A Generic Approach to the Static Analysis of Concurrent Programs with Procedures

Ahmed Bouajjani, Javier Esparza, Tayssir Touili

POPL ’03

Presented by Anjo Vahldiek
Motivation

Is a global configuration $n_3, m_3$ from $n_0, m_0$ reachable?

Procedure $\pi_1$
- Send using channel $a$
  - $a!$

Procedure $\pi_2$
- Call $\pi_2$
  - $a?$
- Wait on channel $b$
  - $b?$
Motivation

Procedure $\pi_1$

- $n_0$ to $n_1$
- $n_1$ to $n_2$
- $n_2$ to $n_0$
- Call $\pi_2$

Procedure $\pi_2$

- $m_0$ to $m_1$
- $m_1$ to $m_2$
- $m_2$ to $m_0$
- a? to b?
Motivation

Procedure $\pi_1$

Procedure $\pi_2$

Call $\pi_2$
Motivation

Procedure $\pi_1$

Procedure $\pi_2$

Call $\pi_2$
Agenda

* Background
* Abstract Framework
  * Informal Abstractions
  * Formal Framework
  * Examples
* A General Approach
Agenda

* Background
* Abstract Framework
  * Informal Abstractions
  * Formal Framework
  * Examples
* A General Approach
Previous Related Work

* Special Case Analysis:
  * intraprocedural with synchronization
  * Interprocedural in synchronization-free case
  * Reachability proven to be undecidable
Example: Undecidability

Initial Configuration

Final Configuration

$L(c_0, c) = \{a^n b^n \mid n \geq 0\}$

Procedure $\pi_1$

Procedure $\pi_2$

Call $\pi_1$

Call $\pi_2$
Usage of Concurrent Analysis

- Code Optimization at compile-time
- Safety Properties
  - Data Races
  - Deadlock detection
Abstract Interpretation

* Complete Lattice of language \( \text{Lab} \):
  \[ \mathcal{L} = (2^{\text{Lab}}, \subseteq, \cup, \cap, \emptyset, \text{Lab}) \]

* Abstract Lattice:
  \[ \mathcal{M} = (M, \subseteq, \cup, \cap, \bot, \top) \]

* Galois connection \((\alpha, \gamma)\) between \(\mathcal{L}\) and \(\mathcal{M}\) with \(\alpha: \mathcal{L} \to \mathcal{M}\) and \(\gamma: \mathcal{M} \to \mathcal{L}\) such that

\[
\forall x \in \mathcal{L}, \forall y \in \mathcal{M}. \alpha(x) \subseteq y \implies x \subseteq \gamma(y)
\]
Galois connection

\[ \gamma(m) \]

\[ \alpha(l) \]

\[ \gamma \]

\[ \alpha \]
Agenda

- Background
- Abstract Framework
  - Informal Abstractions
  - Formal Framework
  - Examples
- A General Approach
Approximate Synchronization

Lab, a set containing synchronization actions between two control locations

\[ L_i = L(C'_i, C_i) \] set of paths leading in control location \( i \) from configuration \( C'_i \) to \( C_i \)

Reachability of two sets of configurations \( (C_1 \times \cdots \times C_n \text{ and } C'_1 \times \cdots \times C'_n) \)

\[ L_1 \cap \cdots \cap L_n \neq \emptyset \]
Abstract Framework

* $L_1 \cap \cdots \cap L_n \neq \emptyset$ is undecidable
* **Find an abstraction** $A(C', C)$ for path language $L(C', C)$
* If the **intersection** of the abstractions **is empty**
  $\Rightarrow C$ is **not reachable** from $C'$
Agenda

- Background
- Abstract Framework
  - Informal Abstractions
  - Formal Framework
  - Examples
- A General Approach
Example 1: First Occurrence Ordering

- \( A(C', C) \) set \( S \) of words \( w \)
- \( w = a_1 \ldots a_n \in S \)
  \( \iff \exists \) a path in \( L(C', C) \) such that the set of synchronizations is \( \{a_1, \ldots, a_n\} \)
- Ordering of \( w \) is the first occurrence
- Intersection of two languages is determined by intersecting the sets \( S \)

\[ S(n_3) = \{ab\} \]
Example 1: First Occurrence Ordering

Is a global configuration \( n_3, m_3 \) from \( n_0, m_0 \) reachable?

