Checking Unwinding Conditions for Finite State Systems

Deepak D’Souza, Raghavendra K.R.

Indian Institute of Science, Bangalore, India

MAKS Framework of Heiko

Events. Visible, Confidential, Neither

MAKS Framework of Heiko

Events. Visible, Confidential, Neither

Trace: finite sequence of events

MAKS Framework of Heiko

Events. Visible, Confidential, Neither

Trace: finite sequence of events

System: A set of traces

MAKS Framework of Heiko

Events. Visible, Confidential, Neither

Trace: finite sequence of events

System: A set of traces

Information flow properties

for all x in L with some conditions $\Rightarrow$

there exists y in L with some conditions

MAKS Framework of Heiko

Events. Visible, Confidential, Neither

Trace: finite sequence of events

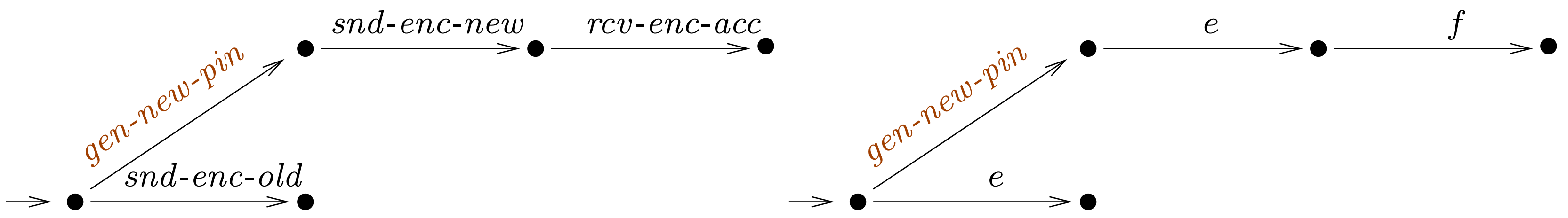
System: A set of traces

Information flow properties
for all x in L with some conditions $\Rightarrow$
there exists y in L with some conditions

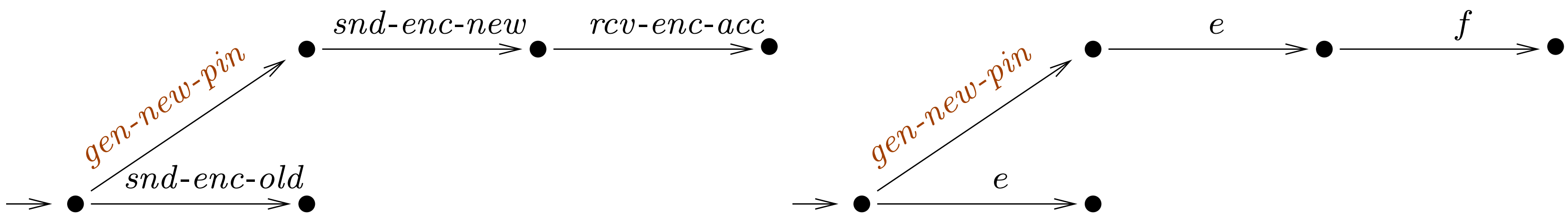
Non-Inference(NF)

$\forall \tau \in L \Rightarrow \exists \tau' \in L \;\; \tau' = \tau \upharpoonright_V$

An Example (1)



An Example (1)

