Branching processes of conservative nested Petri nets

Daniil Frumin, Irina Lomazova

International Laboratory of Process-Aware Information Systems
National Research University Higher School of Economics

July 17, 2014
Talk overview

1. Petri nets and Nested Petri nets

2. Petri net unfoldings

3. Branching processes of NP-nets

4. Conclusion
Petri nets

D. Frumin, I. Lomazova
Petri nets

D. Frumin, I. Lomazova
B-processes of conservative NP-nets
July 17, 2014 4 / 49
Petri nets
Petri nets (definition)

\[ N = (P, T, F, M_0) \]

- \( P \) and \( T \) are disjoint sets of places and transitions;
- \( F \subseteq (P \times T) \cup (T \times P) \) is a flow relation;
- \( M_0 \subseteq P \) is an initial marking of \( N \).

Pre- and post-set functions are defined for each \( x \in T \):

\[
\cdot x = \{ y \mid (y, x) \in F \}
\]

\[
x^\cdot = \{ y \mid (x, y) \in F \}
\]
A transition $t$ in the Petri net $N = (P, T, F, M_0)$ is active under a marking $M$ iff $\bullet t \subseteq M$.

An active transition may fire, leading to a marking $M' = M - \bullet t + t^\bullet$, denoted as $M \xrightarrow{t} M'$.

A marking $M$ is reachable (from the initial marking $M_0$) iff there exists a sequence of firings $M_0 \xrightarrow{t_1} M_1 \xrightarrow{t_2} M_2 \rightarrow \cdots \rightarrow M$ leading to it.
Nested Petri nets

- The “flat” structure of regular Petri nets can be inconvenient, when modelling systems with multiple autonomous agents.
- Nested Petri net (NP-net) is an extension of classical Petri nets used for modelling dynamic multi-agent systems.
- In NP-net tokens can be Petri nets themselves (“nets-within-nets” approach).
Nested Petri nets

\[ \text{NP-net} = \text{system net} + \text{element nets} \]

An instance of the element nets are called net tokens.
Nested Petri nets: formally

An NP-net $NP$ is a tuple $(SN, (EN_1, \ldots, EN_k), \nu, \lambda, W)$, where

- $SN = (P_{SN}, T_{SN}, F_{SN})$ is a Petri net called a system net.
- For each $i = 1, k$, $EN_i = (P_{EN_i}, T_{EN_i}, F_{EN_i})$ is a Petri net called an element net, where all the sets of places and transitions are disjoint; each element net is assigned a type from $Type$.
- $\nu : P_{SN} \rightarrow Type \cup \{\bullet\}$ is a type assignment function
- $\lambda : T_{NP} \rightarrow Lab$ is a partial labeling function, where $T_{NP} = T_{SN} \cup T_{EN_1} \cup \cdots \cup T_{EN_k}$;
- $W : F_{SN} \rightarrow Var \cup \{\bullet\}$ is an arc labeling function s.t. for an arc $r$ adjacent to a place $p$ the type of $W(r)$ coincides with the type of $p$.

A marked element net is called a net token.
A marking of an NP-net maps each place of the system net to a multiset of regular or net tokens. Marking should respect the typing.
NPN markings

A marking of an NP-net maps each place of the system net to a multiset of regular or net tokens. Marking should respect the typing.

\[ M : P_{SN} \rightarrow \mathcal{M}(A \cup \{\bullet\}) \]

where \( A = \{(EN_i, \mu_i) \mid \mu_i \text{ is a marking of } EN_i\} \)
NP-net behaviour: system-autonomous step

\[ p_1 \rightarrow t_1 \rightarrow p_2 \rightarrow \text{Lock}_1 \rightarrow \text{Res} \rightarrow \text{Release}_1 \rightarrow p_4 \]

\[ q_1 \rightarrow t_2 \rightarrow q_2 \rightarrow \text{Lock}_2 \rightarrow \text{Res} \rightarrow \text{Release}_2 \rightarrow q_4 \]

\[ \text{DoStuff} \rightarrow \text{Lock} \rightarrow \text{SomeWork} \rightarrow \text{Release} \rightarrow \text{Lock} \]

D. Frumin, I. Lomazova
NP-net behaviour: system-autonomous step

\[ \begin{align*}
\text{DoStuff} & \quad \text{Lock} \\
& \quad \text{a}_1 \quad \text{L} \\
& \quad \text{a}_2 \\
& \quad \text{a}_3 \\
\text{Release} & \quad \text{R} \\
& \quad \text{SomeWork} \\
\end{align*} \]
NP-net behaviour: system-autonomous step

