Slicing of Extended Finite State Machines

Kelly Androutsopoulos\textsuperscript{1}, David Clark\textsuperscript{1}, Nicolas Gold\textsuperscript{1},
Mark Harman\textsuperscript{1}, Rob Hierons\textsuperscript{2}, Zheng Li\textsuperscript{1} and Laurence Tratt\textsuperscript{3}

\textsuperscript{1}CREST, King’s College London
\textsuperscript{2}Brunel University
\textsuperscript{3}Bournemouth University
Dependence Based Slicing
- Dependence Analysis
- Marked Transitions
- $\varepsilon$-elimination
- Minimisation

Event Restriction Slicing
- BasicSlice
- Constant Propagation
- R-mergeEquivalentStates
- G-mergeEquivalentStates
which other lines affect the selected line?

we only care about this line
Program Slicing

which other lines affect the selected line?

we only care about this line
which other lines affect the selected line?

we only care about this line
Program Slicing
Program Slicing

Program comprehension
Program Slicing

- Program comprehension
- Impact analysis

Reverse engineering

Testing and debugging

Refactoring

Slicing of EFSMs  Zheng Li

CREST, KING’S COLLEGE LONDON
Program Slicing

- Program comprehension
- Dependence cluster analysis
- Impact analysis
- Program slicing
Program Slicing

Testing and debugging

Program comprehension

Dependence cluster analysis

Impact analysis
Can slicing be applied to model level?
If the model like this?
Motivation

- Models tend to be larger and more complex.
Motivation

- Models tend to be larger and more complex.
- Slicing has provided a valuable suite of maintenance techniques at the implementation level, but little at model level.
An Extended Finite State Machine (EFSM) $M$ is a tuple $(S, T, E, V)$ where $S$ is a set of states, $T$ is a set of transitions, $E$ is a set of events, and $V$ is a store represented by a set of variables. Transitions have a source state $\text{source}(t) \in S$, a target state $\text{target}(t) \in S$ and a label $\text{lbl}(t)$. Transition labels are of the form $e_1[c]/a$ where $e_1 \in E$, $c$ is a condition and $a$ a sequence of actions.
An EFSM example: DoorControl

- **Start**
  - T1: setTimer/timer:=5

- **wait**
  - T2: waitTimer/timer:=timer - 1
  - T3: ready/timer:=0

- **closing**
  - T4: closing
  - T5: buttonInterrupt/timer:=3
  - T6: fullyClosed
  - T7: closeTimer

- **opened**
  - T8: open/timer:=10

- **opening**
  - T9: opening

- **closed**
  - T10: fullyOpened

T11: openTimer/timer:=timer - 1
Definition (Slicing Criterion)
A slicing criterion for an EFSM is a pair \((t, V)\) where transition \(t \in T\) and variable set \(V \subseteq \text{Var}\). It designates the point in the evaluation immediately after the execution of the action contained in transition \(t\).

Definition (Slice)
An EFSM slice \(M'\) is a reduced machine, where for all inputs \(i\) it contains at least one execution where the value of \(v \in V\) at \(t\) is equal to the value of \(v\) at \(t\) in the original EFSM \(M\).
The CREST EFSM slicing tool

INPUT
- EFSM model M
- Slicing criterion \((t, V)\)
- Choice between NTSCD or NTICD

DEPENDENCE ANALYSIS
- Control Dependence
- Data Dependence

SLICING
- Dependence Graph
- Mark Transitions in M
- Minimisation

OUTPUT
- EFSM M' without epsilon-transitions

Marked EFSM with epsilon-transitions

epsilon-elimination
The CREST EFSM slicing tool

INPUT
- EFSM model M
- Slicing criterion (t, V)
- Choice between NTSCD or NTICD

DEPENDENCE ANALYSIS
- Control Dependence
- Data Dependence

SLICING
- Dependence Graph
- Mark Transitions in M
- Marked EFSM with epsilon-transitions
- epsilon-elimination
- Minimisation