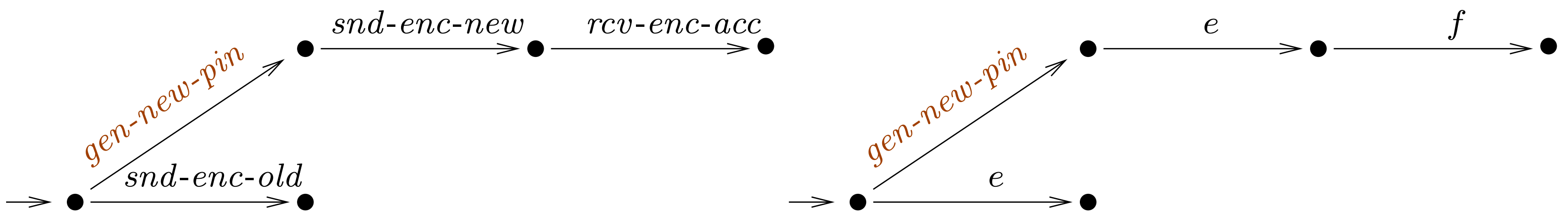


$V = \{e, f\}$

$C = \{gen\text{-}new\text{-}pin\}$

$N = \phi$

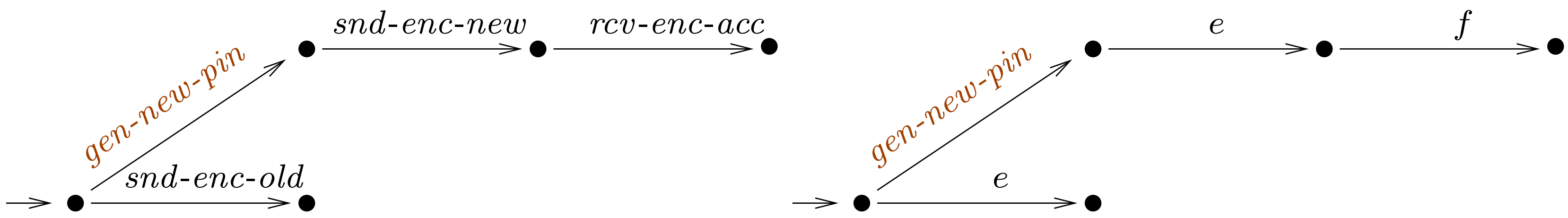
An Example (1)



$V = \{e, f\}$
$C = \{gen\text{-}new\text{-}pin\}$
$N = \phi$

Tr = { $gen\text{-}new\text{-}pin\ e\ f$,
e } + prefixes

An Example (1)

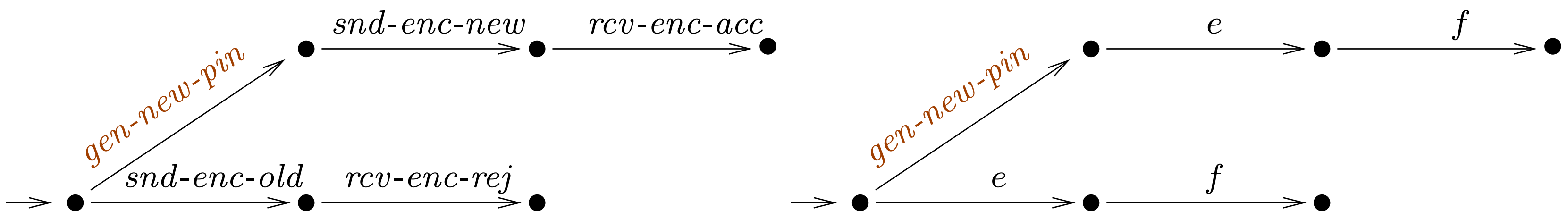


$V = \{e, f\}$
$C = \{gen\text{-}new\text{-}pin\}$
$N = \phi$

Tr = { $gen\text{-}new\text{-}pin\ e\ f$,
e } + prefixes

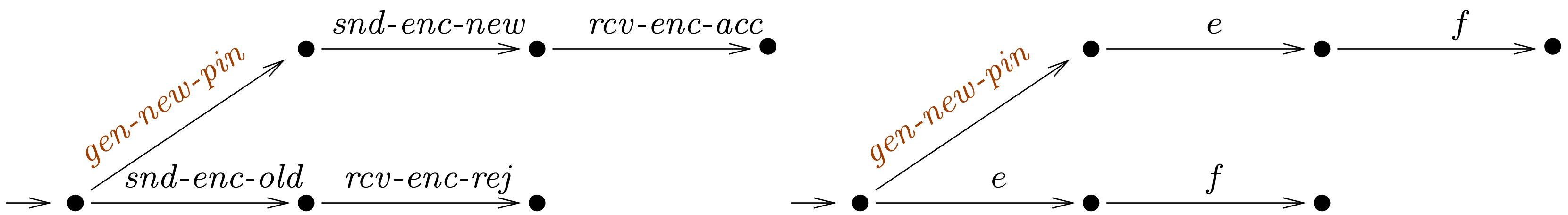
Confidentiality compromised. Noninference fails

An Example (2)



