Construction and Verification of Unfoldings for Petri Nets with Read Arcs

César Rodríguez
joint work with Stefan Schwoon, Paolo Baldan

Laboratorire Spécification et Vérification (LSV)
ENS Cachan & CNRS, France

MOVEP, Marseille, 6 December 2012

Introduction

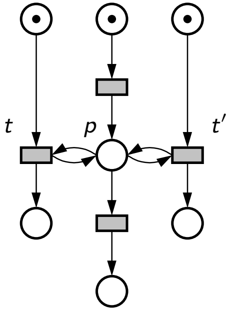
The problem

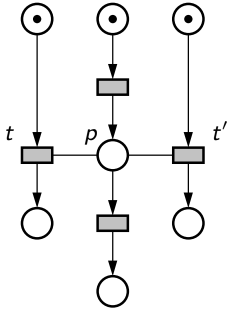
Verification of concurrent systems by means of the unfolding technique, when the system is modelled as a Petri net with read arcs.

- Unfolding up to exponentially more compact
- Unfolding algorithm more involved, but has better efficiency
- Reachability and deadlock-checking

Contextual Petri nets

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





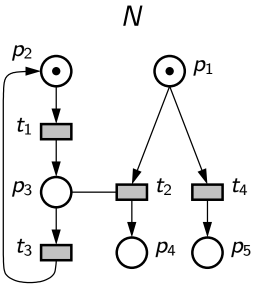
Notation

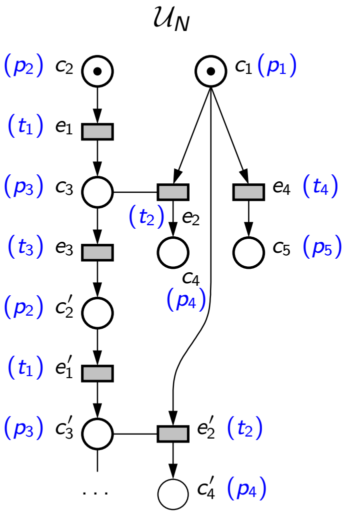
- A c-net is a tuple $\langle P, T, F, C, m_0 \rangle$
- ${}^\bullet x$ for preset, $x^\bullet$ for postset
- $\underline{t} = \{p \in P \mid (t, p) \in C\}$ for context

Example

$\underline{p} = \{t, t'\}$
$\underline{t} = \{p\}$

Contextual net unfoldings

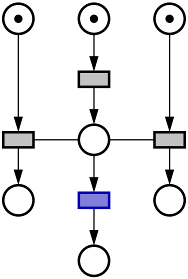




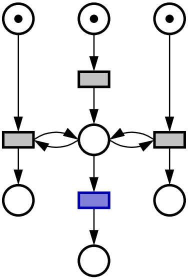
Remarks

- Labeling $f : \mathcal{U}_N \to N$
- $\mathcal{U}_N$ is marking-complete

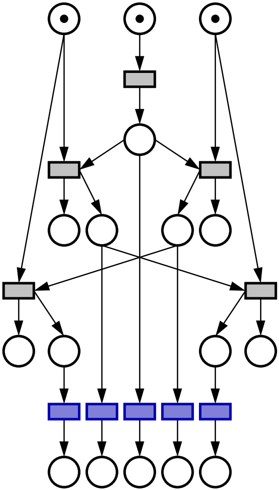
Contextual unfoldings exploit concurrent read access



