Automated Polyhedral Abstraction Proving

Nicolas Amat, **Silvano Dal Zilio**, **Didier Le Botlan**

LAAS-CNRS

Petri Nets, March 29 2023

What's Polyhedral Abstraction?

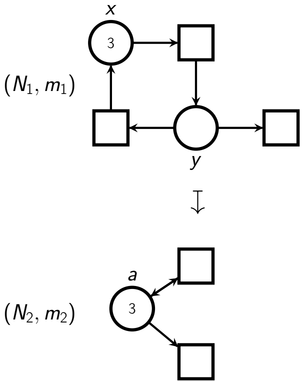
Introduction

$$(N_1, m_1) \equiv_E (N_2, m_2)$$

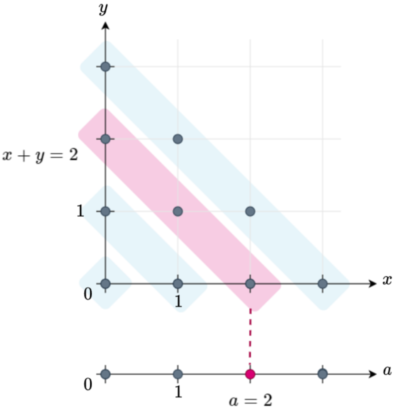
- General notion
- Equivalence between reachable markings (modulo solutions of E)

What's Polyhedral Abstraction?

Introduction

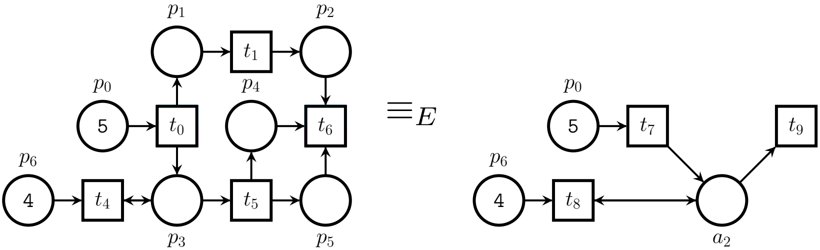


Net reduction example, with equation $E : a = x + y$



Relation between state-spaces

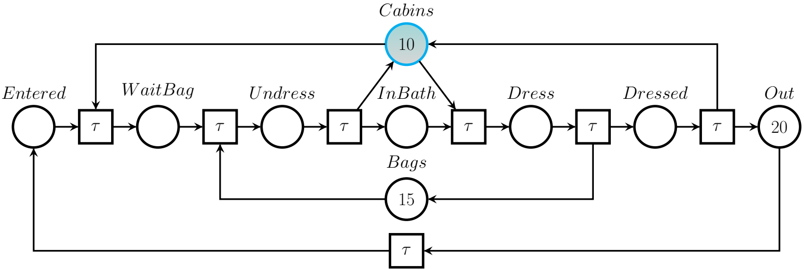
What's Polyhedral Abstraction?
Introduction



$$E \triangleq \begin{cases} p_5 = p_4 \\ a_1 = p_1 + p_2 \\ a_2 = p_3 + p_4 \\ a_1 = a_2 \end{cases}$$

SwimmingPool

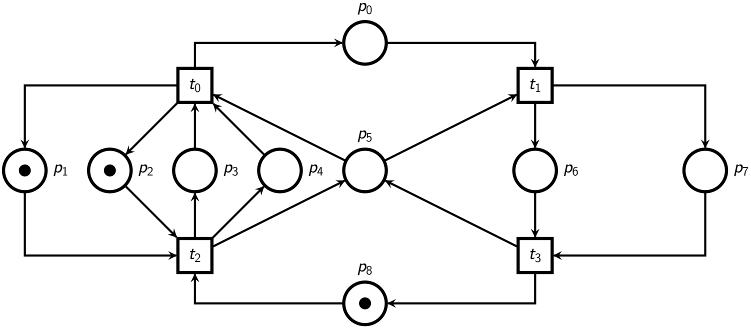
Introduction



$$E \triangleq \begin{cases} Cabins + Dress + Dressed + Undress + WaitBag = 10 \\ Dress + Dressed + Entered + InBath + Out + Undress + WaitBag = 20 \\ Bags + Dress + InBath + Undress = 15 \end{cases}$$

Petri Nets' Flag (Incorrect Abstraction)

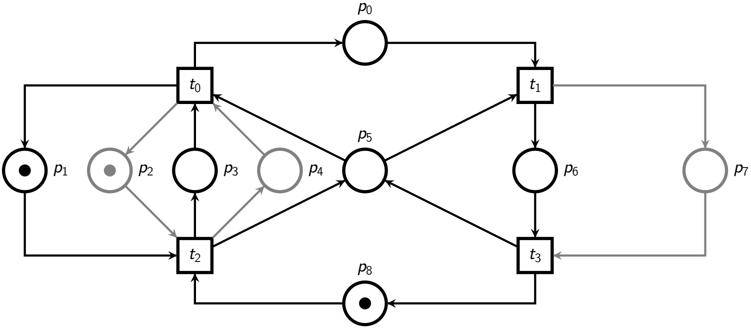
Introduction



$$E \triangleq \begin{cases} p_2 = p_1 \\ p_4 = p_3 \\ p_7 = p_6 \\ p_0 + p_5 + 2.p_6 + p_8 = 2 \\ p_0 + p_3 + p_6 + p_8 = 2 \\ p_1 + p_5 + p_6 = 1 \\ p_1 + p_3 = 1 \end{cases}$$