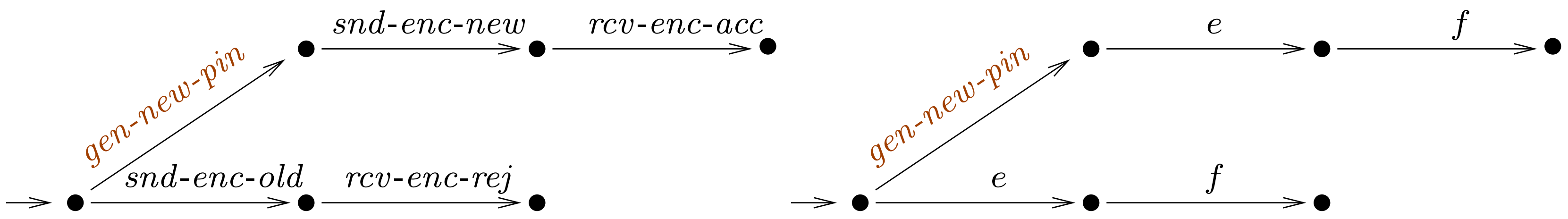
gen-new-pin
snd-enc-new
rcv-enc-acc
snd-enc-old
rcv-enc-rej
gen-new-pin
e
f
e
f

An Example (2)



$V = \{e, f\}$
$C = \{gen\text{-}new\text{-}pin\}$
$N = \phi$

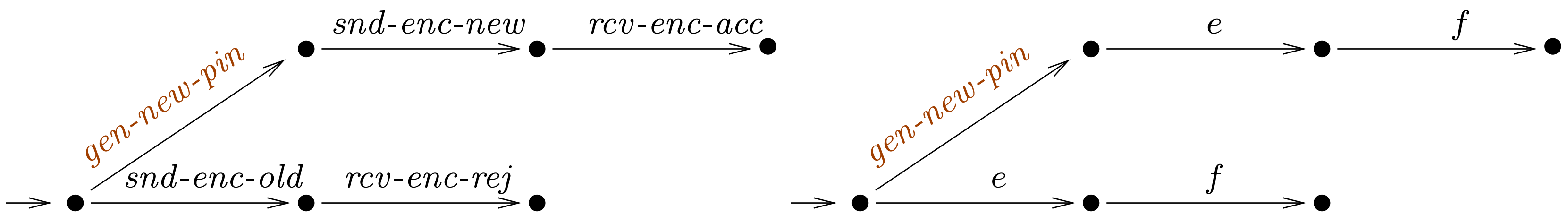
An Example (2)



$V = \{e, f\}$
$C = \{gen\text{-}new\text{-}pin\}$
$N = \phi$

Tr = { $gen\text{-}new\text{-}pin\ e\ f$,
$e\ f$ } + prefixes

An Example (2)



