Incremental, Inductive Coverability
(to appear at CAV 2013)

Johannes Kloos    Rupak Majumdar    Filip Niksic    Ruzica Piskac

Max Planck Institute for Software Systems

March 30, 2013, Belgrade
Abstraction Using Petri Nets

```java
void thread()
{
    // Non-critical section

    synchronized(lock)
    {
        // Critical section
    }
}
```
Abstraction Using Petri Nets

```java
void thread()
{
    // Non-critical section

    synchronized (lock)
    {
        // Critical section
    }
}
```
Abstraction Using Petri Nets

```java
void thread()
{
    //Non-critical section
    synchronized(lock)
    {
        //Critical section
    }
}
```
Abstraction Using Petri Nets

```java
void thread()
{
    //Non-critical section

    synchronized(lock)
    {
        //Critical section
    }
}
```
Abstraction Using Petri Nets

```java
void thread() {
    // Non-critical section
    synchronized (lock) {
        // Critical section
    }
}
```
Abstraction Using Petri Nets

```java
void thread()
{
    //Non-critical section
    synchronized(lock)
    {
        //Critical section
    }
}
```
Abstraction Using Petri Nets

```java
void thread()
{
    // Non-critical section
    synchronized(lock)
    {
        // Critical section
    }
}
```
Abstraction Using Petri Nets

```java
void thread()
{
    // Non-critical section

    synchronized(lock)
    {
        // Critical section
    }
}
```
Motivation

The Algorithm

Summary

Abstraction Using Petri Nets

```java
void thread()
{
    //Non-critical section

    synchronized(lock)
    {
        //Critical section
    }
}
```
void thread()
{
    //Non-critical section

    synchronized(lock)
    {
        //Critical section
    }
}
Abstraction Using Petri Nets

```java
void thread()
{
    //Non−critical section

    synchronized(lock)
    {
        //Critical section
    }
}
```
void thread()
{
    //Non−critical section

    synchronized(lock)
    {
        //Critical section
    }
}
Abstraction Using Petri Nets

```java
void thread()
{
    // Non-critical section

    synchronized (lock)
    {
        // Critical section
    }
}
```
void thread()
{
    //Non−critical section
    synchronized(lock)
    {
        //Critical section
    }
}
Can two or more threads be in the critical section?

- Can we reach \((x, y, z) \geq (2, 0, 0)\)?
- Can we cover \((2, 0, 0)\)?
Coverability Problem

- Can two or more threads be in the critical section?
- Can we reach \((x, y, z) \geq (2, 0, 0)\)?
- Can we cover \((2, 0, 0)\)?
Coverability Problem

- Can two or more threads be in the critical section?
- Can we reach \((x, y, z) \geq (2, 0, 0)\)?
- Can we cover \((2, 0, 0)\)?
Visualization in the Coordinate System
Visualization in the Coordinate System
Visualization in the Coordinate System
Visualization in the Coordinate System
Visualization in the Coordinate System

\[ y \]

\[ 0 \]

\[ 1 \]

\[ 2 \]

\[ \neg P \]

\[ y \]

\[ x \]

\[ x \]

\[ X \]
Visualization in the Coordinate System
Visualization in the Coordinate System

\[ y \]

\[ \neg P \]

\[ x \]

\[ y \]

\[ x \]
Visualization in the Coordinate System
Visualization in the Coordinate System
Visualization in the Coordinate System
Visualization in the Coordinate System
Overview of the Algorithm

- $R_k$ over-approximate states reachable in $k$ steps.
- $I \subseteq R_k$
- $R_k \subseteq R_{k+1}$
- $\text{post}(R_k) \subseteq R_{k+1}$
- $R_k \subseteq P$ for $k < N$
Overview of the Algorithm

- $R_k$ over-approximate states reachable in $k$ steps.
- $I \subseteq R_k$
- $R_k \subseteq R_{k+1}$
- $\text{post}(R_k) \subseteq R_{k+1}$
- $R_k \subseteq P$ for $k < N$
Overview of the Algorithm