A

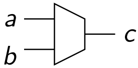


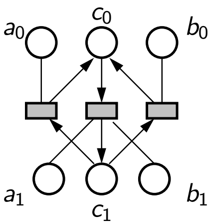
B

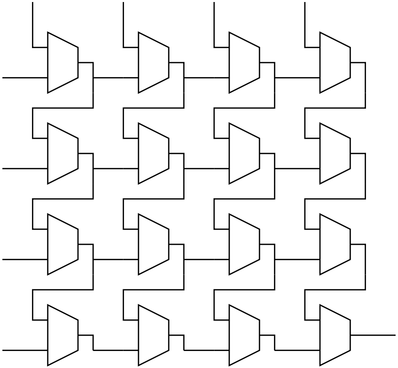


C

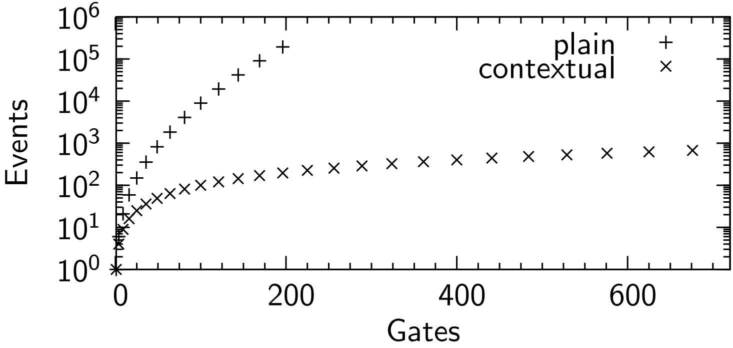
Asynchronous circuits







Asynchronous circuits



Computing prefix extensions

The problem

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

(NP-complete)

Computing prefix extensions

The problem

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

(NP-complete)

- Enumerate sets of conditions S s.t. $f(S) = {}^{\bullet}t \cup \underline{t}$ (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 $f(e) = t$

(NP-complete)

- Enumerate sets of conditions S s.t. $f(S) = {}^\bullet t \cup \underline{t}$ (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 $\mathcal{P}_N$ and t, decide if we can extend $\mathcal{P}_N$ with e where $f(e) = t$ (NP-complete)

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

How this is done for Petri nets?

Definition

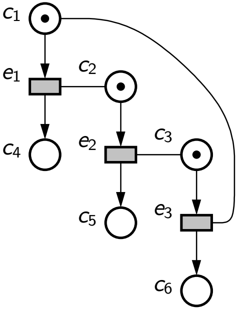
Conditions c, c' are concurrent, $c \parallel c'$, iff some run marks them both

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 unfolding...

...the same approach doesn't work:



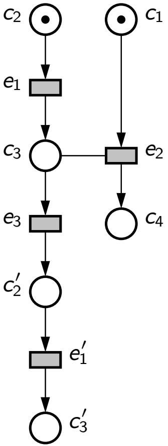
We have $c_4 \parallel c_5$, and $c_4 \parallel c_6$ and $c_5 \parallel c_6$ but $\{c_4, c_5, c_6\}$ is not coverable.

Histories for events and conditions

Definition

A history of e is a set of events H such that:

1. $e \in H$,
2. Events in H can be arranged to form a *run*,
3. *Any* run of the events of H fires e last.

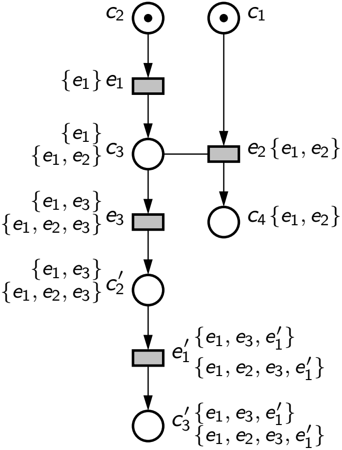


Histories for events and conditions

Definition

A history of e is a set of events H such that:

1. $e \in H$,
2. Events in H can be arranged to form a *run*,
3. *Any* run of the events of H fires e last.