OUTPUT
- EFSM M' without epsilon-transitions
EFSM VS CFG

- **Transition in EFSM VS Node in CFG**
  - Difference
  - 1
  - 2
  - 3

**Slicing of EFSMs**

Zheng Li
CREST, KING’S COLLEGE LONDON
Difference

1. **Transition in EFSM VS Node in CFG**
### Difference

1. **Transition in EFSM VS Node in CFG**
2. **Self-looping edge and multi-edges between two nodes**
**EFSM VS CFG**

**Difference**

1. **Transition in EFSM VS Node in CFG**
2. Self-looping edge and multi-edges between two nodes
3. Non-termination (Exit node)

---

**Slicing of EFSMs**

Zheng Li

CREST, KING’S COLLEGE LONDON
Dependence Analysis of EFSM

- Data Dependence
- Control Dependence
Dependence Analysis of EFSM

- Data Dependence

- Control Dependence
  - Traditional Control Dependence [Korel et al, ICSM 2003]
Dependence Analysis of EFSM

- Data Dependence

- Control Dependence
  - Traditional Control Dependence [Korel et al, ICSM 2003]
  - Non-Termination Insensitive Control Dependence (NTICD) [Ranganath et al. ESOP 2005]
  - Non-Termination Sensitive Control Dependence (NTSCD) [Ranganath et al. ESOP 2005]
Dependence Analysis of EFSM

- Data Dependence

- Control Dependence
  - Traditional Control Dependence [Korel et al, ICSM 2003]
  - Non-Termination Insensitive Control Dependence (NTICD) [Ranganath et al. ESOP 2005]
  - Non-Termination Sensitive Control Dependence (NTSCD) [Ranganath et al. ESOP 2005]
  - Unfair Non-Termination Insensitive Control Dependence (UNTICD) [Androutsopoulos et al. FASE 2009]
Data Dependence

**Definition**

\[ T_i \xrightarrow{\text{DD}} T_j \]

means that transitions \( T_i \) and \( T_j \) are data dependent with respect to a variable \( v \) if:

1. \( v \in D(T_i) \), where \( D(T_i) \) is a set of variables defined by transition \( T_i \), i.e. variables defined by actions and by the event of \( T_i \);
Data Dependence

Definition

\( T_i \xrightarrow{\text{DD}} T_j \) means that transitions \( T_i \) and \( T_j \) are data dependent with respect to a variable \( v \) if:

1. \( v \in D(T_i) \), where \( D(T_i) \) is a set of variables defined by transition \( T_i \), i.e. variables defined by actions and by the event of \( T_i \);
2. \( v \in U(T_j) \), where \( U(T_j) \) is a set of variables used in a condition and actions of transition \( T_j \);

\( T_i : \text{def}(v) \) \hspace{1cm} \( T_j : \text{use}(v) \)
Data Dependence

Definition

$T_i \xrightarrow{DD} T_j$ means that transitions $T_i$ and $T_j$ are data dependent with respect to a variable $v$ if:

1. $v \in D(T_i)$, where $D(T_i)$ is a set of variables defined by transition $T_i$, i.e. variables defined by actions and by the event of $T_i$;
2. $v \in U(T_j)$, where $U(T_j)$ is a set of variables used in a condition and actions of transition $T_j$;
3. there exists a path in an EFSM from the source($T_i$) to the target($T_j$) whereby $v$ is not modified by any of the intermediate transitions.

\[ T_i: \text{def}(v) \xrightarrow{\text{NO \ def}(v)} T_j: \text{use}(v) \]
### Types of Control Dependence and Paths in EFSM

<table>
<thead>
<tr>
<th>Name</th>
<th>Path type</th>
</tr>
</thead>
<tbody>
<tr>
<td>NTSCD</td>
<td>Maximal Path</td>
</tr>
<tr>
<td>NTICD</td>
<td>Sink-bounded Path</td>
</tr>
<tr>
<td>UNTICD</td>
<td>Unfair Sink-bounded Path</td>
</tr>
</tbody>
</table>
Definition (Maximal Path)

A maximal path is any path that terminates in a final transition, or is infinite.
Definition (Maximal Path)

A **maximal path** is any path that terminates in a final transition, or is infinite.
A maximal path is any path that terminates in a final transition, or is infinite.
A **maximal path** is any path that terminates in a final transition, or is infinite.
A **control sink** in an EFSM is a set of transitions $\mathcal{K}$ that form a strongly connected component (SCC) such that, for each transition $t$ in $\mathcal{K}$ each successor of $t$ is also in $\mathcal{K}$.

