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

\[ E \triangleq \left\{ \begin{array}{l}
\text{Cabins} + \text{Dress} + \text{Dressed} + \text{Undress} + \text{WaitBag} = 10 \\
\text{Dress} + \text{Dressed} + \text{Entered} + \text{InBath} + \text{Out} + \text{Undress} + \text{WaitBag} = 20 \\
\text{Bags} + \text{Dress} + \text{InBath} + \text{Undress} = 15
\end{array} \right. \]
Petri Nets’ Flag (Incorrect Abstraction)

Introduction

\[ E \triangleq \begin{cases} 
  p_2 = p_1 \\
  p_4 = p_3 \\
  p_7 = p_6 \\
  p_0 + p_5 + 2p_6 + p_8 = 2 \\
  p_0 + p_3 + p_6 + p_8 = 2 \\
  p_1 + p_5 + p_6 = 1 \\
  p_1 + p_3 = 1 
\end{cases} \]
Petri Nets’ Flag (Incorrect Abstraction)

Introduction

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

\[ m_1 \equiv_E m_2 \iff \exists m \in \mathbb{N}^V . m \models E \land \overline{m_1} \land \overline{m_2} \]
Formalisation

Introduction

\[ m_1 \equiv_E m_2 \iff \exists m \in \mathbb{N}^V. m \models E \land m_1 \land m_2 \]

Definition (\(E\)-abstraction)

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

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

(A2) for all observation sequences \(\sigma \in \Sigma^*\) such that \((N_1, m_1) \xrightarrow{\sigma} (N_1, m'_1)\)

\(\Rightarrow\) there is at least one marking \(m'_2\) of \(N_2\) such that \(m'_1 \equiv_E m'_2\)

\(\Rightarrow\) for all markings \(m'_2\) we have that \(m'_1 \equiv_E m'_2\) implies \((N_2, m_2) \xrightarrow{\sigma} (N_2, m'_2)\)
Formalisation

Introduction

\[ m_1 \equiv_E m_2 \iff \exists m \in \mathbb{N}^V. m \models E \land m_1 \land m_2 \]

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 \(\sigma \in \Sigma^*\) such that \((N_1, m_1) \xrightarrow{\sigma} (N_1, m'_1)\)
  - 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 \((N_2, m_2) \xrightarrow{\sigma} (N_2, m'_2)\)

\(E\)-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

Not a bisimulation!

Not all pairs of reachable markings $m_1', m_2'$ satisfy $(N_1, m_1') \equiv E (N_2, m_2')$.
Not a bisimulation!
Not all pairs of reachable markings $m'_1$, $m'_2$ satisfy $(N_1, m'_1) \equiv_E (N_2, m'_2)$
Theorem

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

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

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

Proof.

- Take $(N_1, m_1) \equiv_{\text{True}} (N_2, m_2)$, 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]
Use-cases

Introduction

▶ Is $F_1$ reachable in $(N_1, m_1)$?
Is $F_1$ reachable in $(N_1, m_1)$?

**Definition (E-transform Formula)**

Formula $F_2(p_2) \triangleq \exists p_1. \tilde{E}(p_1, p_2) \land F_1(p_1)$ is the $E$-transform of $F_1$.
Use-cases

Introduction

▶ Is $F_1$ reachable in $(N_1, m_1)$?

**Definition (E-transform Formula)**

Formula $F_2(p_2) \triangleq \exists p_1. \tilde{E}(p_1, p_2) \land F_1(p_1)$ is the E-transform of $F_1$

▶ Is the E-transform formula $F_2$ reachable in $(N_2, m_2)$?
Challenges and Proposal

Introduction

Challenges:

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

Proposal:

- More general notion of abstraction
- Presburger encoding of the \(\tau\) transitions
- SMT constraints

Is a reduction candidate \((N_1, C_1) > E(N_2, C_2)\) correct?
Challenges and Proposal

Introduction

Challenges:

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

Proposal:

- More general notion of abstraction
- Presburger encoding of the \(\tau\) transitions
- SMT constraints
Challenges and Proposal

Introduction

Challenges:

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

Proposal:

- More general notion of abstraction
- Presburger encoding of the \(\tau\) transitions
- SMT constraints

Is a reduction candidate \((N_1, C_1) \geq_E (N_2, C_2)\) correct?
Outline

Parametric Polyhedral Abstraction

Presburger Arithmetic and Flatness

Core Requirements

Toolchain

Discussion
Coherent Nets
Parametric Polyhedral Abstraction

\[ x = y_1 + y_2 \approx \]

Remark: \( \tau \) transitions may be irreversible choices.
Coherent Nets
Parametric Polyhedral Abstraction

\[ x \approx y_1 + y_2 \]

\[ \sigma_2 = a \]
Coherent Nets
Parametric Polyhedral Abstraction

\[ \sigma_1 = a \]

\[ \sigma_2 = a \]

\[ \tau \text{ transitions may be irreversible choices} \]
Coherent Nets
Parametric Polyhedral Abstraction

