Efficient Reachability of Petri Nets with Read Arcs

César Rodríguez

LSV, ENS Cachan, France

GT-Vérif, ENS Cachan, 17 June 2013

Joint work with Stefan Schwoon, Paolo Baldan, and Victor Khomenko.
Plan

1. Contextual Unfoldings
2. Computing Complete Unfolding Prefixes
3. Reachability with Contextual Unfoldings
4. Contextual Merged Processes
5. Summary
Contextual Petri Nets (c-nets)

- Contextual nets are Petri nets + read arcs
- Natural representation of notion *checking without consuming*

A c-net is a tuple $\langle P, T, F, C, m_0 \rangle$

- $\bullet x$ for preset, $\bullet^* x$ for postset
- $\mathcal{t} = \{ p \in P \mid (t, p) \in C \}$ for context

Example

$p = \{ t, t' \}$
$t = \{ p \}$
Homomorphism $h : \mathcal{U}_N \rightarrow N$

$h(\text{mark}(\mathcal{U}_N)) = \text{mark}(N)$
Homomorphism $h: \mathcal{U}_N \rightarrow N$

$h(\text{mark}(\mathcal{U}_N)) = \text{mark}(N)$
Homomorphism \( h : \mathcal{U}_N \rightarrow N \)

\[ h(\text{mark}(\mathcal{U}_N)) = \text{mark}(N) \]
Contextual Unfoldings (i) — Example

Homomorphism $h : \mathcal{U}_N \rightarrow N$

$h(\text{mark}(\mathcal{U}_N)) = \text{mark}(N)$
Homomorphism $h : \mathcal{U}_N \rightarrow N$

$h(\text{mark}(\mathcal{U}_N)) = \text{mark}(N)$
Contextual Unfoldings (i) — Example

**Homomorphism** $h : \mathcal{U}_N \rightarrow N$

$h(\text{mark}(\mathcal{U}_N)) = \text{mark}(N)$
Homomorphism $h : \mathcal{U}_N \rightarrow N$

$h(\text{mark}(\mathcal{U}_N)) = \text{mark}(N)$
Contextual Unfoldings (i) — Example

Homomorphism $h : \mathcal{U}_N \rightarrow N$

$h(\text{mark}(\mathcal{U}_N)) = \text{mark}(N)$
Contextual Unfoldings (i) — Example

Homomorphism \( h: \mathcal{U}_N \rightarrow N \)

\( h(\text{mark}(\mathcal{U}_N)) = \text{mark}(N) \)
**Contextual Unfoldings (ii) — Complete Prefixes**

- $\mathcal{U}_N$ is the result of applying the construction ‘as much as possible’
- If you stop: finite unfolding prefix $\mathcal{P}_N$

**Definition**

Prefix $\mathcal{P}_N$ is marking-complete if:
for all marking $m$ reachable in $N$, there is marking $\tilde{m}$ reachable in $\mathcal{P}_N$ with 

$$h(\tilde{m}) = m.$$ 

Given $N$, we want to:
- Compute a marking-complete $\mathcal{P}_N$
- Use $\mathcal{P}_N$ to decide deadlock-freeness or coverability of $N$
Contextual Unfoldings Exploit Concurrent Read-Access
Contextual Unfoldings Exploit Concurrent Read-Access

C. Rodríguez (LSV)
Computing Prefix Extensions

The problem

Given $P_N$ and $t$, decide if we can extend $P_N$ with $e$ where $h(e) = t$.

(NP-complete)
The problem

Given $\mathcal{P}_N$ and $t$, decide if we can extend $\mathcal{P}_N$ with $e$ where $h(e) = t$.

(NP-complete)

- Enumerate sets of conditions $S$ s.t. $h(S) = \bullet t \cup t$ (exponential)
- If $S$ is coverable, return YES; otherwise continue (linear)
Computing Prefix Extensions

The problem

Given $\mathcal{P}_N$ and $t$, decide if we can extend $\mathcal{P}_N$ with $e$ where $h(e) = t$.