D. Frumin, I. Lomazova
B-processes of conservative NP-nets
July 17, 2014 14 / 49
NP-net behaviour: element-autonomous step

\[ p_1 \xrightarrow{t_1} q_1 \]
\[ p_2 \xrightarrow{Lock_1} L \xrightarrow{Res} Lock_2 \]
\[ p_3 \xrightarrow{Release_1} R \]
\[ p_4 \]

\[ a_1 \xrightarrow{Lock} a_2 \]
\[ a_3 \xrightarrow{SomeWork} a_4 \]

DoStuff

Lock

Release
NP-net behaviour: element-autonomous step

\begin{itemize}
  \item \textbf{p}_1 \quad \bullet \quad \textbf{q}_1
  \item \textbf{p}_2 \quad t_1 \quad \text{Lock}_1 \quad \text{Res} \quad \text{Release}_1 \quad \textbf{p}_3
  \item \textbf{q}_2
  \item \textbf{p}_4 \quad t_2 \quad \text{Lock}_2 \quad \text{Release}_2 \quad \textbf{q}_3 \quad \textbf{q}_4
  \item \text{DoStuff} \quad \text{Lock} \quad \text{Release} \quad \text{SomeWork}
  \item \textbf{a}_1 \quad \textbf{a}_2 \quad \textbf{a}_3
\end{itemize}
NP-net behaviour: synchronization step
NP-net behaviour: synchronization step

\[ \begin{align*}
  p_1 & \rightarrow t_1 \\
p_2 & \rightarrow \text{Lock}_1 \\
p_3 & \rightarrow \text{Release}_1 \\
p_4 & \\
\end{align*} \]

\[ \begin{align*}
  & \quad \text{Lock}_1 \\
  & \quad \text{Lock}_2 \\
  & \quad \text{Res} \\
  & \quad \text{Release}_2 \\
\end{align*} \]

\[ \begin{align*}
  q_1 & \rightarrow t_2 \\
q_2 & \\
q_3 & \\
q_4 & \\
\end{align*} \]

\[ \begin{align*}
  & \quad \text{DoStuff} \\
  & \quad \text{Lock} \\
  & \quad \text{SomeWork} \\
  & \quad \text{Release} \\
\end{align*} \]

D. Frumin, I. Lomazova

B-processes of conservative NP-nets

July 17, 2014 18 / 49
Talk overview

1. Petri nets and Nested Petri nets
2. Petri net unfoldings
3. Branching processes of NP-nets
4. Conclusion
True concurrency and Petri nets

- Sequential execution: $t_1, t_3, t_6, t_5$ and $t_1, t_6, t_3, t_5$
- There may be several sequential executions corresponding to one set of transitions.
True concurrency

True concurrency vs interleavings

- Non-true concurrency semantics for process algebras:
  \[ a \parallel b \simeq a.b + b.a \]

- True concurrency semantics distinguish between parallel composition and non-deterministic choice between interleavings.
Non-sequential processes captures \textit{concurrent} runs of the net.
Unfoldings and computational trees

All the sequential executions of the net can be bundled in a computational tree.

All the non-sequential processes of the net can be bundled in an unfolding.
Unfoldings in verification

- Unfoldings of Petri nets provide true concurrency semantics to Petri nets.
- If we can “cut” the unfolding to a finite prefix, then we can use it for verification.
- The size of finite prefixes of unfoldings can be much smaller than the size of the reachability graph.
Unfoldings

Unfoldings are represented by a special class of Petri nets - Occurrence nets.

Occurrence nets are acyclic, the flow relation induces a partial order \(<\) transitive closure of \(F\).
Unfoldings are defined using branching processes – partial branching concurrent runs of the system. The function $h$ relates the nodes of a branching process to the nets of the main net.

Places and transitions in a branching process are called conditions and events.
Branching processes

A net, consisting just of places, corresponding to the initial marking of $N$, is a branching process.

The initial marking of a branching process is the $\prec$-minimal set.
If $X$ is a set of reachable conditions of a branching process $B$, and $X$ correspond to a set $h(X)$ of places in the net $N$, that enable a transition $t$.

Then we can obtain a branching process $B'$ by adding a new event $e$ (which corresponds to $t$), and “fresh” post-conditions which correspond to $t^\bullet$. Such $e$ is called a possible extension of $B$. 
Let $BB$ be a (finite or infinite) set of branching processes. The net $\bigcup BB$ is a branching process.
A branching process $B_1$ is said to be a *prefix* of a branching process $B_2$ (denoted as $B_1 \sqsubseteq B_2$), iff $B_1$ can be “included” in $B_2$.