**Diagram:**

- **Start**
- **wait**
  - $T1$: setTimer/timer:=5
  - $T2$: waitTimer[timer > 0]/timer:=timer - 1
- **closing**
  - $T3$: ready[timer==0]
  - $T4$: closing
- **closed**
  - $T5$: buttonInterrupt/timer:=3
  - $T6$: fullyClosed
  - $T7$: closeTimer
- **opening**
  - $T8$: open/timer:=10
  - $T9$: opening
- **opened**
  - $T10$: fullyOpened
  - $T11$: openTimer[timer > 0]/timer:=timer - 1
Definition (Control Sink)

A control sink in an EFSM is a set of transitions $\mathcal{K}$ that form a strongly connected component (SCC) such that, for each transition $t$ in $\mathcal{K}$ each successor of $t$ is also in $\mathcal{K}$.

![Diagram of control sink EFSM]

- **Start**: setTimer/timer:=5
- **Wait**: waitTimer[timer > 0]/timer:=timer - 1
- **Closing**: ready[timer==0]
- **Fully Closed**: openTimer[timer>0]/timer:=timer-1
- **Fully Opened**: fullyOpen
- **Open Timer**: openTimer[timer>0]/timer:=timer-1
- **Timeout**: timeout
- **Open**: opening
- **Ready**: ready[timer==0]
- **Close Timer**: closeTimer
Definition (Sink-bounded Paths)

A maximal path $\pi$ is sink-bounded iff there exists a control sink $\mathcal{K}$ such that:

- $\pi$ contains a transition from $\mathcal{K}$
- if $\pi$ is infinite, then all transitions in $\mathcal{K}$ occur infinitely often.
A maximal path $\pi$ is **sink-bounded** iff there exists a control sink $K$ such that:

- $\pi$ contains a transition from $K$
- if $\pi$ is infinite, then all transitions in $K$ occur infinitely often.
Definition (Unfair Sink-bounded Paths)

A maximal path $\pi$ is **unfair sink-bounded** iff there exists a control sink $K$ such that

1. $\pi$ contains a transition from $K$
2. if $\pi$ is infinite, then all transitions in $K$ occur infinitely often.
Definition (Unfair Sink-bounded Paths)

A maximal path $\pi$ is **unfair sink-bounded** iff there exists a control sink $K$ such that

- $\pi$ contains a transition from $K$
- If $\pi$ is infinite, then all transitions in $K$ occur infinitely often.
Definition (Unfair Sink-bounded Paths)

A maximal path \( \pi \) is **unfair sink-bounded** iff there exists a control sink \( \mathcal{K} \) such that

- \( \pi \) contains a transition from \( \mathcal{K} \)
- if \( \pi \) is infinite, then all transitions in \( \mathcal{K} \) occur infinitely often.
**Definition (Control Dependence)**

\[ T_i \xrightarrow{\text{CD}} T_j \text{ means that a transition } T_j \text{ is control dependent on a transition } T_i \text{ iff } T_i \text{ has at least one sibling } T_k \text{ such that:} \]

1. for all paths \( \pi \in \text{PATHs}(\text{target}(T_i)) \), the \( \text{source}(T_j) \) belongs to \( \pi \);
2. there exists a path \( \pi \in \text{PATHs}(\text{source}(T_k)) \) such that the \( \text{source}(T_j) \) does not belong to \( \pi \).
**Definition (Control Dependence)**

\( T_i \xrightarrow{\text{CD}} T_j \) means that a transition \( T_j \) is control dependent on a transition \( T_i \) iff \( T_i \) has at least one sibling \( T_k \) such that:

1. for all paths \( \pi \in \text{PATHs}(\text{target}(T_i)) \), the source(\( T_j \)) belongs to \( \pi \);
2. there exists a path \( \pi \in \text{PATHs}(\text{source}(T_k)) \) such that the source(\( T_j \)) does not belong to \( \pi \).

