Reachability Analysis for a Class of Petri Nets

Yu Ru\textsuperscript{1} and Christoforos N. Hadjicostis\textsuperscript{2,1}

Dept. of Electrical and Computer Engineering
\textsuperscript{1}Univ. of Illinois at Urbana-Champaign & \textsuperscript{2}Univ. of Cyprus

December 2009
Outline of the Talk

- Background on Reachability Analysis
- Strictly Monotone and Monotone Petri Nets
- Reachability Analysis and Application to Fault Diagnosis
- Conclusions
Reachability Problem

Problem: given a Petri net, check if final state $M$ is reachable from initial state $M_0$

Importance: the problem plays a central role in Petri net theory (e.g., for liveness analysis and deadlock checking)

Methods: the problem is decidable (Mayr, 1984) but its complexity is still unclear

- Reachability tree (Karp et al, 1969; Wang et al, 2004; Ru et al, 2006)
- Results for acyclic nets, marked graphs, etc. (Murata, 1989)
- Petri nets without transition invariants (Kostin, 2003)
- Net transformation based method (Ramachandran et al, 2004)
Our Contribution

- New classes of Petri nets: monotone and strictly monotone Petri nets

- Finite enumeration based reachability algorithm which can be easily adapted to simultaneous reachability checking of multiple final markings

- Computational complexity of the reachability algorithm characterized exactly
Petri Net Notation

- Petri net structure $N = (P, T, F, W)$
  
  $P = \{p_1, p_2, p_3\} \quad \& \quad T = \{t_1, t_2, t_3\}$

- Marking $M : P \mapsto N_0$: $M_0 = (1\ 1\ 0)^T$

- Petri net $G = \langle N, M_0 \rangle$

- $t$ is enabled at $M$ if $M(p) \geq W(p, t)$ for $p \in P$ (e.g., $t_1$ is enabled at $M_0$)

- Firing sequence $S \in T^*$ (e.g., $t_1t_2$)

State equation: $M = M_0 + D\sigma$ (e.g., $(0\ 1\ 1)^T = M_0 + D(1\ 1\ 0)^T$)

\[ D = (D_{ij}) = (W(t_j, p_i) - W(p_i, t_j)) = \begin{pmatrix} -1 & 0 & 1 \\ 1 & -1 & 0 \\ 0 & 1 & -1 \end{pmatrix} \]
Labeled Petri Net Notation

- Labeled Petri net \((G, \Sigma, L)\)
- Alphabet \(\Sigma\): set of labels (e.g., \(\{a, b\}\))
- Labeling function \(L : T \mapsto \Sigma \cup \{\lambda\}\) (e.g., \(L(t_1) = L(t_2) = a\), and \(L(t_3) = b\))
- Given \(S = t_{s_1} t_{s_2} \cdots t_{s_k}\), observation sequence \(L(S) := L(t_{s_1}) L(t_{s_2}) \cdots L(t_{s_k})\)

Consistent marking: Given \((G, \Sigma, L)\) and a sequence of observation \(\omega\),
\[ C(\omega) = \{M | \exists S \in T^*: M_0[S] M \text{ and } L(S) = \omega\} \]

For example, if \(\omega = a\), then \(C(\omega) = \{(0 2 0)^T, (1 0 1)^T\}\)

- Unobservable subnet: Obtained by removing all transitions (not labeled \(\lambda\)) and related arcs
Reachability Problem Formulation

Given a Petri net $G$ with initial marking $M_0$, and a final target marking $M$, is $M$ reachable?

Is $(4\ 0)^T$ reachable from $M_0 = (2\ 0)^T$?
Method: enumerate all reachable markings and check if $M_1$ is one of them

Is $(1\ 0)^T$ reachable from $M_0$?
Does the previous method still work?
No, but we do not have to enumerate all markings (at least for certain Petri nets)
Strictly Monotone Petri Nets

A Petri net $G$ with $n$ places and $m$ transitions is said to be strictly monotone if there exists an $n$-dimensional real vector $Y$ such that $Y^T D > 0_m$.

Remarks

- Example: acyclic Petri nets without source transitions
- There are no transition invariants in strictly monotone Petri nets

Argument: suppose there exists a nonzero integer vector $X$ with nonnegative entries such that $DX = 0_n$, then $Y^T DX = Y^T (DX) = 0$ while $Y^T DX = (Y^T D)X > 0$. A contradiction.