- $R_k$ over-approximate states reachable in $k$ steps.
- $I \subseteq R_k$
- $R_k \subseteq R_{k+1}$
- $\text{post}(R_k) \subseteq R_{k+1}$
- $R_k \subseteq P$ for $k < N$
Overview of the Algorithm

- $R_k$ over-approximate states reachable in $k$ steps.
- $I \subseteq R_k$
- $R_k \subseteq R_{k+1}$
- $\text{post}(R_k) \subseteq R_{k+1}$
- $R_k \subseteq P$ for $k < N$
Overview of the Algorithm

- $R_k$ over-approximate states reachable in $k$ steps.
- $I \subseteq R_k$
- $R_k \subseteq R_{k+1}$
- $\text{post}(R_k) \subseteq R_{k+1}$
- $R_k \subseteq P$ for $k < N$
Proving the Uncoverability

\[ \neg P \]

\( R_0 \)
Proving the Uncoverability

Motivation

The Algorithm

Summary
Proving the Uncoverability

\[ R_0 \]

\[ R_1 \]

\[ \neg P \]
Proving the Uncoverability

$R_0$

$R_1$
Proving the Uncoverability

$R_0$

$R_1$

$R_2$
Proving the Uncoverability

$R_0$

$R_1$

$R_2$
Proving the Uncoverability

\[ R_0 \]

\[ R_1 \]

\[ R_2 \]
Proving the Uncoverability

$R_0$

$R_1$

$R_2$
Proving the Uncoverability
Proving the Uncoverability

\[ \begin{align*}
R_0 & : (1, 1) \\
R_1 & : (1, 1), (2, 1) \\
R_2 & : (0, 1), (1, 1), (2, 1) \\
R_3 & : (0, 1), (1, 1), (2, 1), (2, 2), \neg P
\end{align*} \]
Proving the Uncoverability

Motivation

The Algorithm

Summary

Proving the Uncoverability

\[ R_0 \]

\[ R_1 \]

\[ R_2 \]

\[ R_3 \]

F. Niksic

Incremental, Inductive Coverability
Proving the Uncoverability

$R_0$

$R_1$

$R_2$

$R_3$

$\neg P$
Proving the Uncoverability

$R_0$

$R_1$

$R_2$

$R_3$

$\neg P$
Proving the Uncoverability
Proving the Uncoverability
Proving the Uncoverability
Proving the Uncoverability

Motivation

Proving the Uncoverability

The Algorithm

Proving the Uncoverability

Summary

Motivation

Proving the Uncoverability

The Algorithm

Proving the Uncoverability

Summary

Motivation

Proving the Uncoverability

The Algorithm

Proving the Uncoverability

Summary

Motivation

Proving the Uncoverability

The Algorithm

Proving the Uncoverability

Summary

Motivation

Proving the Uncoverability

The Algorithm

Proving the Uncoverability

Summary

Motivation

Proving the Uncoverability

The Algorithm

Proving the Uncoverability

Summary

Motivation

Proving the Uncoverability

The Algorithm

Proving the Uncoverability

Summary

Motivation

Proving the Uncoverability

The Algorithm

Proving the Uncoverability

Summary

Motivation

Proving the Uncoverability

The Algorithm

Proving the Uncoverability

Summary

Motivation

Proving the Uncoverability

The Algorithm

Proving the Uncoverability

Summary
Proving the Uncoverability
Proving the Uncoverability

Motivation

The Algorithm

Summary

F. Niksic

Incremental, Inductive Coverability
Proving the Uncoverability
Proving the Uncoverability
Proving the Uncoverability

$R_0$

$R_1$

$R_2$

$R_3$

$R_4$
Proving the Uncoverability
Proving the Uncoverability

\[ R_0 \]
\[ R_1 \]
\[ R_2 \]
\[ R_3 \]
\[ R_4 \]
Proving the Uncoverability

\[ R_0 \]

\[ R_1 \]

\[ R_2 \]

\[ R_3 \]

\[ R_4 \]

\[ \neg P \]
Inductive Invariant

During the execution:
- \( I \subseteq R_k \)
- \( R_k \subseteq R_{k+1} \)
- \( \text{post}(R_k) \subseteq R_{k+1} \)
- \( R_k \subseteq P \) for \( k < N \)