Remarks

- Enriched prefix: events and conditions annotated with histories
- A pair (c, H) is called enriched condition
- This is the working *data structure*

A concurrency relation for contextual nets

Definition

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

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

A concurrency relation for contextual nets

Definition

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

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

Proposition

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

$$(c_i, H_i) \parallel (c_j, H_j) \text{ for all } i, j \in \{1, \ldots, n\}$$

A concurrency relation for contextual nets

Definition

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

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

Proposition

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

$$(c_i, H_i) \parallel (c_j, H_j) \text{ for all } i, j \in \{1, \ldots, n\}$$

Proposition

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

$$\rho \parallel \rho' \iff (c' \in e^{\bullet} \wedge H = H') \vee \left(c' \notin {}^{\bullet}e \;\wedge\; \bigwedge_{i=1}^{n} (\rho_i \parallel \rho') \;\wedge\; \underline{{}^{\bullet}e} \cap H' \subseteq H \right)$$

Experiments with Cunf

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

- Contextual unfolding smaller or equal than ordinary unfolding
- And in general faster than unfolding the plain encoding

Encoding deadlock and reachability into SAT

From a marking-complete unfolding prefix $\mathcal{P}$, we construct

- $\phi_{\mathcal{P}}^{\text{dead}}$, satisfiable iff N contains a deadlock
- $\phi_{\mathcal{P}}^{\text{reach, M}}$, satisfiable iff places M are coverable in N

Encoding deadlock and reachability into SAT

From a marking-complete unfolding prefix $\mathcal{P}$, we construct

- $\phi_{\mathcal{P}}^{\text{dead}}$, satisfiable iff N contains a deadlock
- $\phi_{\mathcal{P}}^{\text{reach, M}}$, satisfiable iff places M are coverable in N

Both formulas characterize configurations and reachable markings:

$$\begin{aligned} \phi_{\mathcal{P}}^{\text{dead}} &:= \phi_{\mathcal{P}}^{\text{conf}} \wedge \phi_{\mathcal{P}}^{\text{disable}} \\ \phi_{\mathcal{P}}^{\text{reach, M}} &:= \phi_{\mathcal{P}}^{\text{conf}} \wedge \phi_{\mathcal{P}}^{\text{mark, M}} \end{aligned}$$

where $\phi_{\mathcal{P}}^{\text{conf}}$ is defined as

$$\phi_{\mathcal{P}}^{\text{causal}} \wedge \phi_{\mathcal{P}}^{\text{sym}} \wedge \phi_{\mathcal{P}}^{\text{asym}}$$

- Implementation runs twice faster than the best tool we found

Summary

- Contextual unfoldings are up to exponentially more compact
- In our benchmark, verification based on contextual unfoldings performs better than existing methods
- Unfolder and unfolding-based analysis tool available at:

  `www.lsv.ens-cachan.fr/~rodriguez/tools/cunf/`

Current and future work

- Contextual merged processes
- Application in diagnosis
- We are searching for concurrent systems to evaluate our algorithms !!

Summary

- Contextual unfoldings are up to exponentially more compact
- In our benchmark, verification based on contextual unfoldings performs better than existing methods
- Unfolder and unfolding-based analysis tool available at:

  `www.lsv.ens-cachan.fr/~rodriguez/tools/cunf/`

Current and future work

- Contextual merged processes
- Application in diagnosis
- We are searching for concurrent systems to evaluate our algorithms !!

Thank you for your attention

References

Paolo Baldan, Andrea Corradini, Barbara König, and Stefan Schwoon.
McMillan's complete prefix for contextual nets.
ToPNoC, 1:199–220, 2008.

César Rodríguez.
CUNF.
http://www.lsv.ens-cachan.fr/~rodriguez/tools/cunf/.

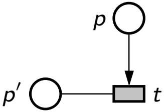
César Rodríguez and Stefan Schwoon.
Verification of Petri Nets with Read Arcs.
In *Proc. of CONCUR'12*, volume 7454 of *LNCS*, September 2012.

César Rodríguez, Stefan Schwoon, and Paolo Baldan.
Efficient contextual unfolding.
In *Proc. of CONCUR'11*, volume 6901 of *LNCS*, pages 342–357, September 2011.