$V = \{e, f\}$

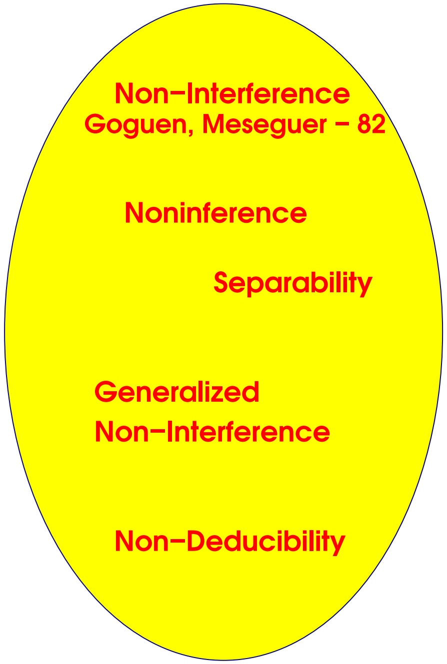
$C = \{gen\text{-}new\text{-}pin\}$

$N = \phi$

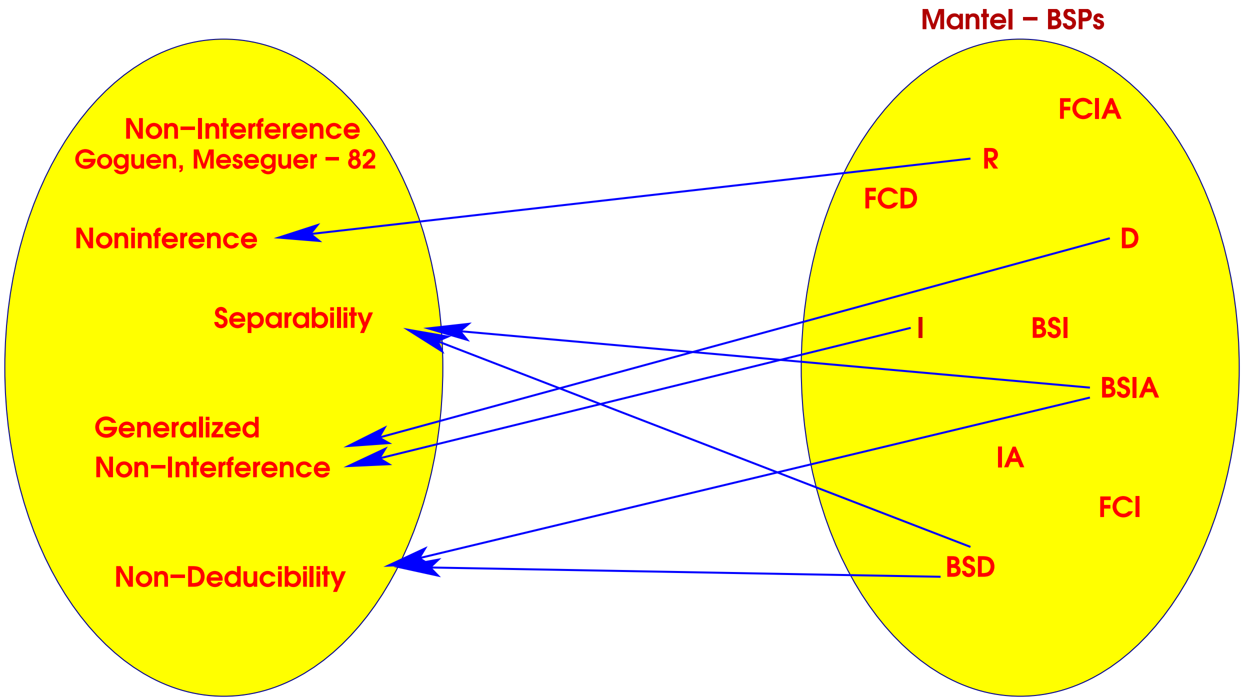
Tr = { $gen\text{-}new\text{-}pin\ e\ f$,
$e\ f$ } + prefixes

Confidentiality maintained. Noninference holds

Information Flow Properties



Information Flow Properties



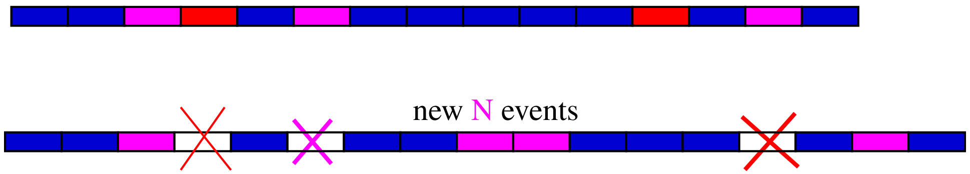
Basic Security Predicates (BSPs)

Trace based information flow properties in BSPs

Basic Security Predicates (BSPs)

Trace based information flow properties in BSPs

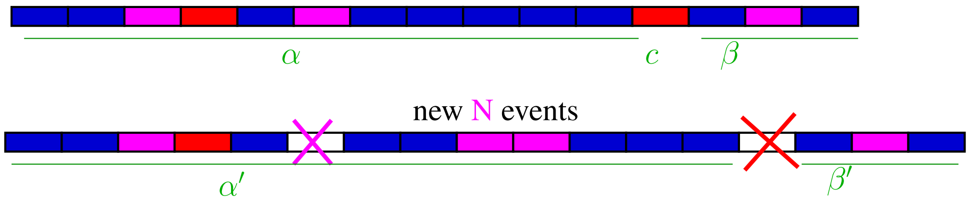
BSP Removal (R)



Basic Security Predicates (BSPs)