Petri Nets' Flag (Incorrect Abstraction)

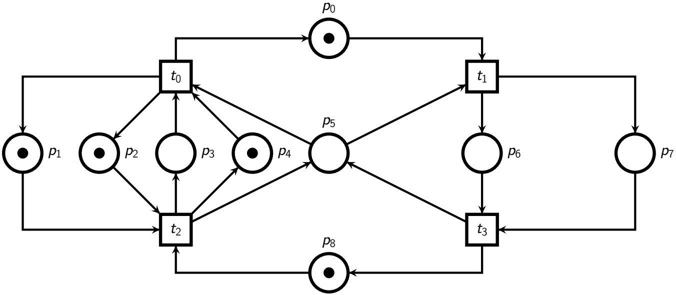
Introduction

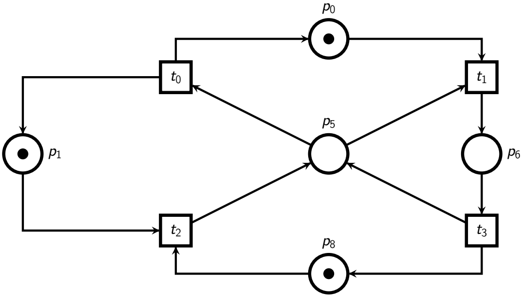


$$E \triangleq \begin{cases} p_2 = p_1 \\ p_4 = p_3 \\ p_7 = p_6 \\ p_0(2) + p_5 + 2.p_6 + p_8 = 2 \\ p_0(2) + p_3 + p_6 + p_8 = 2 \\ p_1(1) + p_5 + p_6 = 1 \\ p_1(1) + p_3 = 1 \end{cases}$$

Petri Nets' Flag (Correct Abstraction)

Introduction





$$E \triangleq \begin{cases} p_2 = p_1 \\ p_4 = p_3 + 1 \\ p_7 = p_6 \\ p_3 = p_5 + p_6 \end{cases}$$

Example of Classes

Introduction

- PR-R (state equation corresponds to the exact state-space)
- Flat nets (Presburger-definable)

Formalisation

Introduction

$$m_1 \equiv_E m_2 \quad \Leftrightarrow \quad \exists m \in \mathbb{N}^V \, . \; m \models E \wedge \underline{m_1} \wedge \underline{m_2}$$

Formalisation

Introduction

$$m_1 \equiv_E m_2 \quad \Leftrightarrow \quad \exists m \in \mathbb{N}^V \, . \, m \models E \wedge \underline{m_1} \wedge \underline{m_2}$$

Definition (E-abstraction)

$(N_1, m_1) \sqsupseteq_E (N_2, m_2)$ iff

(A1) initial markings are compatible with E, meaning $m_1 \equiv_E m_2$

(A2) for all observation sequences $\sigma \in \Sigma^\star$ such that $(N_1, m_1) \stackrel{\sigma}{\Rightarrow} (N_1, m_1')$

- there is at least one marking m_2' of N_2 such that $m_1' \equiv_E m_2'$
- for all markings m_2' we have that $m_1' \equiv_E m_2'$ implies $(N_2, m_2) \stackrel{\sigma}{\Rightarrow} (N_2, m_2')$

Formalisation

Introduction

$$m_1 \equiv_E m_2 \quad \Leftrightarrow \quad \exists m \in \mathbb{N}^V \,.\, m \models E \wedge \underline{m_1} \wedge \underline{m_2}$$

Definition (E-abstraction)

$(N_1, m_1) \sqsupseteq_E (N_2, m_2)$ iff

(A1) initial markings are compatible with E, meaning $m_1 \equiv_E m_2$

(A2) for all observation sequences $\sigma \in \Sigma^\star$ such that $(N_1, m_1) \overset{\sigma}{\Rightarrow} (N_1, m_1')$