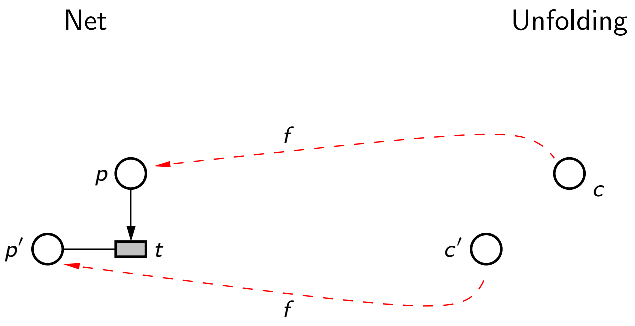
Computing possible extensions

Net

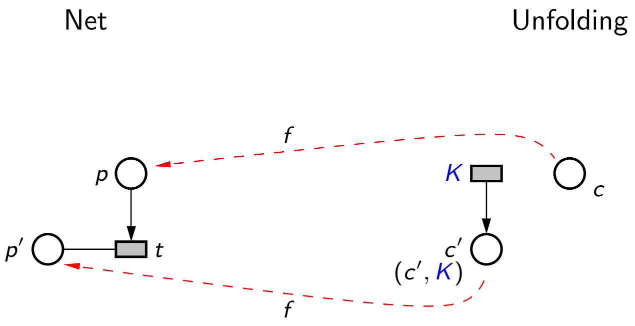
Unfolding



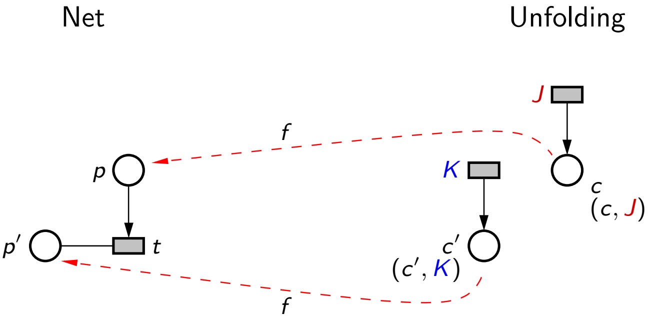
Computing possible extensions



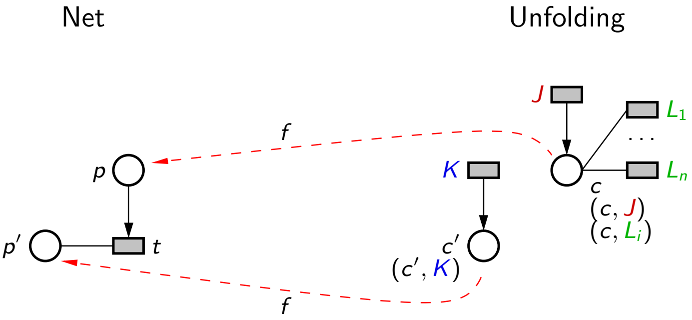
Computing possible extensions



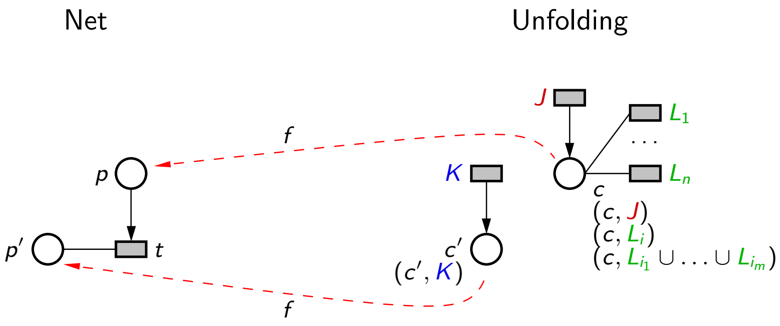
Computing possible extensions



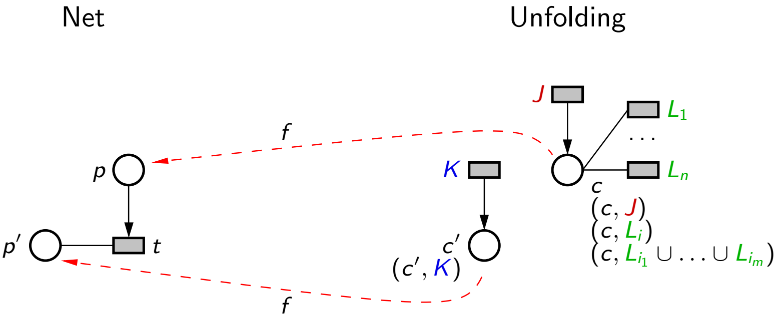
Computing possible extensions