Trace based information flow properties in BSPs

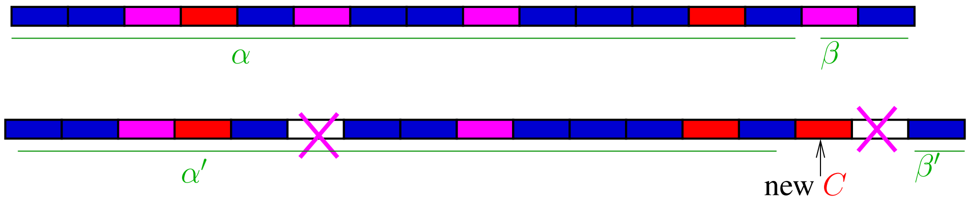
BSP Deletion (D)



Basic Security Predicates (BSPs)

Trace based information flow properties in BSPs

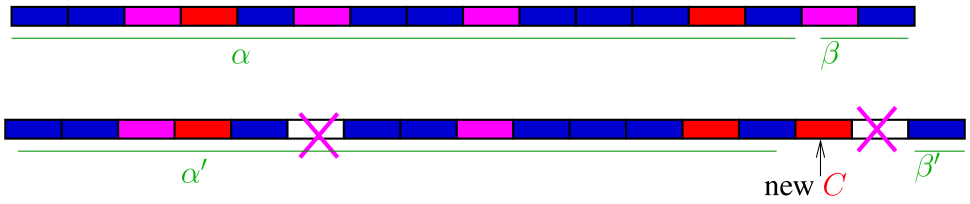
BSP Insertion (I)



Basic Security Predicates (BSPs)

Trace based information flow properties in BSPs

BSP Insertion (I)



Generalized Non-Interference - I and D

Noninference - R

Verification using Model Checking

Properties of sets of traces, Classical Model Checking techniques (Temporal Logic etc) cannot be used

Verification using Model Checking

Properties of sets of traces, Classical Model Checking techniques (Temporal Logic etc) cannot be used

{DRS05} Sound and Complete Model Checking method for Finite State Systems

Verification using Model Checking

Properties of sets of traces, Classical Model Checking techniques (Temporal Logic etc) cannot be used

{DRS05} Sound and Complete Model Checking method for Finite State Systems

L satisfies a BSP P is reduced to $op_1(L) \subseteq op_2(L)$

Verification using Model Checking

Properties of sets of traces, Classical Model Checking techniques (Temporal Logic etc) cannot be used

{DRS05} Sound and Complete Model Checking method for Finite State Systems

L satisfies a BSP P is reduced to $op_1(L) \subseteq op_2(L)$

Examples

- L satisfies Removal R iff $L\upharpoonright_V \subseteq_N L$.
- L satisfies Deletion D iff *l-del*$(L) \subseteq_N L$.

Verification using Model Checking

Properties of sets of traces, Classical Model Checking techniques (Temporal Logic etc) cannot be used

{DRS05} Sound and Complete Model Checking method for Finite State Systems

L satisfies a BSP P is reduced to $op_1(L) \subseteq op_2(L)$

Examples

- L satisfies Removal R iff $L\restriction_V \subseteq_N L$.
- L satisfies Deletion D iff *l-del*$(L) \subseteq_N L$.

Regularity Preserving: Algorithm to construct automata for $op(L)$, given an automata for L

Verification using Model Checking

Properties of sets of traces, Classical Model Checking techniques (Temporal Logic etc) cannot be used

{DRS05} Sound and Complete Model Checking method for Finite State Systems

L satisfies a BSP P is reduced to $op_1(L) \subseteq op_2(L)$

Examples

- L satisfies Removal R iff $L \upharpoonright_V \subseteq_N L$.
- L satisfies Deletion D iff *l-del*$(L) \subseteq_N L$.