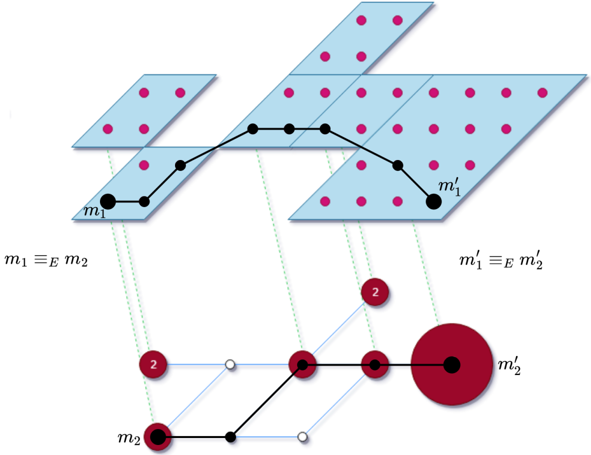
- there is at least one marking m_2' of N_2 such that $m_1' \equiv_E m_2'$
- for all markings m_2' we have that $m_1' \equiv_E m_2'$ implies $(N_2, m_2) \overset{\sigma}{\Rightarrow} (N_2, m_2')$

***E*-abstraction equivalence**

$(N_1, m_1) \equiv_E (N_2, m_2)$ iff $(N_1, m_1) \sqsupseteq_E (N_2, m_2)$ and $(N_2, m_2) \sqsupseteq_E (N_1, m_1)$

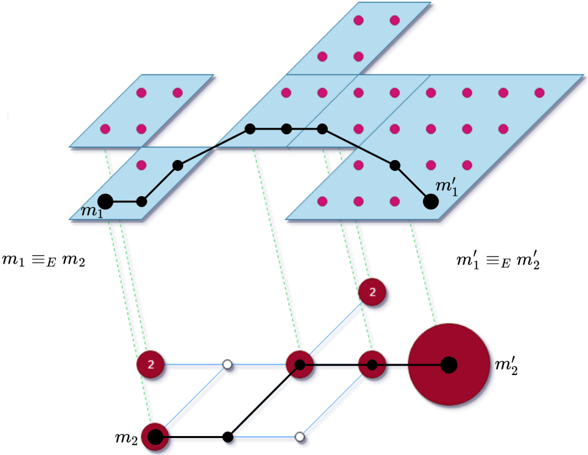
Formalisation

Introduction



Formalisation

Introduction



Not a bisimulation!

Not all pairs of reachable markings m'_1, m'_2 satisfy $(N_1, m'_1) \equiv_E (N_2, m'_2)$

(Un)decidability

Introduction

Theorem

The problem of checking whether a statement $(N_1, m_1) \equiv_E (N_2, m_2)$ *is valid is undecidable.*

Theorem

The problem of checking whether a statement $(N_1, m_1) \equiv_E (N_2, m_2)$ is valid is undecidable.

Proof.

- Take $(N_1, m_1) \equiv_{\mathrm{True}} (N_2, m_2)$, with $P_1 = P_2$

Theorem

The problem of checking whether a statement $(N_1, m_1) \equiv_E (N_2, m_2)$ is valid is undecidable.

Proof.

- Take $(N_1, m_1) \equiv_{\mathrm{True}} (N_2, m_2)$, with $P_1 = P_2$
- Both nets must have same reachability sets

(Un)decidability

Introduction

Theorem

The problem of checking whether a statement $(N_1, m_1) \equiv_E (N_2, m_2)$ is valid is undecidable.

Proof.

- Take $(N_1, m_1) \equiv_{\mathrm{True}} (N_2, m_2)$, with $P_1 = P_2$
- Both nets must have same reachability sets
- Checking marking equivalence is undecidable [Hack 76]

Use-cases
Introduction

- Model counting [Berthomieu et al. 2018]

Use-cases
Introduction

- Model counting [Berthomieu et al. 2018]
- Generalized Reachability Problem [Petri Nets 2021]

Use-cases

Introduction

- Model counting [Berthomieu et al. 2018]
- Generalized Reachability Problem [Petri Nets 2021]
- Concurrent Places Problem [SPIN 2021]

Use-cases
Introduction

- Model counting [Berthomieu et al. 2018]
- **Generalized Reachability Problem** [Petri Nets 2021]
- Concurrent Places Problem [SPIN 2021]

- Is F_1 reachable in (N_1, m_1)?

- Is F_1 reachable in (N_1, m_1)?

Definition (E-transform Formula)
Formula $F_2(\boldsymbol{p}_2) \triangleq \exists \boldsymbol{p}_1.\, \tilde{E}(\boldsymbol{p}_1, \boldsymbol{p}_2) \wedge F_1(\boldsymbol{p}_1)$ is the E-transform of F_1

- Is F_1 reachable in (N_1, m_1)?

Definition (E-transform Formula)

Formula $F_2(\boldsymbol{p}_2) \triangleq \exists \boldsymbol{p}_1 .\, \tilde{E}(\boldsymbol{p}_1, \boldsymbol{p}_2) \wedge F_1(\boldsymbol{p}_1)$ is the E-transform of F_1

- Is the E-transform formula F_2 reachable in (N_2, m_2)?