Computing possible extensions

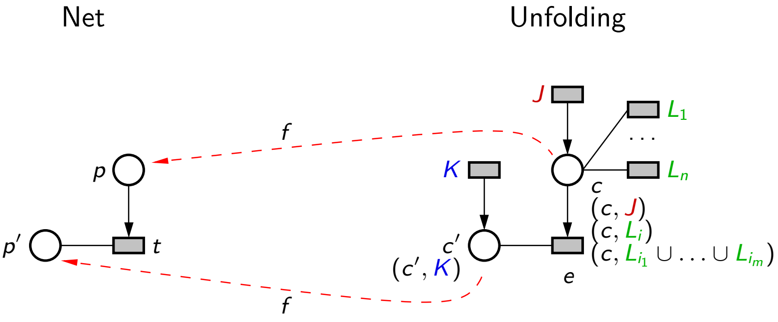


Computing possible extensions



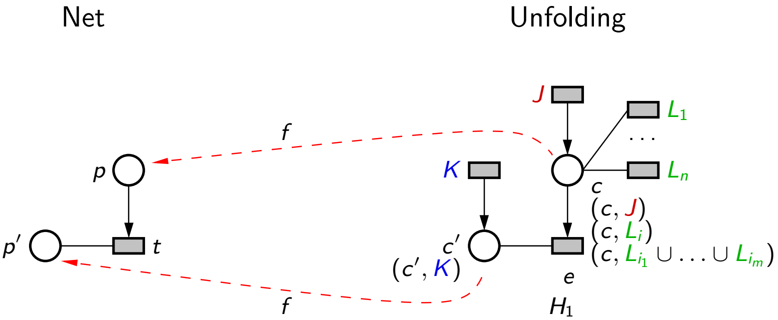
$(c, J) \parallel (c', K)$

Computing possible extensions



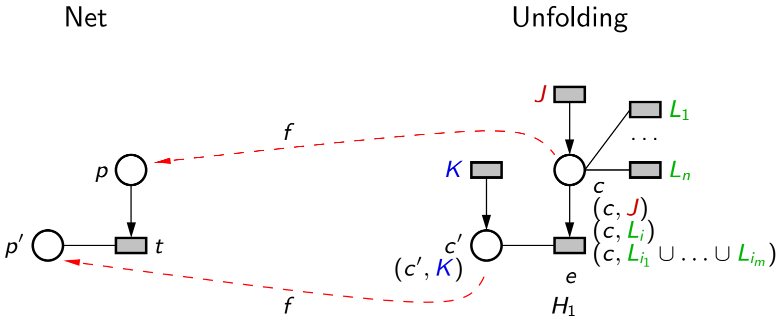
$(c, J) \parallel (c', K)$

Computing possible extensions



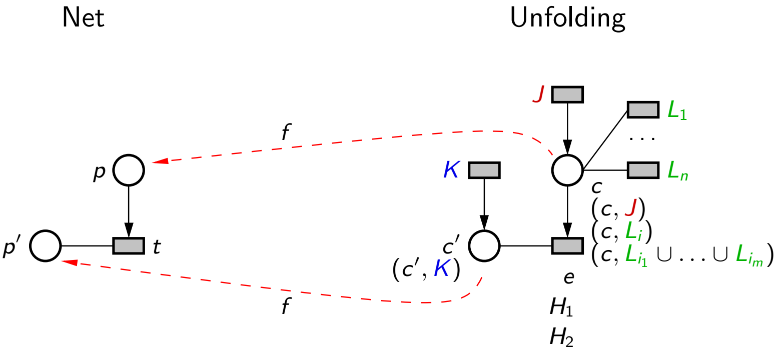
$$(c, J) \parallel (c', K) \qquad \rightsquigarrow H_1 = J \cup K$$