(NP-complete)

- Enumerate sets of conditions $S$ s.t. $h(S) = \bullet t \cup t$ (exponential)
- If $S$ is coverable, return YES; otherwise continue (linear)

How this is done for Petri nets?
Computing Prefix Extensions

The problem

Given \( \mathcal{P}_N \) and \( t \), decide if we can extend \( \mathcal{P}_N \) with \( e \) where \( h(e) = t \). (NP-complete)

- Enumerate sets of conditions \( S \) s.t. \( h(S) = \bullet t \cup t \) (exponential)
- If \( S \) is coverable, return YES; otherwise continue (linear)

How this is done for Petri nets?

Definition

Conditions \( c, c' \) are concurrent, \( c \parallel c' \), iff some run marks them both.
Computing Prefix Extensions

The problem

Given $P_N$ and $t$, decide if we can extend $P_N$ with $e$ where $h(e) = t$.

(NP-complete)

- Enumerate sets of conditions $S$ s.t. $h(S) = \bullet t \cup t$ (exponential)
- If $S$ is coverable, return YES; otherwise continue (linear)

How this is done for Petri nets?

Definition

Conditions $c, c'$ are concurrent, $c \parallel c'$, iff some run marks them both.

Proposition

Conditions $c_1, \ldots, c_n$ are coverable iff $c_i \parallel c_j$ holds for all $i, j \in \{1, \ldots, n\}$
However, for Contextual Unfoldings...

...the same approach doesn’t work:

- $c_4 \parallel c_5$ and $c_4 \parallel c_6$ and $c_5 \parallel c_6$ but $\{c_4, c_5, c_6\}$ is not coverable
- Cycle $e_1 \uparrow e_2 \uparrow e_3 \uparrow e_1$ of asymmetric conflict
Definition

A **history** of event \( e \) is any configuration \( H \) s.t.:
1. \( e \in H \)
2. Any run of the events of \( H \) fires \( e \) last

- **Enriched prefix**: label every condition \( c \) with the histories of \( \cdot c \) and \( \overline{c} \).
- **Enriched conditions**: pairs \( \langle c, H \rangle \)
Definition

Two enriched conditions $\rho = \langle c, H \rangle$ and $\rho' = \langle c', H' \rangle$ are concurrent, written $\rho \parallel \rho'$, iff:

$$\neg (H \neq H') \quad \text{and} \quad c, c' \in (H \cup H')^*$$
A Concurrency Relation for c-nets

**Definition**

Two enriched conditions $\rho = \langle c, H \rangle$ and $\rho' = \langle c', H' \rangle$ are concurrent, written $\rho \parallel \rho'$, iff:

$$\neg (H \neq H') \quad \text{and} \quad c, c' \in (H \cup H')^\bullet$$

**Proposition**

Conditions $c_1, \ldots, c_n$ coverable iff there is histories $H_1, \ldots, H_n$ verifying

$$\langle c_i, H_i \rangle \parallel \langle c_j, H_j \rangle \text{ for all } i, j \in \{1, \ldots, n\}.$$
A Concurrency Relation for c-nets

**Definition**

Two enriched conditions \( \rho = \langle c, H \rangle \) and \( \rho' = \langle c', H' \rangle \) are concurrent, written \( \rho \parallel \rho' \), iff:

\[
\neg (H \not\equiv H') \quad \text{and} \quad c, c' \in (H \cup H')^\bullet
\]

**Proposition**

Conditions \( c_1, \ldots, c_n \) coverable iff there is histories \( H_1, \ldots, H_n \) verifying

\[
\langle c_i, H_i \rangle \parallel \langle c_j, H_j \rangle \quad \text{for all} \quad i, j \in \{1, \ldots, n\}.
\]