Challenges:

- Semi-procedure
- Parametric nets (N_1, C_1) and (N_2, C_2)

Challenges and Proposal

Challenges:

- Semi-procedure
- Parametric nets (N_1, C_1) and (N_2, C_2)

Proposal:

- More general notion of abstraction
- Presburger encoding of the τ transitions
- SMT constraints

Challenges and Proposal

Challenges:

- Semi-procedure
- Parametric nets (N_1, C_1) and (N_2, C_2)

Proposal:

- More general notion of abstraction
- Presburger encoding of the τ transitions
- SMT constraints

Is a reduction candidate $(N_1, C_1) >_E (N_2, C_2)$ correct?

Outline

Parametric Polyhedral Abstraction

Presburger Arithmetic and Flatness

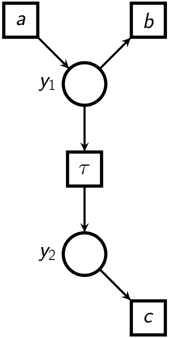
Core Requirements

Toolchain

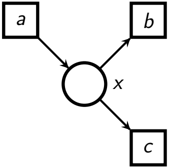
Discussion

Coherent Nets

Parametric Polyhedral Abstraction

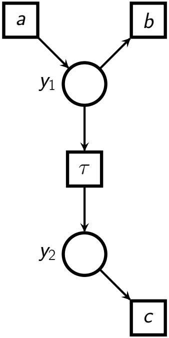


$$\approxeq_{x = y_1 + y_2}$$

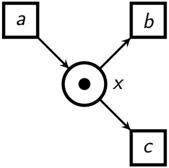


Coherent Nets

Parametric Polyhedral Abstraction



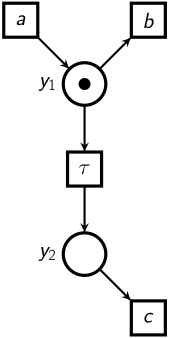
$$\approxeq_{x = y_1 + y_2}$$



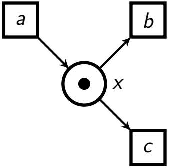
$$\sigma_2 = a$$

Coherent Nets

Parametric Polyhedral Abstraction



$$\approx_{x = y_1 + y_2}$$

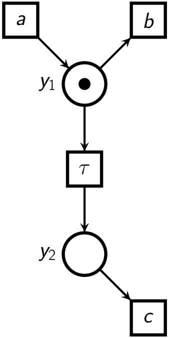


$\sigma_1 = a$

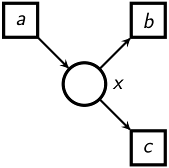
$\sigma_2 = a$

Coherent Nets

Parametric Polyhedral Abstraction



$$\approxeq_{x = y_1 + y_2}$$

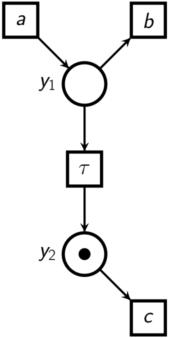


$\sigma_1 = a$

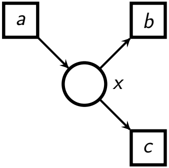
$\sigma_2 = a{\cdot}c$

Coherent Nets

Parametric Polyhedral Abstraction



$$\approxeq_{x = y_1 + y_2}$$

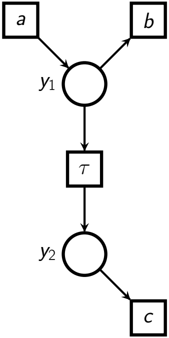


$\sigma_1 = a$

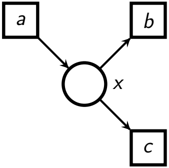
$\sigma_2 = a \cdot c$

Coherent Nets

Parametric Polyhedral Abstraction



$$\approxeq_{x = y_1 + y_2}$$

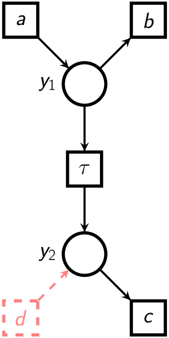


$\sigma_1 = a{\cdot}c$

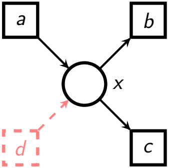
$\sigma_2 = a{\cdot}c$

Coherent Nets

Parametric Polyhedral Abstraction

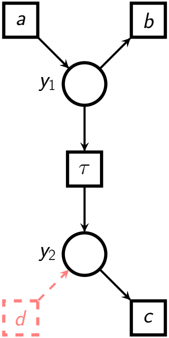


$$\approxeq x = y_1 + y_2$$

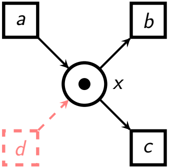


Coherent Nets

Parametric Polyhedral Abstraction



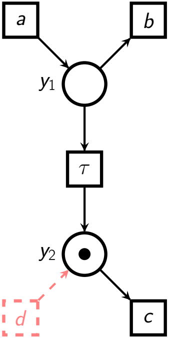
$$\approxeq_{x = y_1 + y_2}$$



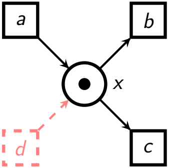
$$\sigma_2 \triangleq d$$

Coherent Nets

Parametric Polyhedral Abstraction



$\approxeq_{x = y_1 + y_2}$

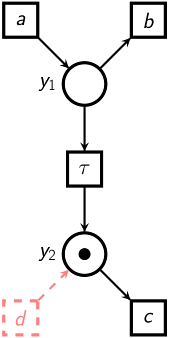


$\sigma_1 \triangleq d$

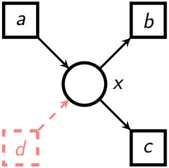
$\sigma_2 \triangleq d$

Coherent Nets

Parametric Polyhedral Abstraction



$$\approxeq x = y_1 + y_2$$

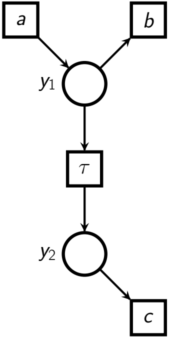


$\sigma_1 \triangleq d$

$\sigma_2 \triangleq d \cdot b$

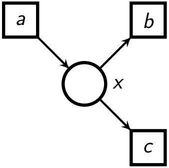