Procedure \( \pi_1 \)

Procedure \( \pi_2 \)

Call \( \pi_2 \)
Example 1: First Occurrence Ordering

Procedure $\pi_1$

$S(n_3) = \{ab\}$
Example 1: First Occurrence Ordering

Procedure $\pi_1$

$S(n_3) = \{ab, b\}$
Example 1: First Occurrence Ordering

\[ S(m_3) = \{ba\} \]

Procedure \( \pi_2 \)

Call \( \pi_2 \)

Anjo Vahldiek
Example 1: First Occurrence Ordering

Procedure $\pi_1$

$S(n_3) = \{ab, b\}$

$S(m_3) = \{ba\}$

$S(n_3) \cap S(m_3) = \emptyset$

Procedure $\pi_2$

Call $\pi_2$
Example 2: Label Bitvectors

- Forget about the first occurrence
- $A(C', C)$ is a set $S$ of bitvectors $Lab \rightarrow \mathbb{B}$
- Bitvector $b$ belongs to $S \iff$ a path in $L(C', C)$ such that $b(a) = 1$ if it occurs, otherwise $b(a) = 0$ ($a \in Lab$)

$S(n_3) = \{(1)\}$
**Example 2: Label Bitvectors**

- Forget about the first occurrence
- $A(C', C)$ is a set $S$ of bitvectors $Lab \rightarrow \mathbb{B}$
- Bitvector $b$ belongs to $S \iff$ a path in $L(C', C)$ such that $b(a) = 1$ if it occurs, otherwise $b(a) = 0$ ($a \in Lab$)

![Diagram]

$S(n_3) = \{(1), (0)\}$
Is a global configuration $n_3, m_3$ from $n_0, m_0$ reachable?

**Procedure $\pi_1$**

- $n_0$
- $n_1$
- $n_2$
- $n_3$

**Procedure $\pi_2$**

- $m_0$
- $m_1$
- $m_2$
- $m_3$

Call $\pi_2$ a? b?
Label Bitvectors: Example 1

\[ S(n_3) = \{ (1), (0, 1) \} \]
Label Bitvectors: Example 1

\[ S(m_3) = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \]

Procedure \( \pi_2 \)

Call \( \pi_2 \)
Label Bitvectors: Example 1

\[ S(n_3) = \{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \} \]
\[ S(m_3) = \{ \begin{pmatrix} 1 \\ 1 \end{pmatrix} \} \]
\[ S(n_3) \cap S(m_3) = \emptyset \]
Is a global configuration $n_3, m_3$ from $n_0, m_0$ reachable?

Procedure $\pi_1$

Procedure $\pi_2$

Call $\pi_2$
Label Bitvectors: Example 2

\[ S(n_3) = \{ \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \} \]

Procedure \( \pi_1 \)

\[ S(n_3) \cap S(m_3) = \{ \begin{pmatrix} 1 \\ 1 \end{pmatrix} \} \]

Procedure \( \pi_2 \)

Call \( \pi_2 \)
Agenda

* Background
* Abstract Framework
  * Informal Abstractions
  * Formal Framework
  * Examples
* A General Approach
Abstract Interpretation

* $\mathcal{L} = (2^{\text{Lab}}, \subseteq, \cup, \cap, \emptyset, \text{Lab})$, a complete lattice over languages Lab

* $\mathcal{M} = (\mathcal{M}, \sqsubseteq, \sqcup, \sqcap, \bot, \top)$, an abstract lattice

* A Galois connection between $\mathcal{L}$ and $\mathcal{M}$ exists

* Goals:
  * Find $\alpha$ such that it is regular and the emptiness of the intersection is decidable in the domain $\mathcal{M}$
An idempotent semi-ring

\( \mathcal{K} = (K, \oplus, \odot, 0, 1) \)

\( \oplus \) is associative, commutative and idempotent

\( \odot \) is associative

0 is neutral element for \( \oplus \)

1 is neutral element for \( \odot \)

0 is an annihilator for \( \odot \) \( (a \odot 0 = 0) \)
* $a^0 = 1$
* $a^{n+1} = a \odot a^n$
* $a^* = \bigoplus_{n \geq 0} a^n$
* Adding $*$-Operator to $\mathcal{K}$ makes it to a Kleene Algebra
Construction of the semi-ring

- Using 0 (empty language) and 1 (ε)
- \( v_a \) represents language \( \{a\} \) of each \( a \in Lab \)
- \( \oplus \) corresponds to \( \cup \)
- \( \odot \) corresponds to concatenating
Define Abstract Lattice

* Define $\mathcal{M} = (M, \sqsubseteq, \sqcup, \sqcap, \bot, \top)$ using $\mathcal{K}$
* Order: $x \sqsubseteq y$ if $x \oplus y = y = x \cup y$
* $\bot = 0$
* $\top = \oplus$
* $\top \in K$
* $\sqcap$ corresponds to the language intersection
Galois Connection