**Proposition**

Let \( \rho = \langle c, H \rangle \) and \( e \) be the last enriched condition and event appended to the prefix, let \( \rho' = \langle c', H' \rangle \) be an arbitrary enriched condition. Then,

\[
\rho \parallel \rho' \iff (c' \in e^\bullet \land H = H') \lor \left( c' \notin e^\bullet \land c' \in H' \subseteq H \right)
\]
## Experiments with CUNF

<table>
<thead>
<tr>
<th>Net</th>
<th>Contextual Events</th>
<th>Contextual $t_C$</th>
<th>Ordinary Events</th>
<th>Ordinary $t_P$</th>
<th>$t_C/t_P$</th>
<th>$t_C/t_R$</th>
</tr>
</thead>
<tbody>
<tr>
<td>bds_1.sync</td>
<td>1866</td>
<td>0.14</td>
<td>12900</td>
<td>0.51</td>
<td>0.27</td>
<td>0.54</td>
</tr>
<tr>
<td>byzagr4_1b</td>
<td>8044</td>
<td>2.90</td>
<td>14724</td>
<td>3.40</td>
<td>0.85</td>
<td>0.55</td>
</tr>
<tr>
<td>ftp_1.sync</td>
<td>50928</td>
<td>34.21</td>
<td>83889</td>
<td>76.74</td>
<td>0.45</td>
<td>0.30</td>
</tr>
<tr>
<td>furnace_4</td>
<td>95335</td>
<td>18.34</td>
<td>146606</td>
<td>40.39</td>
<td>0.45</td>
<td>0.42</td>
</tr>
<tr>
<td>key_4.fsa</td>
<td>4754</td>
<td>6.33</td>
<td>67954</td>
<td>2.21</td>
<td>2.86</td>
<td>1.47</td>
</tr>
<tr>
<td>rw_1w3r</td>
<td>14490</td>
<td>0.45</td>
<td>15401</td>
<td>0.38</td>
<td>1.18</td>
<td>0.65</td>
</tr>
<tr>
<td>q_1.sync</td>
<td>10722</td>
<td>1.13</td>
<td>10722</td>
<td>1.21</td>
<td>0.93</td>
<td>0.52</td>
</tr>
<tr>
<td>dpd_7.sync</td>
<td>10457</td>
<td>0.91</td>
<td>10457</td>
<td>0.88</td>
<td>1.03</td>
<td>0.92</td>
</tr>
<tr>
<td>elevator_4</td>
<td>16856</td>
<td>1.26</td>
<td>16856</td>
<td>2.01</td>
<td>0.63</td>
<td>$&gt;0.01$</td>
</tr>
<tr>
<td>rw_12.sync</td>
<td>98361</td>
<td>3.10</td>
<td>98361</td>
<td>3.95</td>
<td>0.78</td>
<td>0.41</td>
</tr>
<tr>
<td>rw_2w1r</td>
<td>9241</td>
<td>0.40</td>
<td>9241</td>
<td>0.30</td>
<td>1.33</td>
<td>0.04</td>
</tr>
</tbody>
</table>

- C-net unfolding smaller or equal ordinary unfoldings
- In general faster than plain encoding
- Consistently faster than place-replication ($t_R$)
Recall

Marking \( m \) reachable in \( N \) iff there is a configuration \( C \) s.t. \( \text{mark}(C) = m \).

\[ \phi_{\text{reach}, M} := \phi_{\text{conf}} \land \phi_{\text{mark}, M} \]

Satisfying assignments of \( \phi_{\text{reach}, M} \) encode configurations reaching \( M \).

Example

Is \( p_3 \) reachable in \( N \)?

\[ c_3 \lor c'_3 (\text{\( p_3 \) marked}) \]

\[ c_3 \rightarrow e_1 \land \neg e_3 \] (token conservation for \( c_3 \))
Encoding Coverability into SAT

Recall

Marking $m$ reachable in $N$ iff there is a configuration $C$ s.t. $\text{mark}(C) = m$.

Example

Is $p_3$ reachable in $N$?

$c_3 \lor c'_3$  
$c_3 \rightarrow e_1 \land \neg e_3$  

(p3 marked)  

(token conservation for c3)
Recall

Marking $m$ reachable in $N$ iff there is a configuration $C$ s.t. $\text{mark}(C) = m$.