Coherent Nets

Parametric Polyhedral Abstraction



$C_1 \triangleq y_2 = 0$

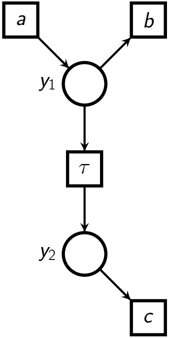
$$\approxeq_{x = y_1 + y_2}$$



$C_2 \triangleq \mathrm{True}$

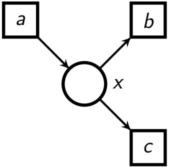
Coherent Nets

Parametric Polyhedral Abstraction



$C_1 \triangleq y_2 = 0$

$$\approxeq_{x = y_1 + y_2}$$

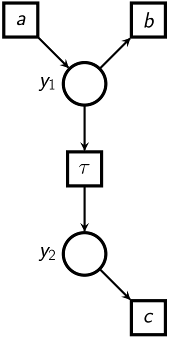


$C_2 \triangleq \mathrm{True}$

Equivalence rule (concat), $(N_1, C_1) \approxeq_E (N_2, C_2)$ with $E \triangleq (x = y_1 + y_2)$.

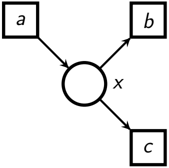
Coherent Nets

Parametric Polyhedral Abstraction



$C_1 \triangleq y_2 = 0$

$$\approxeq_{x = y_1 + y_2}$$



$C_2 \triangleq \mathrm{True}$

Equivalence rule (concat), $(N_1, C_1) \approxeq_E (N_2, C_2)$ with $E \triangleq (x = y_1 + y_2)$.

Remark: τ transitions may be irreversible choices

Coherent Nets

Parametric Polyhedral Abstraction

We introduce some **coherency constraints** C

- hold on the initial state
- sufficient large subset of reachable markings

Coherent Nets

Parametric Polyhedral Abstraction

We introduce some **coherency constraints** C

- hold on the initial state
- sufficient large subset of reachable markings

$m \overset{\sigma\rangle}{\Longrightarrow} m'$: do not finish with a τ transition

Coherent Nets

We introduce some **coherency constraints** C

- hold on the initial state
- sufficient large subset of reachable markings

$m \stackrel{\sigma\rangle}{\Longrightarrow} m'$: do not finish with a τ transition

Definition (Coherent Net (N,C))

For all firing sequences $m \stackrel{\sigma}{\Rightarrow} m'$ with $m \in C$ we have:

$$\exists m'' \in C \;.\; m \stackrel{\sigma\rangle}{\Longrightarrow} m'' \wedge m'' \stackrel{\epsilon}{\Rightarrow} m'$$

We introduce some **coherency constraints** C

- hold on the initial state
- sufficient large subset of reachable markings

$m \overset{\sigma\rangle}{\Longrightarrow} m'$: do not finish with a τ transition

Definition (Coherent Net (N,C))

For all firing sequences $m \overset{\sigma}{\Rightarrow} m'$ with $m \in C$ we have:

$$\exists m'' \in C \;.\; m \overset{\sigma\rangle}{\Longrightarrow} m'' \wedge m'' \overset{\epsilon}{\Rightarrow} m'$$

Can reach a coherent marking by firing the "necessary" τ transitions

$$m_1 \langle C_1 E C_2 \rangle\, m_2 \triangleq m_1 \models C_1 \wedge m_1 \equiv_E m_2 \wedge m_2 \models C_2$$

Definition (Parametric E-abstraction)

$(N_1, C_1) \preceq_E (N_2, C_2)$ iff

(S1) For all markings m_1 satisfying C_1 there exists a marking m_2 such that $m_1 \langle C_1 E C_2 \rangle\, m_2$.

(S2) For all firing sequences $m_1 \overset{\epsilon}{\Rightarrow} m_1'$ and all markings m_2, we have $m_1 \equiv_E m_2$ implies $m_1' \equiv_E m_2$.