Computing possible extensions



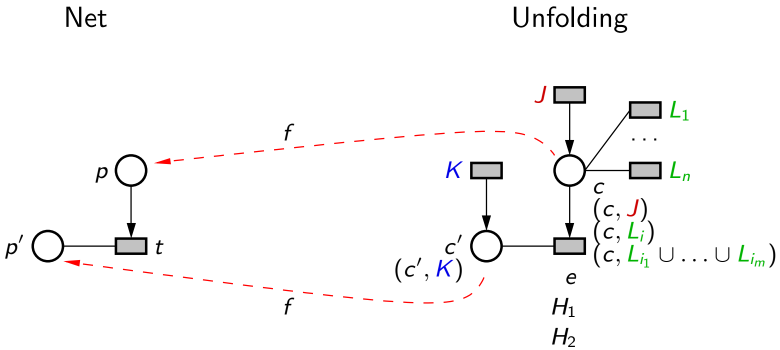
$$(c, J) \parallel (c', K) \qquad \rightsquigarrow H_1 = J \cup K$$
$$(c, L_i) \parallel (c', K)$$

Computing possible extensions



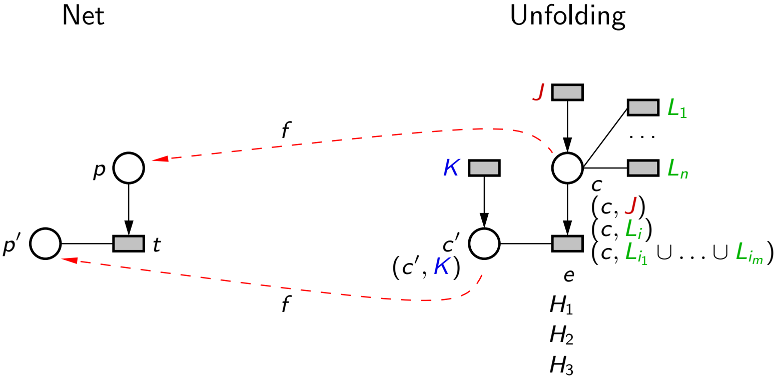
$$(c, J) \parallel (c', K) \quad \rightsquigarrow H_1 = J \cup K$$
$$(c, L_i) \parallel (c', K) \quad \rightsquigarrow H_2 = L_i \cup K$$

Computing possible extensions



$$(c, J) \parallel (c', K) \quad \rightsquigarrow H_1 = J \cup K$$
$$(c, L_i) \parallel (c', K) \quad \rightsquigarrow H_2 = L_i \cup K$$
$$(c, L_{i_1} \cup \ldots \cup L_{i_m}) \parallel (c', K)$$

Computing possible extensions



$$
\begin{aligned}
(c, J) \parallel (c', K) &\quad \rightsquigarrow H_1 = J \cup K \\
(c, L_i) \parallel (c', K) &\quad \rightsquigarrow H_2 = L_i \cup K \\
(c, L_{i_1} \cup \ldots \cup L_{i_m}) \parallel (c', K) &\quad \rightsquigarrow H_3 = L_{i_1} \cup \ldots \cup L_{i_m} \cup K
\end{aligned}
$$

Contextual unfolding — inductive definition

For a 1-safe contextual net $N = \langle P, T, F, C, m_0 \rangle$, the full unfolding $\mathcal{U}_N = \langle P', T', F', C', m'_0 \rangle$ is the 1-safe acyclic contextual net defined by the next inductive rules:

Mapping $f : \mathcal{U}_N \to N$ labels every event $\langle A, B, t \rangle$ with t and every condition $\langle e, p \rangle$ with p.