\( \tau \rightarrow y_1 \)
\( y_2 \rightarrow \tau \rightarrow \)

\( \sigma_1 = a \)

\( \approx x = y_1 + y_2 \)

\( \sigma_2 = a \cdot c \)
Coherent Nets
Parametric Polyhedral Abstraction

\[ x = y_1 + y_2 \]

\[ \sigma_1 = a \]

\[ \sigma_2 = a \cdot c \]
Coherent Nets
Parametric Polyhedral Abstraction

\[ x = y_1 + y_2 \]

\[ \sigma_1 = a \cdot c \]

\[ \sigma_2 = a \cdot c \]
Coherent Nets
Parametric Polyhedral Abstraction

\[ \approx x = y_1 + y_2 \]
Coherent Nets

Parametric Polyhedral Abstraction

\[ x \approx y_1 + y_2 \]

Remark: \( \tau \) transitions may be irreversible choices.
Coherent Nets

Parametric Polyhedral Abstraction

\[ \sigma_1 \triangleq d \]

\[ \sigma_2 \triangleq d \]

\[ \approx x = y_1 + y_2 \]
Coherent Nets
Parametric Polyhedral Abstraction

\[ \approx x = y_1 + y_2 \]

\( \sigma_1 \triangleq d \)

\( \sigma_2 \triangleq d \cdot b \)
Coherent Nets
Parametric Polyhedral Abstraction

\[ C_1 \triangleq y_2 = 0 \]

\[ C_2 \triangleq \text{True} \]

\[ x \approx y_1 + y_2 \]
Coherent Nets
Parametric Polyhedral Abstraction

\[
\begin{align*}
C_1 & \triangleq y_2 = 0 \\
C_2 & \triangleq \text{True}
\end{align*}
\]

\[\approx x = y_1 + y_2\]

Equivalence rule (concat), \((N_1, C_1) \approx_E (N_2, C_2)\) with \(E \triangleq (x = y_1 + y_2)\).
Coherent Nets
Parametric Polyhedral Abstraction

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

Remark: \(\tau\) transitions may be irreversible choices
We introduce some coherency constraints \( C \)

- hold on the initial state
- sufficient large subset of reachable markings
We introduce some **coherency constraints** $C$

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

$m \xrightarrow{\sigma'} m'$: do not finish with a $\tau$ transition
Coherent Nets
Parametric Polyhedral Abstraction

We introduce some **coherency constraints** $C$

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

$m \xrightarrow{\sigma} m'$: do not finish with a $\tau$ transition

**Definition (Coherent Net (N,C))**
For all firing sequences $m \xrightarrow{\sigma} m'$ with $m \in C$ we have:

$$\exists m'' \in C . \ m \xrightarrow{\sigma} m'' \land m'' \xrightarrow{\xi} m'$$
Coherent Nets
Parametric Polyhedral Abstraction

We introduce some **coherency constraints** \( C \)
- hold on the initial state
- sufficient large subset of reachable markings

\[ m \xrightarrow{\sigma^\dagger} m' : \text{do not finish with a } \tau \text{ transition} \]

**Definition (Coherent Net \((N,C)\))**
For all firing sequences \( m \xrightarrow{\sigma} m' \) with \( m \in C \) we have:

\[ \exists m'' \in C . m \xrightarrow{\sigma^\dagger} m'' \land m'' \xrightarrow{\xi} m' \]

Can reach a coherent marking by firing the “necessary” \( \tau \) transitions
\[
m_1 \langle C_1 EC_2 \rangle m_2 \triangleq m_1 \models C_1 \land m_1 \equiv_E m_2 \land m_2 \models C_2
\]

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 \(m_1 \langle C_1 EC_2 \rangle m_2\).

(S2) For all firing sequences \(m_1 \xrightarrow{\sigma} m'_1\) 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 \(m_1 \xrightarrow{\sigma} m'_1\) and all marking pairs \(m_2, m'_2\), if \(m_1 \langle C_1 EC_2 \rangle m_2\) and \(m'_1 \equiv_E m'_2\) then we have \(m_2 \xrightarrow{\sigma} m'_2\).
Parametric Abstraction
Parametric Polyhedral Abstraction

\[ m_1 \langle C_1 EC_2 \rangle m_2 \triangleq m_1 \models C_1 \land m_1 \equiv_E m_2 \land m_2 \models C_2 \]

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 
\( m_1 \langle C_1 EC_2 \rangle m_2 \).

(S2) For all firing sequences \( m_1 \Rightarrow m'_1 \) 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 \( m_1 \Rightarrow m'_1 \) and all marking pairs \( m_2, m'_2 \), if 
\( m_1 \langle C_1 EC_2 \rangle m_2 \) and \( m'_1 \equiv_E m'_2 \) then we have \( m_2 \Rightarrow m'_2 \).