(S3) For all firing sequences $m_1 \overset{\sigma}{\Rightarrow} m_1'$ and all marking pairs m_2, m_2', if $m_1 \langle C_1 E C_2 \rangle\, m_2$ and $m_1' \equiv_E m_2'$ then we have $m_2 \overset{\sigma}{\Rightarrow} m_2'$.

Parametric Abstraction

Parametric Polyhedral Abstraction

$$m_1 \langle C_1 E C_2 \rangle m_2 \triangleq m_1 \models C_1 \land m_1 \equiv_E m_2 \land m_2 \models C_2$$

Definition (Parametric E-abstraction)

$(N_1, C_1) \preceq_E (N_2, C_2)$ iff

(S1) For all markings m_1 satisfying C_1 there exists a marking m_2 such that $m_1 \langle C_1 E C_2 \rangle m_2$.

(S2) For all firing sequences $m_1 \overset{\epsilon}{\Rightarrow} m_1'$ and all markings m_2, we have $m_1 \equiv_E m_2$ implies $m_1' \equiv_E m_2$.

(S3) For all firing sequences $m_1 \overset{\sigma}{\Rightarrow} m_1'$ and all marking pairs m_2, m_2', if $m_1 \langle C_1 E C_2 \rangle m_2$ and $m_1' \equiv_E m_2'$ then we have $m_2 \overset{\sigma}{\Rightarrow} m_2'$.

$(N_1, C_1) \approxeq_E (N_2, C_2)$ iff $(N_1, C_1) \preceq_E (N_2, C_2)$ and $(N_2, C_2) \preceq_E (N_1, C_1)$.

Theorem (Parametric E-abstraction Instantiation)

Assume $(N_1, C_1) \preceq_E (N_2, C_2)$ is a parametric E-abstraction. Then for every pair of markings m_1, m_2, $m_1 \langle C_1 E C_2 \rangle m_2$ implies $(N_1, m_1) \sqsubseteq_E (N_2, m_2)$.

Outline

Silent State-space

To prove $(N_1, C_1) \approxeq_E (N_2, C_2)$ we need to express $m \stackrel{\epsilon}{\Rightarrow} m'$

Silent State-space

To prove $(N_1, C_1) \approxeq_E (N_2, C_2)$ we need to express $m \overset{\epsilon}{\Rightarrow} m'$

A Preburger predicate, say τ^*_C such that

$$R_\tau(N, C) = \{m' \mid m' \models \exists \boldsymbol{x} \ . \ C(\boldsymbol{x}) \wedge \tau^*_C(\boldsymbol{x}, \boldsymbol{x'})\}$$

Silent State-space

To prove $(N_1, C_1) \approxeq_E (N_2, C_2)$ we need to express $m \stackrel{\epsilon}{\Rightarrow} m'$

A Preburger predicate, say τ_C^* such that

$$R_\tau(N, C) = \{m' \mid m' \models \exists \boldsymbol{x} \ . \ C(\boldsymbol{x}) \wedge \tau_C^*(\boldsymbol{x}, \boldsymbol{x'})\}$$

Theorem

Given a parametric E-abstraction equivalence $(N_1, C_1) \approxeq_E (N_2, C_2)$, the silent reachability set $R_\tau(N_1, C_1)$ is Presburger-definable.

Theorem (Leroux, 2013)

For every VASS V, for every Presburger set C_{in} of configurations, the reachability set $\mathrm{ReachV}(C_{in})$ is Presburger if, and only if, V is flattable from C_{in}.

Theorem (Leroux, 2013)

For every VASS V, for every Presburger set C_{in} of configurations, the reachability set $\mathrm{ReachV}(C_{in})$ is Presburger if, and only if, V is flattable from C_{in}.

If candidate correct: we have methods to compute τ_C^*

Theorem (Leroux, 2013)
For every VASS V, for every Presburger set C_{in} of configurations, the reachability set $\mathrm{ReachV}(C_{in})$ is Presburger if, and only if, V is flattable from C_{in}.

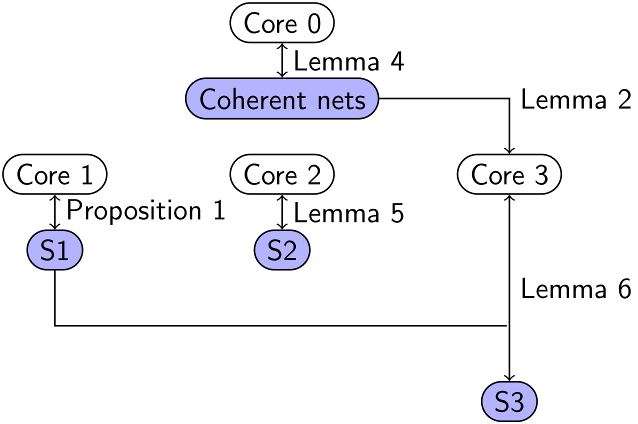
If candidate correct: we have methods to compute τ_C^*