If \( R_i = R_{i+1} \):
- \( I \subseteq R_i \)
- \( \text{post}(R_i) \subseteq R_i \)
  \( \Rightarrow R_i \) is inductive invariant
- \( R_i \subseteq P \)
  \( \Rightarrow P \) is invariant.
Inductive Invariant

During the execution:
- $I \subseteq R_k$
- $R_k \subseteq R_{k+1}$
- $\text{post}(R_k) \subseteq R_{k+1}$
- $R_k \subseteq P$ for $k < N$

If $R_i = R_{i+1}$:
- $I \subseteq R_i$
- $\text{post}(R_i) \subseteq R_i$
  $\Rightarrow R_i$ is inductive invariant
- $R_i \subseteq P$
  $\Rightarrow P$ is invariant.
Inductive Invariant

During the execution:
- \( I \subseteq R_k \)
- \( R_k \subseteq R_{k+1} \)
- \( \text{post}(R_k) \subseteq R_{k+1} \)
- \( R_k \subseteq P \) for \( k < N \)

If \( R_i = R_{i+1} \):
- \( I \subseteq R_i \)
- \( \text{post}(R_i) \subseteq R_i \)
  \( \Rightarrow R_i \) is inductive invariant
- \( R_i \subseteq P \)
  \( \Rightarrow P \) is invariant.
Inductive Invariant

During the execution:

- \( I \subseteq R_k \)
- \( R_k \subseteq R_{k+1} \)
- \( \text{post}(R_k) \subseteq R_{k+1} \)
- \( R_k \subseteq P \) for \( k < N \)

If \( R_i = R_{i+1} \):

- \( I \subseteq R_i \)
- \( \text{post}(R_i) \subseteq R_i \)
  \(\Rightarrow\) \( R_i \) is inductive invariant
- \( R_i \subseteq P \)
  \(\Rightarrow\) \( P \) is invariant.
Inductive Invariant

During the execution:
- $I \subseteq R_k$
- $R_k \subseteq R_{k+1}$
- $\text{post}(R_k) \subseteq R_{k+1}$
- $R_k \subseteq P$ for $k < N$

If $R_i = R_{i+1}$:
- $I \subseteq R_i$
- $\text{post}(R_i) \subseteq R_i$
  $\Rightarrow R_i$ is inductive invariant
- $R_i \subseteq P$
  $\Rightarrow P$ is invariant.
Inductive Invariant

During the execution:
- $I \subseteq R_k$
- $R_k \subseteq R_{k+1}$
- $\text{post}(R_k) \subseteq R_{k+1}$
- $R_k \subseteq P$ for $k < N$

If $R_i = R_{i+1}$:
- $I \subseteq R_i$
- $\text{post}(R_i) \subseteq R_i$
  $\Rightarrow$ $R_i$ is inductive invariant
- $R_i \subseteq P$
  $\Rightarrow$ $P$ is invariant.
Inductive Invariant

During the execution:
- $I \subseteq R_k$
- $R_k \subseteq R_{k+1}$
- $\text{post}(R_k) \subseteq R_{k+1}$
- $R_k \subseteq P$ for $k < N$

If $R_i = R_{i+1}$:
- $I \subseteq R_i$
- $\text{post}(R_i) \subseteq R_i$
  $\Rightarrow R_i$ is inductive invariant
- $R_i \subseteq P$
  $\Rightarrow P$ is invariant.
Incremental, Inductive Coverability

- Generalizes Aaron Bradley’s IC3 to well-structured transition systems (which include Petri nets).
- Terminates for downward-finite WSTS.
- Efficient implementation for Petri nets.
Incremental, Inductive Coverability

- Generalizes Aaron Bradley’s IC3 to well-structured transition systems (which include Petri nets).
- Terminates for downward-finite WSTS.
- Efficient implementation for Petri nets.
Incremental, Inductive Coverability

- Generalizes Aaron Bradley’s IC3 to well-structured transition systems (which include Petri nets).
- Terminates for downward-finite WSTS.
- Efficient implementation for Petri nets.