---

$1$ A Petri net is deadlock structurally bounded if $\exists y$ with strictly positive integer entries such that $y^T D < 0^T$. Acyclic Petri nets without source transitions are deadlock structurally bounded (Ru, et. al., 2009), hence also strictly monotone.
### Monotone Petri Nets

A Petri net $G$ is said to be monotone if there exists an $n$-dimensional nonzero real vector $Y$ such that $Y^T D \geq 0^T_m$.

**Remarks**

- Example: structurally bounded Petri nets
- (Strictly) Monotone Petri nets are not necessarily (structurally) bounded

![Petri net diagram]

The net on the left

- Strictly monotone verified using $Y = (0.5 \ 0.8)^T$
- Not (structurally) bounded

---

2 A Petri net is bounded if for any marking $M$ reachable from $M_0$, there exists an integer constant $K$ such that $M(p) \leq K$ for any $p \in P$. A Petri net is structurally bounded if it is bounded for any $M_0$. 

---

Y. Ru & C.N. Hadjicostis: Reachability Analysis
CDC 2009, Shanghai, China
Comparison with State Machines and Marked Graphs

State Machines: (a) and (b)
- (a) is strictly monotone while (b) is not
- All state machines are monotone using $Y = 1_n$

Marked Graphs: (b) and (c)
- No marked graph is strictly monotone
  Suppose $\exists Y$, $Y^T D > 0_m$. $1_n^T D^T = 0_m$ as $D^T$ is the incidence matrix of a state machine. Then, $1_n^T D^T Y = (1_n^T D^T) Y = 0$; on the other hand, $1_n^T D^T Y = 1_n^T (D^T Y) = 1_n^T (Y^T D)^T > 0$. A contradiction
- (b) is monotone while (c) is not
Given a strictly monotone Petri net \( \langle N, M_0 \rangle \), if \( M \) is reachable, then

\[
M = M_0 + D\sigma \\
Y^T(M - M_0) = Y^TD\sigma
\]

Let \( Z = Y^TD \), we have

\[
\left( \min_i Z(i) \right) \times \sum_{t \in T} \sigma(t) \leq Z\sigma \leq \left( \max_i Z(i) \right) \times \sum_{t \in T} \sigma(t)
\]

Then, \( l \leq \sum_{t \in T} \sigma(t) \leq k \) where

\[
l = \left[ \frac{Y^T(M - M_0)}{\max_i Z(i)} \right] \quad \text{(1)}
\]

and

\[
k = \left[ \frac{Y^T(M - M_0)}{\min_i Z(i)} \right] \quad \text{(2)}
\]
Reachability Algorithm

Basic idea: as $\sum_{t \in T} \sigma(t)$ is the length of the firing sequence $S$, if the marking $M$ is reachable, $l \leq |S| \leq k$

Algorithm

Reachability Algorithm for Strictly Monotone Petri Nets

Input: Petri net $G = \langle N, M_0 \rangle$ and a target marking $M$

Output: $M$ is reachable or unreachable

1) Check if $G$ is strictly monotone. If not, exit without output;
2) Obtain $Y$ such that $Y^T D > 0_m^T$. Then calculate $l$ using Eq. (1) and $k$ using Eq. (2). If $k \leq 0$, exit with $M$ unreachable;
3) Enumerate all markings reachable by a firing sequence of length less than or equal to $k$ and check if $M$ is one of them (checking only needed for firing sequences longer than or equal to $l$): if $M$ is one of these markings, then $M$ is reachable; otherwise, $M$ is unreachable.
Reachability Analysis for (Strictly) Monotone Petri Nets

### Example

- $M_0 = (2 \ 0)^T$
- $D = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix}$
- The net is strictly monotone (using $Y = (0.5 \ 0.8)^T$)

- Check if $M_1 = (1 \ 0)^T$ is reachable: as $Z = Y^T D = (0.2 \ 0.3)$,
  \[
  k = \left\lfloor \frac{Y^T(M_1 - M_0)}{\min_i Z(i)} \right\rfloor = \left\lfloor \frac{-0.5}{0.2} \right\rfloor = -2. \text{ Unreachable.}
  \]

- Check if $M_2 = (4 \ 0)$ is reachable: as $k = 5$ and $l = 3$, we run the algorithm and $M_2$ is reachable with a firing sequence of length 4 satisfying $l \leq 4 \leq k$. 
Strict Monotonicity Checking and Algorithmic Complexity

Strict Monotonicity Checking

- The existence of $Y$ satisfying $Y^T D > 0$ is characterized by the I-rank (Dines, 1919) and an exponentially convergent algorithm is given in (Ho et al., 1965).
- Linear programming formulation: $\min C^T Y \ s.t. \ Y^T D > 0^T_m$
- Nonlinear programming formulation: $\min \left[ \frac{Y^T(M-M_0)}{\min_i Z(i)} \right] \ s.t. \ Y^T D > 0^T_m$

Solution $Y$ will minimize the search depth $k$

E.g., check if $M_2$ is reachable in the previous example: we get $Y = (0.2 \ 0.3)^T$ and $k = l = 4$ using the nonlinear programming formulation. Result is the same as before but with less running time.