But, checking flatness is undecidable $\rightarrow$ semi-procedure

Outline

Big Picture

Core Requirements

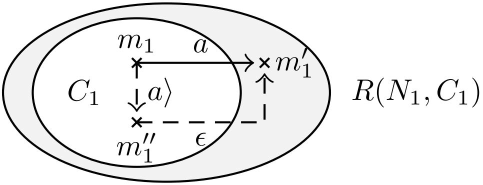


Core 0 — (Coherent Net)

Core Requirements

(Coherent net) For all firing sequences $m \overset{\sigma}{\Rightarrow} m'$ with $m \in C$:

$$\exists m'' \in C \ . \ m \overset{\sigma\rangle}{\Longrightarrow} m'' \wedge m'' \overset{\epsilon}{\Rightarrow} m'$$

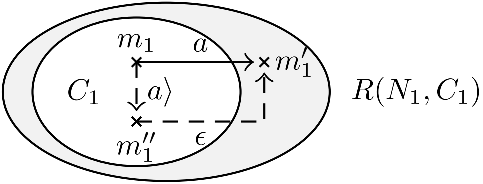


Core 0 — (Coherent Net)

Core Requirements

(Coherent net) For all firing sequences $m \overset{\sigma}{\Rightarrow} m'$ with $m \in C$:

$$\exists m'' \in C \;.\; m \overset{\sigma\rangle}{\Longrightarrow} m'' \wedge m'' \overset{\epsilon}{\Rightarrow} m'$$



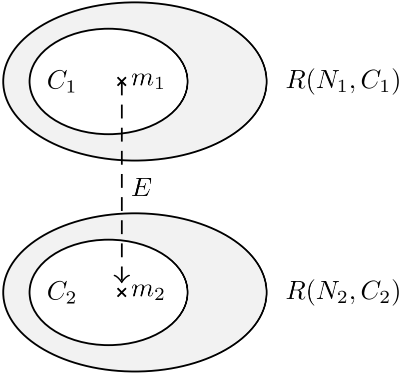
$$\forall \boldsymbol{p}, \boldsymbol{p'}, a \;.\; C(\boldsymbol{p}) \wedge \acute{T}_C(\boldsymbol{p}, \boldsymbol{p'}, a) \\ \implies \exists \boldsymbol{p''} \;.\; C(\boldsymbol{p''}) \wedge \acute{T}_C(\boldsymbol{p}, \boldsymbol{p''}, a) \wedge \tau^*_C(\boldsymbol{p''}, \boldsymbol{p'})$$

Core 1 — (S1)

Core Requirements

(S1) For all markings m_1 satisfying C_1:

$$\exists m_2 \ . \ m_1 \langle C_1 E C_2 \rangle m_2$$

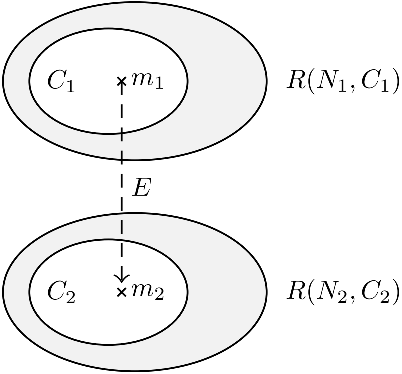


Core 1 — (S1)

Core Requirements

(S1) For all markings m_1 satisfying C_1:

$$\exists m_2 \ . \ m_1 \langle C_1 E C_2 \rangle m_2$$



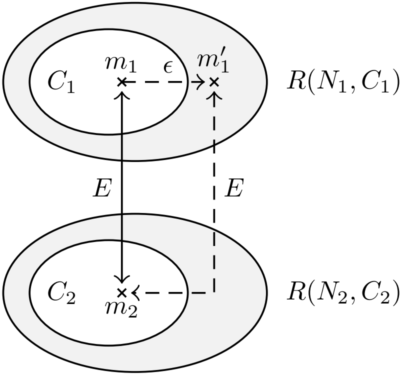
$$\forall \boldsymbol{x} \ . \ C_1(\boldsymbol{x}) \implies \exists \boldsymbol{y} \ . \ \tilde{E}(\boldsymbol{x}, \boldsymbol{y}) \wedge C_2(\boldsymbol{y})$$

Core 2 — (S2)

Core Requirements

(S2) For all firing sequences $m_1 \overset{\epsilon}{\Rightarrow} m_1'$ and all markings m_2:

$$m_1 \equiv_E m_2 \implies m_1' \equiv_E m_2$$

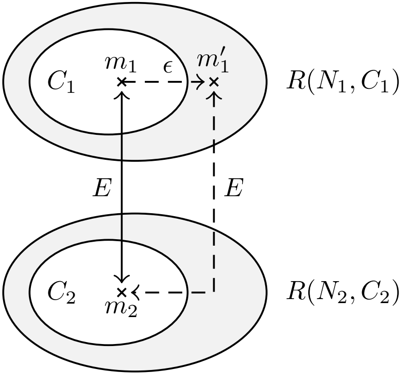


Core 2 — (S2)