Regularity Preserving: Algorithm to construct automata for $op(L)$, given an automata for L

Running time: Exponential in the size of the system

Unwinding - Definitions

Σ-labelled transition system $\mathcal{T} = (Q, s, \longrightarrow)$

Unwinding - Definitions

Σ-labelled transition system $\mathcal{T} = (Q, s, \longrightarrow)$

Unwinding relation $\ltimes$: a binary relation on Q satisfying osc

Unwinding - Definitions

Σ-labelled transition system $\mathcal{T} = (Q, s, \longrightarrow)$

Unwinding relation $\ltimes$: a binary relation on Q satisfying osc

osc

$$\begin{tikzcd} p \arrow[r, "e"] \arrow[d, dash, "\ltimes"'] & q \\ r & \end{tikzcd}$$

Unwinding - Definitions

Σ-labelled transition system $\mathcal{T} = (Q, s, \longrightarrow)$

Unwinding relation $\ltimes$: a binary relation on Q satisfying osc

osc

$$\begin{tikzcd} p \arrow[r, "e"] \arrow[d, dash, "\ltimes"'] & q \arrow[d, dash, "\ltimes"] \\ r \arrow[r, "\delta"] & t \end{tikzcd}$$

Unwinding - Definitions

Σ-labelled transition system $\mathcal{T} = (Q, s, \longrightarrow)$

Unwinding relation $\ltimes$: a binary relation on Q satisfying osc

$\mathcal{T}$ satisfies unwinding condition lrf w.r.t. $\ltimes$

Unwinding - Definitions

Σ-labelled transition system $\mathcal{T} = (Q, s, \longrightarrow)$

Unwinding relation $\ltimes$: a binary relation on Q satisfying osc

$\mathcal{T}$ satisfies unwinding condition lrf w.r.t. $\ltimes$

$$\mathrm{p} \xrightarrow{\quad c \quad} \mathrm{q}$$

Unwinding - Definitions

Σ-labelled transition system $\mathcal{T} = (Q, s, \longrightarrow)$

Unwinding relation $\ltimes$: a binary relation on Q satisfying osc

$\mathcal{T}$ satisfies unwinding condition lrf w.r.t. $\ltimes$

$$
\begin{tikzcd}
\mathrm{p} \arrow[r, "c"] \arrow[r, dash, bend right, "\rtimes"'] & \mathrm{q}
\end{tikzcd}
$$

Unwinding - Definitions

Σ-labelled transition system $\mathcal{T} = (Q, s, \longrightarrow)$

Unwinding relation $\ltimes$: a binary relation on Q satisfying osc

$\mathcal{T}$ satisfies unwinding condition lrf w.r.t. $\ltimes$

$$\begin{tikzcd} p \arrow[r, "c"] \arrow[r, no head, bend right, "\rtimes"'] & q \end{tikzcd}$$

$\mathcal{T}$ satisfies unwinding condition lrb w.r.t. $\ltimes$

Unwinding - Definitions

Σ-labelled transition system $\mathcal{T} = (Q, s, \longrightarrow)$

Unwinding relation $\ltimes$: a binary relation on Q satisfying osc

$\mathcal{T}$ satisfies unwinding condition lrf w.r.t. $\ltimes$

$$\begin{tikzcd} \mathrm{p} \arrow[r, "c"] \arrow[r, bend right, dash, "\rtimes"'] & \mathrm{q} \end{tikzcd}$$

$\mathcal{T}$ satisfies unwinding condition lrb w.r.t. $\ltimes$

p

Unwinding - Definitions

Σ-labelled transition system $\mathcal{T} = (Q, s, \longrightarrow)$

Unwinding relation $\ltimes$: a binary relation on Q satisfying osc

$\mathcal{T}$ satisfies unwinding condition lrf w.r.t. $\ltimes$

$$\begin{tikzcd} p \arrow[r, "c"] \arrow[r, bend right, dash, "\rtimes"'] & q \end{tikzcd}$$

$\mathcal{T}$ satisfies unwinding condition lrb w.r.t. $\ltimes$

$$\begin{tikzcd} p \arrow[r, "c"] \arrow[r, bend right, dash, "\ltimes"'] & q \end{tikzcd}$$

Unwinding - Definitions

Σ-labelled transition system $\mathcal{T} = (Q, s, \longrightarrow)$

Unwinding relation $\ltimes$: a binary relation on Q satisfying osc

$\mathcal{T}$ satisfies unwinding condition lrf w.r.t. $\ltimes$

$$\begin{tikzcd} p \arrow[r, "c"] \arrow[r, bend right, dash, "\rtimes"'] & q \end{tikzcd}$$

$\mathcal{T}$ satisfies unwinding condition lrb w.r.t. $\ltimes$

$$\begin{tikzcd} p \arrow[r, "c"] \arrow[r, bend right, dash, "\ltimes"'] & q \end{tikzcd}$$