<table>
<thead>
<tr>
<th>CD</th>
<th>PATH type</th>
</tr>
</thead>
<tbody>
<tr>
<td>NTSCD</td>
<td>Maximal Path</td>
</tr>
<tr>
<td>NTICD</td>
<td>Sink-bounded Path</td>
</tr>
<tr>
<td>UNTICD</td>
<td>Unfair Sink-bounded Path</td>
</tr>
</tbody>
</table>
Example (NTSCD)

- $T_3 \xrightarrow{\text{NTSCD}} T_4, T_5, T_6$
- $T_5 \xrightarrow{\text{NTSCD}} T_9, T_{10}$

**States:**
- **Start**
- **wait**
- **closing**
- **opened**
- **opening**
- **closed**

**Transitions:**
- $T_1$: setTimer/timer:= 5
- $T_2$: waitTimer[timer > 0]/timer:=timer - 1
- $T_3$: ready[timer==0]
- $T_4$: closing
- $T_5$: buttonInterrupt/timer:=3
- $T_6$: fullyClosed
- $T_7$: closeTimer
- $T_8$: open/timer:=10
- $T_9$: opening
- $T_10$: fullyOpened
- $T_{11}$: openTimer[timer > 0]/timer:=timer-1

**Notes:**
- Slicing of EFSMs
- Zheng Li
- CREST, KING’S COLLEGE LONDON
Example (NTSCD)

$T_3 \xrightarrow{\text{NTSCD}} T_4, T_5, T_6$

$T_5 \xrightarrow{\text{NTSCD}} T_9, T_{10}$
Example (NTICD)

NO NTICD in this example

Slicing of EFSMs Zheng Li CREST, KING’S COLLEGE LONDON 22 / 34
Example (UNTICCD)

- $T_3$  

- $T_5 \xrightarrow{UNTICD} T_9, T_{10}$

Diagram:
- Start
  - $T_1$: setTimer/timer:=5
  - $T_2$: waitTimer[timer > 0]/timer:=timer - 1
  - $T_3$: ready[timer==0]
  - $T_4$: closing
  - $T_5$: buttonInterrupt/timer:=3
  - $T_6$: fullyClosed
  - $T_7$: closeTimer
  - $T_8$: open/timer:=10
  - $T_9$: opening
  - $T_{10}$: fullyOpened
  - $T_{11}$: openTimer[timer > 0]/timer:=timer-1

Slicing of EFSMs
Zheng Li
CREST, KING’S COLLEGE LONDON
Example (UNTICD)

- $T_3 \xrightarrow{UNTICD} T_5 \xrightarrow{UNTICD} T_9, T_{10}$

Transitions:
- $T_{11}$: openTimer[timer > 0]/timer := timer - 1
- $T_{12}$: timeout
- $T_8$: openTimer[timer > 0]/timer := timer - 1

States:
- Start
- Wait
- Closed
- Opening
- Opened
- Closed

Actions:
- T1: setTimer/timer := 5
- T2: waitTimer[timer > 0]/timer := timer - 1
- T3: ready[timer == 0]
- T4: closing
- T5: buttonInterrupt/timer := 3
- T6: fullyClosed
- T7: closeTimer
- T9: opening
- T10: fullyOpened
Metrics

Definition (Slice Size)

