Slicing of Extended Finite State Machines

Kelly Androutsopoulos[1], David Clark[1], Nicolas Gold[1],

Mark Harman[1], Rob Hierons[2], Zheng Li[1] and Laurence Tratt[3]

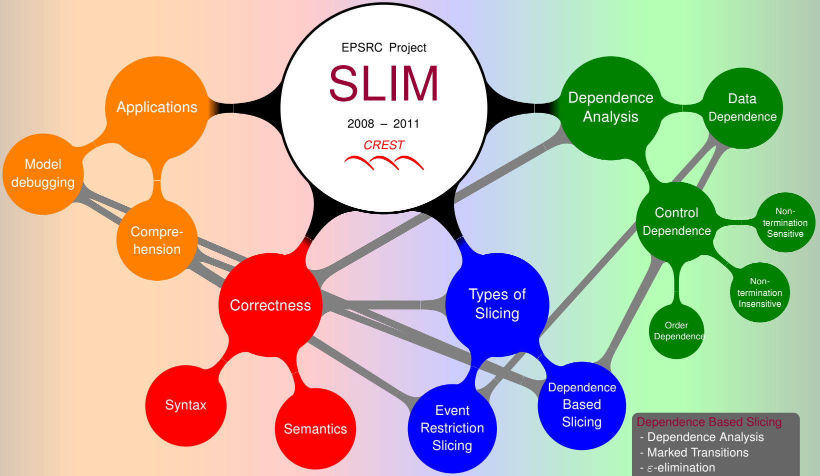
Centre for Research in Evolution,
Search & Testing

[1]CREST, King's College London

[2]Brunel University

[3]Bournemouth University

SLIcing state based Models



Benchmarks

ATM
Cashier
CruiseControl
FuelPump
PrintToken
VendingMachine
DoorController
INRES protocol
TCP
TCSbin(Motorola)

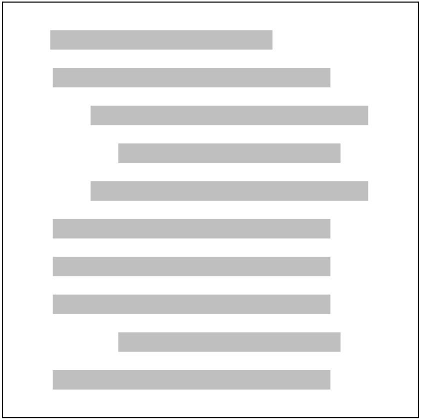
Dependence Based Slicing
- Dependence Analysis
- Marked Transitions
- ε-elimination
- Minimisation

Event Restriction Slicing
- BasicSlice
- Constant Propagation
- R-mergeEquivalentStates
- G-mergeEquivalentStates

KING'S College LONDON

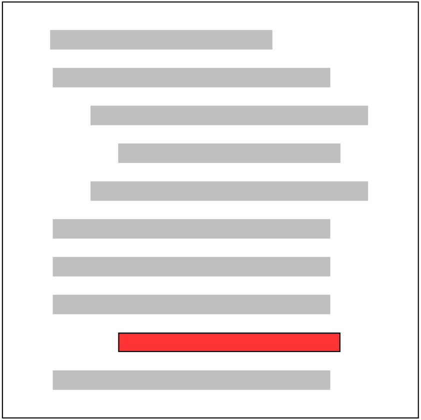
slim.dcs.kcl.ac.uk

Program Slicing



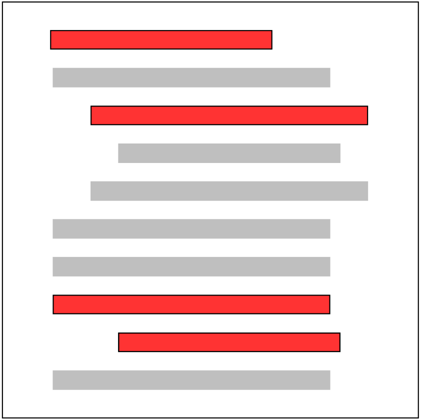
Program Slicing

which other lines affect the selected line?



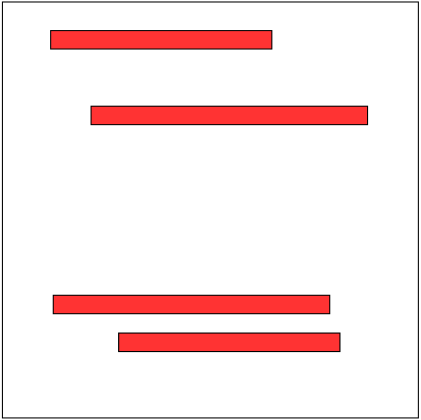
we only care about this line

which other lines affect the selected line?



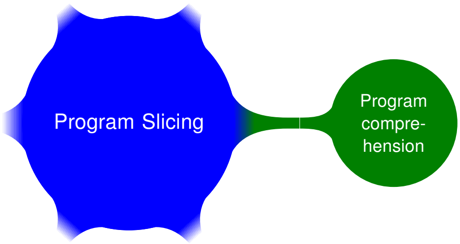
we only care about this line

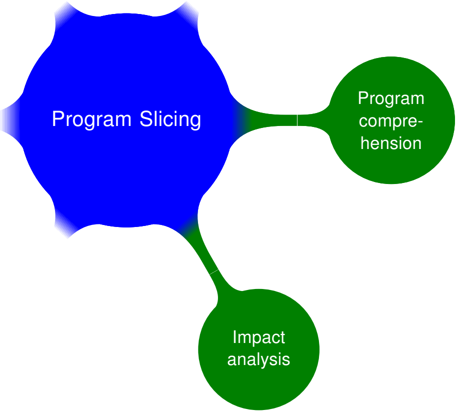
which other lines affect the selected line?