$fcrf, fcrb, lrbe, fcrbe$

Verification using Unwinding

Heiko's results

$\mathcal{T}$ satisfies

- BSD if there exists an unwinding relation $\ltimes$ such that $\mathcal{T}$ satisfies lrf w.r.t. $\ltimes$
- BSI if ... $\mathcal{T}$ satisfies lrb w.r.t. $\ltimes$
- $BSIA$ if ... $\mathcal{T}$ satisfies $lrbe$ w.r.t. $\ltimes$
- FCD if ... $\mathcal{T}$ satisfies $fcrf$ w.r.t. $\ltimes$
- FCI if ... $\mathcal{T}$ satisfies $fcrb$ w.r.t. $\ltimes$
- $FCIA$ if ... $\mathcal{T}$ satisfies $fcrbe$ w.r.t. $\ltimes$

Verification using Unwinding

Heiko's results

$\mathcal{T}$ satisfies

- BSD if there exists an unwinding relation $\ltimes$ such that $\mathcal{T}$ satisfies lrf w.r.t. $\ltimes$
- BSI if ... $\mathcal{T}$ satisfies lrb w.r.t. $\ltimes$
- $BSIA$ if ... $\mathcal{T}$ satisfies $lrbe$ w.r.t. $\ltimes$
- FCD if ... $\mathcal{T}$ satisfies $fcrf$ w.r.t. $\ltimes$
- FCI if ... $\mathcal{T}$ satisfies $fcrb$ w.r.t. $\ltimes$
- $FCIA$ if ... $\mathcal{T}$ satisfies $fcrbe$ w.r.t. $\ltimes$

Sufficient Conditions

Verification using Unwinding

Heiko's results

$\mathcal{T}$ satisfies

- BSD if there exists an unwinding relation $\ltimes$ such that $\mathcal{T}$ satisfies lrf w.r.t. $\ltimes$
- BSI if ... $\mathcal{T}$ satisfies lrb w.r.t. $\ltimes$
- $BSIA$ if ... $\mathcal{T}$ satisfies $lrbe$ w.r.t. $\ltimes$
- FCD if ... $\mathcal{T}$ satisfies $fcrf$ w.r.t. $\ltimes$
- FCI if ... $\mathcal{T}$ satisfies $fcrb$ w.r.t. $\ltimes$
- $FCIA$ if ... $\mathcal{T}$ satisfies $fcrbe$ w.r.t. $\ltimes$

Sufficient Conditions

Finitely many relations if finite states

Observations on Unwinding

Unwinding relations are closed under union.

Hence, maximal unwinding relation $\ltimes_{\mathcal{T}}$ exists

Observations on Unwinding

Unwinding relations are closed under union.
Hence, maximal unwinding relation $\ltimes_{\mathcal{T}}$ exists

Unwinding Conditions(lrf, ...) are upward closed i.e,
Let $\ltimes_1 \subseteq \ltimes_2$. If $\ltimes_1$ satisfies lrf, then so does $\ltimes_2$

Observations on Unwinding

Unwinding relations are closed under union.
Hence, maximal unwinding relation $\ltimes_{\mathcal{T}}$ exists

Unwinding Conditions(lrf, ...) are upward closed i.e,
Let $\ltimes_1 \subseteq \ltimes_2$. If $\ltimes_1$ satisfies lrf, then so does $\ltimes_2$

$\mathcal{T}$ satisfies lrf w.r.t. some $\ltimes$ iff $\mathcal{T}$ satisfies lrf w.r.t. $\ltimes_{\mathcal{T}}$

Observations on Unwinding

Unwinding relations are closed under union.
Hence, maximal unwinding relation $\ltimes_{\mathcal{T}}$ exists

Unwinding Conditions(lrf, ...) are upward closed i.e,
Let $\ltimes_1 \subseteq \ltimes_2$. If $\ltimes_1$ satisfies lrf, then so does $\ltimes_2$

$\mathcal{T}$ satisfies lrf w.r.t. some $\ltimes$ iff $\mathcal{T}$ satisfies lrf w.r.t. $\ltimes_{\mathcal{T}}$

Similarly for lrb,...