For a model \( M \), \( t' \) is a transition dependent on \( t \) (i.e., \( t' \in T \land t \rightarrow t' \)), the size of slice with respect to \( t \) is:

\[
|S(M, t)| = \frac{\sum t'}{|M|}
\]
**Definition (Slice Size)**

For a model $M$, $t'$ is a transition dependent on $t$ (i.e., $t' \in T \land t \rightarrow t'$), the size of slice with respect to $t$ is:

$$|S(M, t)| = \frac{\sum t'}{|M|}$$

**Definition (Average Slice Size)**

For a model $M$, $NT$ is subset of transitions of $M$ with non-zero slice size (i.e., $NT \subseteq T$ and $\forall t \in NT, |S(M, t)| > 0$). Thus, the average slice size of $M$ is:

$$\text{Avg}(M) = \frac{\sum_{t \in NT} |S(M, t)|}{|NT|}$$
<table>
<thead>
<tr>
<th>Models</th>
<th>#S</th>
<th>#T</th>
<th>#V</th>
<th>EXIT</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATM</td>
<td>9</td>
<td>23</td>
<td>8</td>
<td>Yes</td>
<td>Automated Teller Machine</td>
</tr>
<tr>
<td>Cashier</td>
<td>12</td>
<td>21</td>
<td>10</td>
<td>Yes</td>
<td>Cashier Machine</td>
</tr>
<tr>
<td>CruiseControl</td>
<td>5</td>
<td>17</td>
<td>18</td>
<td>Yes</td>
<td>Cruise Control System</td>
</tr>
<tr>
<td>FuelPump</td>
<td>13</td>
<td>25</td>
<td>12</td>
<td>Yes</td>
<td>Fuel Pump System</td>
</tr>
<tr>
<td>PrinTok</td>
<td>11</td>
<td>89</td>
<td>5</td>
<td>Yes</td>
<td>Print Token</td>
</tr>
<tr>
<td>VendingMachine</td>
<td>7</td>
<td>28</td>
<td>7</td>
<td>Yes</td>
<td>Vending Machine system</td>
</tr>
<tr>
<td>INRES</td>
<td>8</td>
<td>18</td>
<td>8</td>
<td>No</td>
<td>INRES protocol</td>
</tr>
<tr>
<td>TCP</td>
<td>12</td>
<td>57</td>
<td>31</td>
<td>No</td>
<td>TCP Standard(RFC793)</td>
</tr>
<tr>
<td>TCSbin</td>
<td>24</td>
<td>65</td>
<td>61</td>
<td>No</td>
<td>Telephony Control Protocol</td>
</tr>
<tr>
<td>Lift</td>
<td>6</td>
<td>12</td>
<td>1</td>
<td>No</td>
<td>Lift System</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>107</td>
<td>355</td>
<td>161</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Slice Size

<table>
<thead>
<tr>
<th>Dependence</th>
<th>Forward Slices</th>
<th>Backward Slices</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># T</td>
<td>Avg</td>
</tr>
<tr>
<td>DD+NTSCD</td>
<td>276</td>
<td>87.45%</td>
</tr>
<tr>
<td>DD+NTICD</td>
<td>220</td>
<td>61.99%</td>
</tr>
<tr>
<td>DD+UNTICD</td>
<td>267</td>
<td>83.20%</td>
</tr>
<tr>
<td>DD</td>
<td>161</td>
<td>35.67%</td>
</tr>
<tr>
<td>NTSCD</td>
<td>205</td>
<td>86.10%</td>
</tr>
<tr>
<td>NTICD</td>
<td>92</td>
<td>78.67%</td>
</tr>
<tr>
<td>UNTICD</td>
<td>190</td>
<td>82.21%</td>
</tr>
</tbody>
</table>
## Slice Size

<table>
<thead>
<tr>
<th>Dependence</th>
<th>Forward Slices</th>
<th>Backward Slices</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># T</td>
<td>Avg</td>
</tr>
<tr>
<td>DD+NTSCD</td>
<td>276</td>
<td>87.45%</td>
</tr>
<tr>
<td>DD+NTICD</td>
<td>220</td>
<td>61.99%</td>
</tr>
<tr>
<td>DD+UNTICD</td>
<td>267</td>
<td>83.20%</td>
</tr>
<tr>
<td>DD</td>
<td>161</td>
<td>35.67%</td>
</tr>
<tr>
<td>NTSCD</td>
<td>205</td>
<td>86.10%</td>
</tr>
<tr>
<td>NTICD</td>
<td>92</td>
<td>78.67%</td>
</tr>
<tr>
<td>UNTICD</td>
<td>190</td>
<td>82.21%</td>
</tr>
</tbody>
</table>
### Slice Size