The maximal (w.r.t. $\sqsubseteq$) branching process is called an *unfolding*, and is denoted by $U(N)$.
Fundamental property of unfoldings

Let $M$ be a reachable marking of $N$, and $M_U$ be a reachable marking of $U(N)$, such that $M_U$ correspond to $M$.

1. if there is a step $M_U \xrightarrow{t_U} M'_U$ of $U(N)$, then there is a step $M \xrightarrow{t} M'$ of $N$, such that $h(t_U) = t \land h(M'_U) = M'$;

2. if there is a step $M \xrightarrow{t} M'$ of $N$, then there is a step $M_U \xrightarrow{t_U} M'_U$ in $U(N)$, such that $h(t_U) = t \land h(M'_U) = M'$. 
NP-nets are good for modeling multi-agent systems
Multi-agent systems inherently posses a high grade of concurrency
Unfolding NP-nets can be beneficial compared to state-space exploration
Talk overview

1. Petri nets and Nested Petri nets
2. Petri net unfoldings
3. Branching processes of NP-nets
4. Conclusion
Conservative NP-nets

Here we deal with safe conservative NP-nets.

- A net $N$ is called safe iff $\forall M \in \mathcal{RM}(N), M(p) \leq 1$.
- A net $N$ is called conservative iff any transition firing does not change the number of net tokens in the system net (inner markings of the net tokens can be changed).

D. Frumin, I. Lomazova

B-processes of conservative NP-nets

July 17, 2014
Conservative NP-nets

Here we deal with safe conservative NP-nets.

- A net $N$ is called safe iff $\forall M \in \mathcal{RM}(N), M(p) \leq 1$.
- A net $N$ is called conservative iff any transition firing does not change the number of net tokens in the system net (inner markings of the net tokens can be changed).

1. For all $t \in T_{SN}$ and for all $p \in \dot{t}$, $\exists ! p' \in t^* . W(p, t) = W(t, p')$ or $W(p, t)$ bullet.
2. For all $t \in T_{SN}$ and for all $p \in t^*$, $\exists ! p' \in \dot{t} . W(t, p) = W(p', t)$ or $W(t, p)$ bullet.

Such place $p'$ is said to be adjacent to $p$ via $t$. 
NP-nets unfoldings

Unfoldings for NP-net are defined using branching processes, similarly to the case of classical Petri nets.

$NP_2$: system net

$NP_2$: element net
Each net token has an ID assigned to it. The set of identified net tokens is denoted as $NTok$.

- Each condition in the element-indexed branching process is paired with an id from $NTok$.
- This corresponds nicely to the intuitive meaning of “conditions” and “events” in occurrence nets.
Initial branching process

System net

Net token $N_1$

Net token $N_2$

Initial b-process
Possible extensions of branching processes

System net

Net token $N_1$

Net token $N_2$

Branching process
Possible extensions of branching processes

System net

Net token $N_1$

Net token $N_2$

Branching process
Union of branching processes
The \( \sqsubseteq \) relation can be generalized to element-indexed branching processes.

Unfolding of an NP-net \( NP \) is the maximal element-indexed branching process \( U(NP) \).
Properties of branching processes

**Property**

Every element-indexed branching process is an occurrence net.

**Property**

A flat P/T-net is a special case of an NP-net with the empty set of element nets and no vertical synchronization.

Let $N$ be a P/T-net. The set of branching processes of $N$ is isomorphic to the set of element-indexed branching processes of $N$, when $N$ is considered as an NP-net.

**Property**

The behaviour of the unfolding is isomorphic to the behaviour of the net.
Verification with branching processes

- The theory of *canonical prefixes* can be directly applied to the element-indexed branching processes.
- The existing algorithms can be applied with minor changes.
Finite prefix generation example

\[ C' \approx C'' \iff \text{Mark}(C') = \text{Mark}(C'') \text{ and } C' \prec C'' \iff |C'| < |C''| \]
Execution problem

- Can a transition $t$ be executed in the net?
- We just have to check if $BP_c$ has a transition labeled by $t$. 
Deadlock problem

- Is there a deadlock in the net?
- The net has a deadlock iff there exists a configuration in the prefix that does not contain cut-off events and the corresponding marking is a deadlock in the prefix.

D. Frumin, I. Lomazova
B-processes of conservative NP-nets
July 17, 2014 46 / 49
Talk overview

1. Petri nets and Nested Petri nets
2. Petri net unfoldings
3. Branching processes of NP-nets
4. Conclusion
Future work

- Extending the technique to (a bigger) NP-net classes.
- Trying a more algebraic and compositional approach.
Thank you for your attention!