Simulation Relation

Simulation Relation

$$\begin{tikzcd} p \arrow[r, "e"] \arrow[d, dash, "\curlywedge"'] & r \\ q & \end{tikzcd}$$

Simulation Relation

Simulation Relation

$$
\begin{tikzcd}
p \arrow[r, "e"] \arrow[d, dash, "\curlywedge"'] & r \arrow[d, dash, "\curlywedge"] \\
q \arrow[r, "e"] & t
\end{tikzcd}
$$

Simulation Relation

Simulation Relation

$$
\begin{tikzcd}
p \arrow[r, "e"] \arrow[d, dash, "\curlywedge"'] & r \arrow[d, dash, "\curlywedge"] \\
q \arrow[r, "e"] & t
\end{tikzcd}
$$

Maximal simulation relation exists.

Well known Algorithms: {HHK}

Simulation Relation

Simulation Relation

$$\begin{tikzcd} p \arrow[r, "e"] \arrow[d, dash, "\curlywedge"'] & r \arrow[d, dash, "\curlywedge"] \\ q \arrow[r, "e"] & t \end{tikzcd}$$

Maximal simulation relation exists.

Well known Algorithms: {HHK}

Naive Algorithm: Computing Maximal Simulation Relation

Input: $\mathcal{T}$, a finite state LTS

Output: $\prec_{\mathcal{T}}$, the maximal simulation relation for $\mathcal{T}$

for $p \in Q$

$\quad sim(p) = \{q \in Q \mid$ for all e enabled at p, e is also enabled at $q\}$

while there are states p, q, r and $e \in \Sigma$ such that

$r \in post_e(p),\ q \in sim(p)$ and $post_e(q) \cap sim(r) = \phi$ {

$\quad sim(p) = sim(p) \setminus \{q\}$

}

$\prec_{\mathcal{T}} = \bigcup_{q \in Q}\{\{q\} \times sim(q)\}$

Unwinding as a Simulation Relation (1)

Theorem $\ltimes_{\mathcal{T}}$ for $\mathcal{T}$ coincides with the maximal simulation relation $\prec_{\mathcal{T}_V}$ for $\mathcal{T}_V$

Unwinding as a Simulation Relation (1)

Theorem $\ltimes_{\mathcal{T}}$ for $\mathcal{T}$ coincides with the maximal simulation relation $\prec_{\mathcal{T}_V}$ for $\mathcal{T}_V$

Construct $\mathcal{T}_V$ from $\mathcal{T}$ by

1. deleting C edges
2. replacing N edges with ϵ edges
3. compute transitive closure (Warshall's Algorithm - $O(n^3)$)

Unwinding as a Simulation Relation (1)

Theorem $\ltimes_{\mathcal{T}}$ for $\mathcal{T}$ coincides with the maximal simulation relation $\prec_{\mathcal{T}_V}$ for $\mathcal{T}_V$

Construct $\mathcal{T}_V$ from $\mathcal{T}$ by

1. deleting C edges
2. replacing N edges with ϵ edges
3. compute transitive closure (Warshall's Algorithm - $O(n^3)$)

States are same

Unwinding as a Simulation Relation (1)

Theorem $\ltimes_{\mathcal{T}}$ for $\mathcal{T}$ coincides with the maximal simulation relation $\prec_{\mathcal{T}_V}$ for $\mathcal{T}_V$

Construct $\mathcal{T}_V$ from $\mathcal{T}$ by

1. deleting C edges
2. replacing N edges with ϵ edges
3. compute transitive closure (Warshall's Algorithm - $O(n^3)$)

States are same

Proof follows due to the construction of $\mathcal{T}_V$

Unwinding as a Simulation Relation (2)

Checking Unwinding Conditions

1. Construct $\mathcal{T}_V$ from $\mathcal{T}$
2. Compute the maximal simulation relation $\prec_{\mathcal{T}_V}$ using standard algorithms
3. Check the unwinding conditions (lrf, ...) w.r.t. $\prec_{\mathcal{T}_V}$

Unwinding as a Simulation Relation (2)

Checking Unwinding Conditions

1. Construct $\mathcal{T}_V$ from $\mathcal{T}$
2. Compute the maximal simulation relation $\prec_{\mathcal{T}_V}$ using standard algorithms
3. Check the unwinding conditions ($lrf, \ldots$) w.r.t. $\prec_{\mathcal{T}_V}$

Feasible way to check, though not complete

Thank You