\((N_1, C_1) \equiv_E (N_2, C_2) \) 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 $m_1, m_2$, $m_1 \langle C_1 EC_2 \rangle m_2$ implies $(N_1, m_1) \sqsubseteq_E (N_2, m_2)$. 
Outline

Parametric Polyhedral Abstraction

Presburger Arithmetic and Flatness

Core Requirements

Toolchain

Discussion
To prove \((N_1, C_1) \simeq_E (N_2, C_2)\) we need to express \(m \Rightarrow ^e m'\)
Silent State-space

To prove \((N_1, C_1) \equiv_E (N_2, C_2)\) we need to express \(m \xrightarrow{\epsilon} m'\)

A Preburger predicate, say \(\tau^*_C\) such that

\[
R_{\tau}(N, C) = \{ m' \mid m' \models \exists x . C(x) \land \tau^*_C(x, x') \}
\]
Silent State-space

To prove \((N_1, C_1) \equiv_E (N_2, C_2)\) we need to express \(m \Rightarrow m'\)

A Presburger predicate, say \(\tau^*_C\) such that

\[
R_\tau(N, C) = \{ m' \mid m' \models \exists x \cdot C(x) \land \tau^*_C(x, x') \}
\]

Theorem

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

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

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

If candidate correct: we have methods to compute $\tau_C^*$
**Theorem (Leroux, 2013)**

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

If candidate correct: we have methods to compute $\tau^*_C$

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

Parametric Polyhedral Abstraction

Presburger Arithmetic and Flatness

Core Requirements

Toolchain

Discussion
Big Picture
Core Requirements

Coherent nets

Core 0

Lemma 4

Core 1

Proposition 1

S1

Core 2

Lemma 5

S2

Core 3

Lemma 2

S3

Lemma 6
Core 0 — (Coherent Net)

Core Requirements

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

$$\exists m'' \in C . \; m \xrightarrow{\sigma} m'' \land m'' \xrightarrow{\epsilon} m'$$

[Diagram showing the relationship between firing sequences and transitions in a coherent net.]
Core Requirements

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

$$\exists m'' \in C . m \xrightarrow{\sigma} m'' \land m'' \xrightarrow{\epsilon} m'$$

$$\forall p, p', a . C(p) \land \dot{T}_C(p, p', a) \implies \exists p'' . C(p'') \land \dot{T}_C(p, p'', a) \land \tau^*_C(p'', p')$$
Core 1 — (S1)

Core Requirements

(S1) For all markings $m_1$ satisfying $C_1$:

$$\exists m_2 . \; m_1 \langle C_1 EC_2 \rangle m_2$$

\[
\begin{align*}
C_1 &\quad \times m_1 \\
E &\quad R(N_1, C_1) \\
C_2 &\quad \downarrow m_2 \\
&\quad \downarrow E \\
&\quad \downarrow R(N_2, C_2)
\end{align*}
\]
Core 1 — (S1)

Core Requirements

(S1) For all markings $m_1$ satisfying $C_1$:

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

$$\forall x \cdot C_1(x) \implies \exists y \cdot \tilde{E}(x, y) \land C_2(y)$$
Core 2 — (S2)

Core Requirements

(S2) For all firing sequences $m_1 \xrightarrow{\epsilon} m_1'$ 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 $m_1 \xrightarrow{\xi} m'_1$ and all markings $m_2$:

$$m_1 \equiv_E m_2 \implies m'_1 \equiv_E m_2$$

$$\forall p_1, p_2, p'_1 . \tilde{E}(p_1, p_2) \land \tau(p_1, p'_1) \implies \tilde{E}(p'_1, p_2)$$
Core Requirements

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

$$m_1 \langle C_1 EC_2 \rangle m_2 \land m'_1 \equiv_E m'_2 \implies m_2 \xrightarrow{\sigma} m'_2$$
Core 3 — (S3) Core Requirements

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

$$m_1 \langle C_1 EC_2 \rangle m_2 \land m'_1 \equiv_E m'_2 \implies m_2 \xrightarrow{\sigma} m'_2$$

∀ $p_1, p_2, a, p'_1, p'_2$. $\langle C_1 EC_2 \rangle(p_1, p_2) \land \hat{T}_{C_1}(p_1, p'_1) \land \tilde{E}(p'_1, p'_2)$

$$\implies \hat{T}_{C_2}(p_2, p'_2)$$
Outline

Parametric Polyhedral Abstraction

Presburger Arithmetic and Flatness

Core Requirements

Toolchain

Discussion
Reductron
Toolchain

Compute $\tau^*$ using the tool FAST

▶ LIA theory in z3 (use SMT-LIB)

Allowed us to prove all our reduction rules!

github.com/nicolasAmat/Reductron
Compute $\tau^*_C$ using the tool FAST

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

LIA theory in z3 (use SMT-LIB)

Allowed us to prove all our reduction rules!
Outline

Parametric Polyhedral Abstraction

Presburger Arithmetic and Flatness

Core Requirements

Toolchain

Discussion
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?