$\phi_{\text{reach}, M} := \phi_{\text{conf}} \land \phi_{\text{mark}, M}$

- Satisfying assignments of $\phi_{\text{reach}, M}$ encode configurations reaching $M$.
- Boolean variables: events + conditions of $P$

Example

Is $p_3$ reachable in $N$?

$c_3 \lor c'_3$  
$p_3$ marked

$c_3 \rightarrow e_1 \land \neg e_3$  
(token conservation for $c_3$)

...
### Experiments with CNA

<table>
<thead>
<tr>
<th>Net</th>
<th>Res.</th>
<th>Ordinary unfolding</th>
<th>c-net unfolding</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Unfolding</td>
<td>CLP</td>
</tr>
<tr>
<td>bds_1.sync</td>
<td>L</td>
<td>0.58</td>
<td>0.04</td>
</tr>
<tr>
<td>byzagr4_1b</td>
<td>L</td>
<td>3.71</td>
<td>0.53</td>
</tr>
<tr>
<td>dme11</td>
<td>L</td>
<td>6.56</td>
<td>0.60</td>
</tr>
<tr>
<td>dpd_7.sync</td>
<td>L</td>
<td>1.21</td>
<td>0.10</td>
</tr>
<tr>
<td>ftp_1.sync</td>
<td>L</td>
<td>45.37</td>
<td>1.13</td>
</tr>
<tr>
<td>furnace_4</td>
<td>L</td>
<td>37.44</td>
<td>1.29</td>
</tr>
<tr>
<td>rw_12.sync</td>
<td>L</td>
<td>3.95</td>
<td>0.08</td>
</tr>
<tr>
<td>rw_1w3r</td>
<td>L</td>
<td>0.30</td>
<td>0.11</td>
</tr>
<tr>
<td>rw_2w1r</td>
<td>L</td>
<td>0.22</td>
<td>0.04</td>
</tr>
<tr>
<td>elevator_4</td>
<td>D</td>
<td>2.58</td>
<td>0.24</td>
</tr>
<tr>
<td>key_4</td>
<td>D</td>
<td>1.68</td>
<td>0.07</td>
</tr>
<tr>
<td>mmgt_4.fsa</td>
<td>D</td>
<td>1.16</td>
<td>0.02</td>
</tr>
<tr>
<td>q_1.sync</td>
<td>D</td>
<td>1.76</td>
<td>&lt;0.01</td>
</tr>
<tr>
<td><strong>∑</strong></td>
<td></td>
<td>106.52</td>
<td>4.25</td>
</tr>
</tbody>
</table>
Conflicts Blow up (c-net) Unfoldings

\[ p_1 \]
\[ u_1 \rightarrow t_1 \]
\[ q_2 \]
\[ u_2 \rightarrow t_2 \]
\[ q_3 \]
\[ \vdots \]
\[ p_n \]
\[ u_n \rightarrow t_n \]
\[ q_{n+1} \]
\[ \vdots \]
\[ p_{n+1} \]

\[ p_1 \]
\[ u_1 \rightarrow t_1 \]
\[ q_2 \]
\[ p_2 \]
\[ u_2 \rightarrow t_2 \]
\[ q_3 \]
\[ p_3 \]
\[ \vdots \]
\[ p_{n+1} \]
Combining Two Methods

We integrate two partial-order representations:

- **Contextual unfoldings**: solves concurrent read access
- **Merged Processes**: solves **conflicts** + non-safeness

Resulting method: **Contextual Merged Processes** (CMPs)
- Often orders of magnitudes more compact
Definition

The occurrence depth of any node $x \in \mathcal{U}_N$ is the maximum number of $h(x)$-labelled nodes in any path from $\tilde{m}_0$ to $x$ in the digraph $(\tilde{m}_0 \cup [x] \cup [x]^\bullet, <_i)$. 
Contextual Merged Process (CMPs)