<table>
<thead>
<tr>
<th>Dependence</th>
<th>Forward Slices</th>
<th>Backward Slices</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># T</td>
<td>Avg</td>
</tr>
<tr>
<td>DD+NTSCD</td>
<td>276</td>
<td>87.45%</td>
</tr>
<tr>
<td>DD+NTICD</td>
<td>220</td>
<td>61.99%</td>
</tr>
<tr>
<td>DD+UNTICD</td>
<td>267</td>
<td>83.20%</td>
</tr>
<tr>
<td>DD</td>
<td>161</td>
<td>35.67%</td>
</tr>
<tr>
<td>NTSCD</td>
<td>205</td>
<td>86.10%</td>
</tr>
<tr>
<td>NTICD</td>
<td>92</td>
<td>78.67%</td>
</tr>
<tr>
<td>UNTICD</td>
<td>190</td>
<td>82.21%</td>
</tr>
</tbody>
</table>
Backward Slice size using NTICD

Average Slice Size

- Data-NTICD
- NTICD
- Data

Slicing of EFSMs
Zheng Li
CREST, KING’S COLLEGE LONDON
## Correlation of Slice Size

<table>
<thead>
<tr>
<th>Model</th>
<th>Dependence</th>
<th>Forward</th>
<th>Backward</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>NTICD</td>
<td>UNTICD</td>
</tr>
<tr>
<td>ATM</td>
<td>NTICD</td>
<td>-</td>
<td>1.000</td>
</tr>
<tr>
<td></td>
<td>UNTICD</td>
<td>1.000</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>NTSCD</td>
<td>.652</td>
<td>652.</td>
</tr>
<tr>
<td>Cashier</td>
<td>NTICD</td>
<td>-</td>
<td>1.000</td>
</tr>
<tr>
<td></td>
<td>UNTICD</td>
<td>1.000</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>NTSCD</td>
<td>.898</td>
<td>.898</td>
</tr>
<tr>
<td>CruiseControl</td>
<td>NTICD</td>
<td>-</td>
<td>1.000</td>
</tr>
<tr>
<td></td>
<td>UNTICD</td>
<td>1.000</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>NTSCD</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>FuelPump</td>
<td>NTICD</td>
<td>-</td>
<td>1.000</td>
</tr>
<tr>
<td></td>
<td>UNTICD</td>
<td>1.000</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>NTSCD</td>
<td>.786</td>
<td>.786</td>
</tr>
<tr>
<td>PrinTok</td>
<td>NTICD</td>
<td>-</td>
<td>1.000</td>
</tr>
<tr>
<td></td>
<td>UNTICD</td>
<td>1.000</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>NTSCD</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>VendingMachine</td>
<td>NTICD</td>
<td>-</td>
<td>1.000</td>
</tr>
<tr>
<td></td>
<td>UNTICD</td>
<td>1.000</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>NTSCD</td>
<td>.360</td>
<td>.360</td>
</tr>
<tr>
<td>INRES</td>
<td>NTICD</td>
<td>-</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>UNTICD</td>
<td>x</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>NTSCD</td>
<td>x</td>
<td>1.000</td>
</tr>
<tr>
<td>Lift</td>
<td>NTICD</td>
<td>-</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>UNTICD</td>
<td>x</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>NTSCD</td>
<td>x</td>
<td>.813</td>
</tr>
<tr>
<td>TCP</td>
<td>NTICD</td>
<td>-</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>UNTICD</td>
<td>x</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>NTSCD</td>
<td>x</td>
<td>.1000</td>
</tr>
<tr>
<td>TCSbin</td>
<td>NTICD</td>
<td>-</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>UNTICD</td>
<td>x</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>NTSCD</td>
<td>x</td>
<td>1.000</td>
</tr>
</tbody>
</table>
## Correlation of Slice Size