* $\alpha: 2^{Lab^*} \rightarrow K$ and $\gamma: K \rightarrow 2^{Lab^*}$
* $\alpha(L) = \bigoplus_{a_1 \ldots a_n \in L} v_{a_1} \odot \ldots \odot v_{a_n}$
* $\gamma(x) = \{a_1, \ldots, a_n \in 2^{Lab^*} | v_{a_1} \odot \ldots \odot v_{a_n} \leq x\}$
* Properties $\alpha(\emptyset) = \bot$ and $\gamma(\emptyset) = \bot$

$\forall L_1, \ldots, L_n. \; \alpha(L_1) \cap \ldots \cap \alpha(L_n) = \bot \Rightarrow L_1 \cap \ldots \cap L_n = \emptyset$
Properties of the Abstractions

* Finite-Chain abstractions
  * Semi-lattice \((K, \oplus)\) does not contain infinite ascending chains
  * Typically \(K\) is finite

* Commutative abstractions
  * \(\odot\) is commutative
  * Forget about the order of labels
  * \(\mathcal{K}\) is a commutative Kleene algebra
Agenda

- Background
- **Abstract Framework**
  - Informal Abstractions
  - Formal Framework
  - Example
- A General Approach
Example 1: First Occurrence Ordering

* $\mathcal{W}$ is the set of words where each letter occurs once
* $\mathcal{K}$ is defined as:
  * $K = 2^\mathcal{W}$ (elements $\nu_a$ for each $a \in \text{Lab}$ where $\nu_a = \{a\}$)
  * $\preceq = \subseteq$
  * $\oplus = \cup$
  * $\odot$ concatenates different labels ($\text{abd} \odot \text{abc} = \text{abdc}$)
  * $0 = \emptyset$
  * $1 = \{\varepsilon\}$
* Abstract lattice: $\top = \mathcal{W}$ and $\sqcap = \cap$
Example 1: First Occurrence Ordering

\[ W = \{a, b, ab, ba\} \]

\[ K = 2^W \]

\[ \alpha(\{ab, b\}) = (\{a\} \circ \{b\}) \oplus \{b\} = \{ab\} \cup \{b\} = \{ab, b\} \]

\[ \alpha(L) = \bigoplus_{a_1 \ldots a_n \in L} \nu_{a_1} \circ \ldots \circ \nu_{a_n} \]
Example 1: First Occurrence Ordering

\[ W = \{a, b, c, ab, \ldots\} \]
\[ K = 2^W \]

\[ \alpha(\{ba, c^*ba\}) = (\{a\} \odot \{b\}) \oplus (\bigoplus_{n \geq 0} c^n) \odot \{b\} \odot \{a\} \]
\[ = \{ba, cba, \ldots, c^nba\} \]

\[ \alpha(L) = \bigoplus_{a_1 \ldots a_n \in L} v_{a_1} \odot \cdots \odot v_{a_n} \]
Example 1: First Occurrence Ordering

\[ \alpha(\{ab, b\}) \cap \alpha(\{ba, c^*ba\}) = \{ab, b\} \cap \{ba, cba, ..., c^nba\} = \emptyset \]
Agenda

- Background
- Abstract Framework
  - Informal Abstractions
  - Formal Framework
  - Examples
- A General Approach
Overview

1. Compute Automaton
2. Compute pre*-image
3. Label transitions
4. Extract (in)equalities
5. Solve (in)equalities
1. Compute Automaton

Automaton accepting any way to $m_3$
2. Compute pre*-image

Use saturation rule

\[ m_0, m_1, m_2, m_3 \]

\[ m_0, m_1, m_2, m_3 \]

\[ m_0, m_1, m_2, m_3 \]
3. Tagging transitions

$(m_0, e_8)$, $(m_1, e_5)$, $(m_2, e_7)$, $(m_3, e_8)$

$(m_0, e_4)$, $(m_1, e_1)$, $(m_2, e_2)$, $(m_3, e_3)$

$(m_0, e_9)$, $(m_1, e_9)$, $(m_2, e_9)$, $(m_3, e_9)$
4. Extract (in)equalities

\[
\begin{align*}
(x_4 \odot x_9) \oplus (x_8 \odot x_3) & \leq x_1 \\
v_a \odot x_3 & \leq x_2 \\
1 & \leq x_3 \\
(v_a \odot x_1) \oplus (v_b \odot x_2) & \leq x_4 \\
x_8 \odot x_6 & \leq x_5 \\
1 & \leq x_6 \\
v_a \odot x_6 & \leq x_7 \\
(v_b \odot x_7) \oplus (v_a \odot x_5) & \leq x_8 \\
1 & \leq x_9
\end{align*}
\]
5. Solve (in)equalities

* Use Gaussian elimination
* In case of a commutative Kleene algebra special solving algorithm can be used
Summary

- Use Abstract Interpretation to construct a formal framework
- Instantiate different Kleene algebra’s
- Same algorithm for different analyses (precision, ... )
Thanks for your Attention!

Any Questions?