Algorithmic Complexity

- Steps 1 and 2 of the algorithm can be solved with polynomial complexity using linear programming solvers.
- Step 3 has the complexity $O(nmk^{2m-1})$ based on the bounds in (Ru et al., 2009).
Application to Fault Diagnosis

Fault Diagnosis Setting

- Given a labeled Petri net with $L(t) = t$ for observable $t$ and $\lambda$ for unobservable $t$, a finite set of faulty (undesirable) states $M_f$, and an observation sequence $\omega$, determine the set of possible faulty states $M_p := C(\omega) \cap M_f$
- If $M_p = \emptyset$, there are no faults

Methods

- Calculate $C(\omega)$ and check what faulty states are in $C(\omega)$
  $\implies$ Complexity: $O(|C(\omega)| \times |M_f|)$ if $|C(\omega)|$ is finite
- Can we improve the complexity?
Basis Marking Based Method

Basis markings (Giua et al., 2005)

- $M_{b,\omega}$: a marking reached from $M_0$ by firing $\omega$ and all unobservable transitions whose firings are strictly necessary to enable $\omega$
- $C(\omega) = \{ M : \exists M_{b,\omega}, S \in T_{uo}^*, M_{b,\omega}[S]M \}$, if the unobservable subnet is acyclic
- Advantage: the number of basis markings is much smaller than consistent markings (e.g., if the unobservable subnet is also backward conflict free, then there is only one basis marking)

Fault Diagnosis

- Diagnosis problem: determine faulty states that are reachable from any basis marking in the unobservable subnet
- Method: if the unobservable subnet is acyclic without source transitions, reachability can be checked via previous algorithm
- Efficiency: calculate the smallest $l$ and largest $k$ for all faulty markings and then check their reachability
Extension to Monotone Petri Nets

Monotone Petri Nets ($\exists Y$ such that $Y^T D \geq 0$)

- Choose $Y_{\text{max}}$ to maximize the number of strictly positive entries in $Y^T D$
- Let $Z_{\text{max}} = Y_{\text{max}}^T D$, then

$$Y_{\text{max}}^T (M - M_0) = Y_{\text{max}}^T D \sigma = \sum_{i: Z_{\text{max}}(i) > 0} Z_{\text{max}}(i) \sigma(t_i).$$

As before, $l \leq \sum_{i: Z_{\text{max}}(i) > 0} \sigma(t_i) \leq k$ where

$$l = \left\lfloor \frac{Y_{\text{max}}^T (M - M_0)}{\max_{i: Z_{\text{max}}(i) > 0} Z(i)} \right\rfloor \quad \text{(3)}$$

and

$$k = \left\lceil \frac{Y_{\text{max}}^T (M - M_0)}{\min_{i: Z_{\text{max}}(i) > 0} Z(i)} \right\rceil \quad \text{(4)}$$
Reachability Analysis for (Strictly) Monotone Petri Nets

Basic Idea

\[ l \leq \sum_{i: Z_{\text{max}}(i) > 0} \sigma(t_i) \leq k \]

Given monotone \( G \), calculating all reachable markings satisfying the above inequalities \( \Leftrightarrow \) Calculating markings that are consistent with \( \omega = e^i \) for \( i = l, l + 1, \ldots, k \) in labeled Petri net \( (G, \Sigma, L)_Z \), where \( \Sigma = \{e\} \) and \( L(t_i) = e \) if \( Z_{\text{max}}(i) > 0 \), \( L(t_i) = \lambda \) otherwise.

If the unobservable subnet of \( (G, \Sigma, L)_Z \) is structurally bounded, then \( C(\omega) \) is finite and previous reachability algorithm can be extended to monotone Petri nets.
Conclusions

Reachability Analysis for (Strictly) Monotone Petri nets

- New classes of Petri nets: strictly monotone and monotone Petri nets
- Reachability Algorithm based on finite enumeration
- Application of the algorithm to fault diagnosis
- Future work: reachability analysis for general Petri nets
References