<table>
<thead>
<tr>
<th>Model</th>
<th>Dependence</th>
<th>Forward</th>
<th>Backward</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>NTICD</td>
<td>UNTICD</td>
</tr>
<tr>
<td>ATM</td>
<td>NTICD</td>
<td>-</td>
<td>1.000</td>
</tr>
<tr>
<td></td>
<td>UNTICD</td>
<td>1.000</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>NTSCD</td>
<td>.652</td>
<td>652.</td>
</tr>
<tr>
<td>Cashier</td>
<td>NTICD</td>
<td>-</td>
<td>1.000</td>
</tr>
<tr>
<td></td>
<td>UNTICD</td>
<td>1.000</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>NTSCD</td>
<td>.898</td>
<td>.898</td>
</tr>
<tr>
<td>CruiseControl</td>
<td>NTICD</td>
<td>-</td>
<td>1.000</td>
</tr>
<tr>
<td></td>
<td>UNTICD</td>
<td>1.000</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>NTSCD</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>FuelPump</td>
<td>NTICD</td>
<td>-</td>
<td>1.000</td>
</tr>
<tr>
<td></td>
<td>UNTICD</td>
<td>1.000</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>NTSCD</td>
<td>.786</td>
<td>.786</td>
</tr>
<tr>
<td>PrinTok</td>
<td>NTICD</td>
<td>-</td>
<td>1.000</td>
</tr>
<tr>
<td></td>
<td>UNTICD</td>
<td>1.000</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>NTSCD</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>VendingMachine</td>
<td>NTICD</td>
<td>-</td>
<td>1.000</td>
</tr>
<tr>
<td></td>
<td>UNTICD</td>
<td>1.000</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>NTSCD</td>
<td>.360</td>
<td>.360</td>
</tr>
<tr>
<td>INRES</td>
<td>NTICD</td>
<td>-</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>UNTICD</td>
<td>x</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>NTSCD</td>
<td>x</td>
<td>1.000</td>
</tr>
<tr>
<td>Lift</td>
<td>NTICD</td>
<td>-</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>UNTICD</td>
<td>x</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>NTSCD</td>
<td>x</td>
<td>.813</td>
</tr>
<tr>
<td>TCP</td>
<td>NTICD</td>
<td>-</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>UNTICD</td>
<td>x</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>NTSCD</td>
<td>x</td>
<td>-</td>
</tr>
<tr>
<td>TCSbin</td>
<td>NTICD</td>
<td>-</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>UNTICD</td>
<td>x</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>NTSCD</td>
<td>x</td>
<td>1.000</td>
</tr>
</tbody>
</table>
Properties of Control Dependence

- UNTICD and NTSCD dependences for all transitions within control sinks are identical.
- UNTICD and NTICD dependences for all transitions outside of control sinks are identical.
- The transitive closure for NTICD is contained in the transitive closure for UNTICD.
Properties of Control Dependence

Proposition

For an EFSM $M$, if $T_i \in M$ is a self-looping transition, then there is no transition $T_j$ that is control dependent (NTSCD, NTICD or UNTICD) on $T_i$. 
**Proposition**

For an EFSM $M$, if two transitions $T_i$ and $T_j$ have the same source and target states, and $T_i \xrightarrow{\text{CD}} T_l$ (using NTSCD, NTICD or UNTICD) then $T_j \xrightarrow{\text{CD}} T_l$ (using NTSCD, NTICD or UNTICD respectively).
Properties of Control Dependence

Proposition

For an EFSM $M$, if all states $s \in M$ where $s \neq \text{START}$ have a transition $T_i$ where $\text{source}(T_i) = s$ and $\text{target}(T_i) = \text{EXIT}$, then the set of transitions that are directly control dependent on $T_i$ are the same for all types of control dependence, i.e. NTSCD, NTICD and UNTICD.
Summary of Dependence Analysis

- NTSCD, NTICD and UNTICD are defined for EFSM
- The properties are formally proved
- Empirically studies on dependence size
Questions?

zheng.li@kcl.ac.uk

http://slim.dcs.kcl.ac.uk/