Core Requirements

(S2) For all firing sequences $m_1 \overset{\epsilon}{\Rightarrow} m_1'$ and all markings m_2:

$$m_1 \equiv_E m_2 \implies m_1' \equiv_E m_2$$



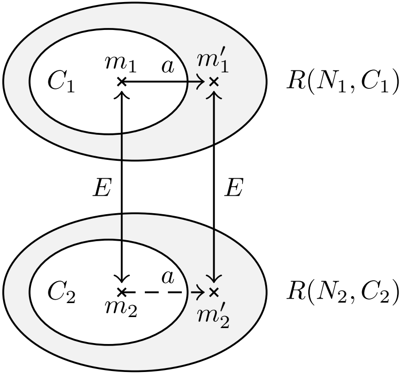
$$\forall \boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_1' \ . \ \tilde{E}(\boldsymbol{p}_1, \boldsymbol{p}_2) \wedge \tau(\boldsymbol{p}_1, \boldsymbol{p}_1') \implies \tilde{E}(\boldsymbol{p}_1', \boldsymbol{p}_2)$$

Core 3 — (S3)

Core Requirements

(S3) For all firing sequences $m_1 \overset{\sigma}{\Rightarrow} m_1'$ and all marking pairs m_2, m_2':

$$m_1 \langle C_1 E C_2 \rangle\, m_2 \wedge m_1' \equiv_E m_2' \implies m_2 \overset{\sigma}{\Rightarrow} m_2'$$

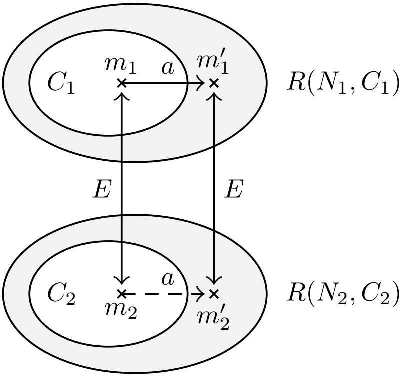


Core 3 — (S3)

Core Requirements

(S3) For all firing sequences $m_1 \overset{\sigma}{\Rightarrow} m'_1$ and all marking pairs m_2, m'_2:

$$m_1 \langle C_1 E C_2 \rangle\, m_2 \wedge m'_1 \equiv_E m'_2 \implies m_2 \overset{\sigma}{\Rightarrow} m'_2$$



$$\forall \boldsymbol{p}_1, \boldsymbol{p}_2, a, \boldsymbol{p}'_1, \boldsymbol{p}'_2 \;.\; \langle C_1 E C_2 \rangle(\boldsymbol{p}_1, \boldsymbol{p}_2) \wedge \hat{T}_{C_1}(\boldsymbol{p}_1, \boldsymbol{p}'_1) \wedge \tilde{E}(\boldsymbol{p}'_1, \boldsymbol{p}'_2)$$
$$\implies \hat{T}_{C_2}(\boldsymbol{p}_2, \boldsymbol{p}'_2)$$

Outline

Reductron
Toolchain

REDUCTRON

THE POLYHEDRAL ABSTRACTION PROVER

github.com/nicolasAmat/Reductron

Reductron

Toolchain

REDUCTRON
THE POLYHEDRAL ABSTRACTION PROVER

github.com/nicolasAmat/Reductron

- Compute τ_C^* using the tool FAST
- LIA theory in z3 (use SMT-LIB)

Reductron

Toolchain

REDUCTRON

THE POLYHEDRAL ABSTRACTION PROVER

github.com/nicolasAmat/Reductron

- Compute τ_C^* using the tool FAST
- LIA theory in z3 (use SMT-LIB)
- Allowed us to prove all our reduction rules!

Outline

Discussion About Automated Proving

- Consolidates reliability (for Tina and SMPT model checkers)

Discussion About Automated Proving

- Consolidates reliability (for Tina and SMPT model checkers)
- Better **understanding** of what's **behind** polyhedral reduction

Discussion About Automated Proving

- Consolidates reliability (for Tina and SMPT model checkers)
- Better **understanding** of what's **behind** polyhedral reduction
- A tool to experiment with **new reduction rules**

Discussion About Automated Proving

- Consolidates reliability (for Tina and SMPT model checkers)
- Better **understanding** of what's **behind** polyhedral reduction
- A tool to experiment with **new reduction rules**
- Concrete use-case of the **"flattable"** notion

Discussion About Polyhedral Abstraction

- Many nets are flat, actually all bounded models are flat
  But it is difficult to find the equation system E
- We show that we can **find pieces of flatness** inside the reachable markings of nets
  This is the meaning of our polyhedral abstraction
- We can exhibit such equivalences using structural reductions

Thank you for your attention!

github.com/nicolasAmat/Reductron

Any questions?