_2), \ldots, (r_n, P_n)\}$.

Deciding Reachability (Sketch)

Theorem 11.8 (Leroux 2012) If a marking m_1 is **not reachable** from m_0 (in a net N), there there is a semilinear set $\mathcal{M}$ of markings of N such that

1. $m_0 \in \mathcal{M}$;
2. if $m \in \mathcal{M}$ and $m \xrightarrow{t} m'$, then $m' \in \mathcal{M}$; and
3. $m_1 \notin \mathcal{M}$.

Deciding Reachability (Sketch)

Given a semi-linear set $\{(r_1, P_1), ..., (r_n, P_n)\}$, checking 1. and 3. amounts to solving the equations

$$m_0 = r_i + \sum_{p \in P_i} \lambda_p \cdot p \text{ and } m_1 = r_i + \sum_{p \in P_i} \lambda_p \cdot p$$

for unknowns λ_p for integer solutions.

Checking 2. reduces to checking the validity of a formula in Presburger arithmetic (easy, decidable).

Deciding Reachability (Sketch)

Deciding reachability consists of the two semi-decision procedures running in parallel:

1. Construction of the reachability graph in breadth-first-search, stopping and *accepting* when the goal marking has been constructed.
2. Enumeration of all semilinear sets, stopping and *rejecting* if a a set satisfies 1.–3.

Summary and Outlook

- decision procedures for bounded Petri nets (e.g., liveness)
- decision procedure for the boundedness problem
- decision procedures based on reachability
- sketch of decision of reachability problem

Next Lecture

- Petri net languages
  - Hack's separator languages $\mathscr{P}_0$ and $\mathscr{Q}_0$
- Complexity (sketch) of reachability
  - *Immerman-Szelepcsényi* (1988) on Steroids due to *Lipton* (1976)