Definition

The **Contextual Merged Process** (CMP) of the unfolding prefix $\mathcal{P}_N$ is the labelled c-net $\mathcal{M}_N$ resulting from

1. Merging all conditions with same occurrence depth and label
2. Merging all events with same label, preset, postset and context.
Reachability of $n$-bounded c-net is

- $\text{PSPACE}$-complete on $N$
- $\text{NP}$-complete on marking-complete $P_N$
- $\text{NP}$-complete on marking-complete $M_N$

We present reduction into SAT for 1-safe c-nets
Proposition

Let $N$ be 1-safe and $M_N$ marking-complete:
Marking $m$ is reachable in $N$ iff there is a set $X$ of mp-events satisfying:

1. $\forall \hat{e} \in X : \forall \hat{c} \in \cdot \hat{e} \cup \hat{e} : (\hat{c} \in \hat{m}_0 \lor \hat{c} \in X \cdot)$, and
2. $\uparrow_X$ is acyclic, and
3. $m = \text{mark}(X)$.

Remarks

- (1) and (3) can be encoded into SAT in size linear in $M_N$
- (2) can be encoded in quadratic size, subject to more work
## Experiments with CMPs

| Benchmark | Name | | Unfolding | | Merged Process | |
|-----------|------|------|-----------|-----------|-----------|
|           |      | | Plain | Contextual | Plain | Contextual | |
| BDS       | 59   | | 21.73 | 5.73      | 1.14   | 44        |
| BRUJIN    | 165  | | 3.22  | 1.64      | 1.44   | 127       |
| BYZ       | 409  | | 46.11 | 25.57     | 1.03   | 303       |
| FTP       | 529  | | 85.74 | 82.51     | 1.05   | 455       |
| KNUTH     | 137  | | 2.88  | 1.59      | 1.31   | 112       |
| DME(8)    | 392  | | 10.64 | 10.64     | 1.04   | 360       |
| DME(10)   | 490  | | 15.53 | 15.53     | 1.04   | 450       |
| ELEV(3)   | 783  | | 6.48  | 6.48      | 1.00   | 346       |
| ELEV(4)   | 1939 | | 11.38 | 11.38     | 1.00   | 841       |
| KEY(2)    | 92   | | 3.92  | 1.82      | 2.50   | 105       |
| KEY(3)    | 133  | | 19.93 | 4.33      | 4.13   | 186       |
| KEY(4)    | 174  | | 113.82| 12.54     | 5.26   | 290       |
| MGMT(3)   | 172  | | 4.01  | 4.01      | 1.00   | 355       |
| MGMT(4)   | 232  | | 11.68 | 11.68     | 1.00   | 638       |
Summary and Future Work

- Contextual unfolding is feasible and efficient
  - Faster than ordinary unfolding
  - In our benchmark, deadlock-checking outperforms existing methods
  - Tool support: $\text{CUNF} + \text{CNA}$

- Contextual Merged Processes
  - New condensed representation
  - Fights three sources of state-explosion
  - Orders of magnitude smaller
Summary and Future Work

- Contextual unfolding is feasible and efficient
  - Faster than ordinary unfolding
  - In our benchmark, deadlock-checking outperforms existing methods
  - Tool support: CUNF + CNA
- Contextual Merged Processes
  - New condensed representation
  - Fights three sources of state-explosion
  - Orders of magnitude smaller

Future work

- Unfolding construction without concurrency relation
- Implementing model checking and direct construction of CMPs
Summary and Future Work

- Contextual unfolding is feasible and efficient
  - Faster than ordinary unfolding
  - In our benchmark, deadlock-checking outperforms existing methods
  - Tool support: $\text{Cunf} + \text{Cna}$

- Contextual Merged Processes
  - New condensed representation
  - Fights three sources of state-explosion
  - Orders of magnitude smaller

Future work

- Unfolding construction without concurrency relation
- Implementing model checking and direct construction of CMPs

Thank you for your attention