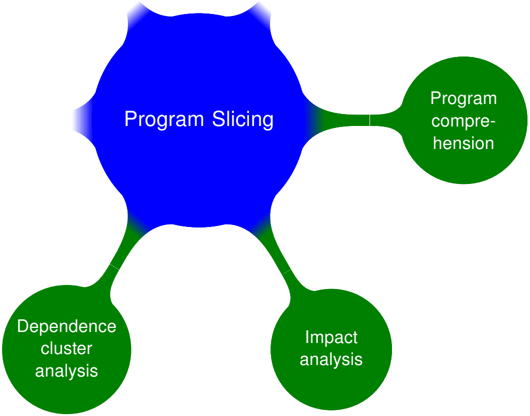


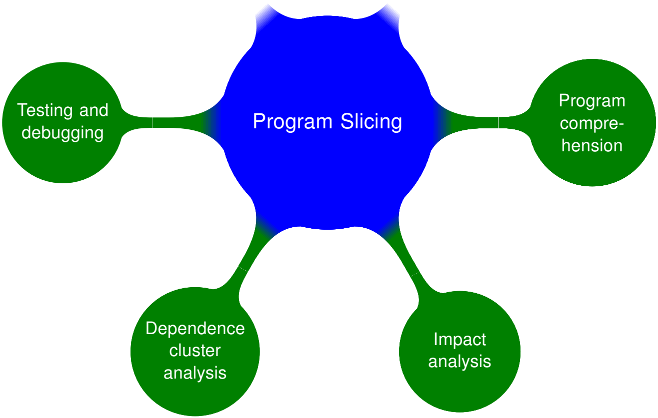
we only care about this line

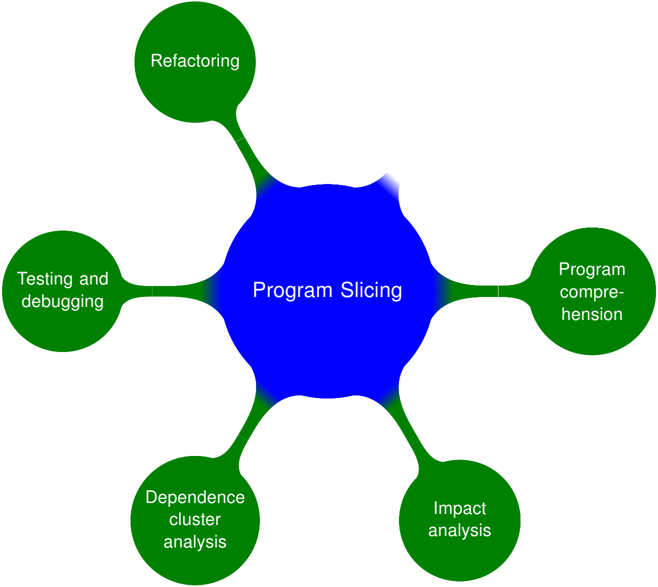
Program Slicing

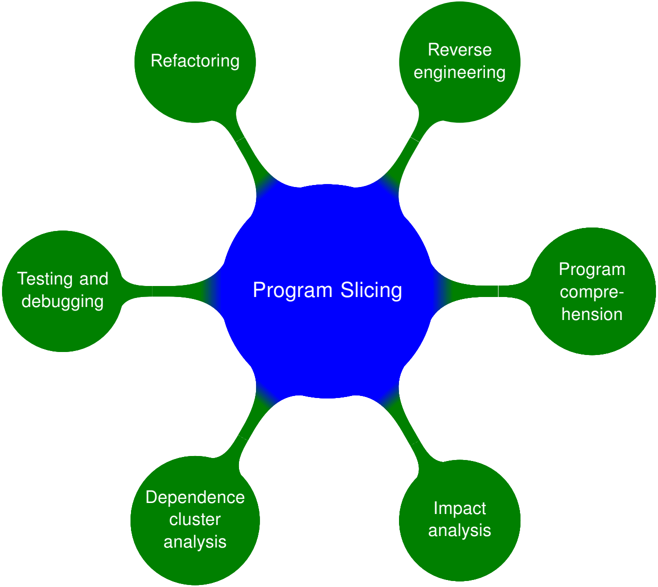

Program Slicing
Program compre-hension


Program Slicing
Program compre- hension
Impact analysis


Program Slicing
Program compre-
hension
Dependence
cluster
analysis
Impact
analysis

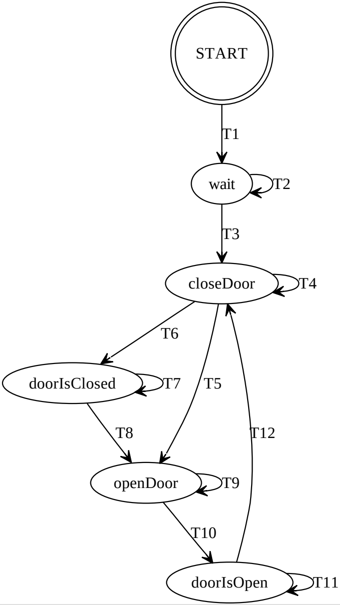

Program Slicing
Testing and debugging
Program compre-hension
Dependence cluster analysis
Impact analysis


Refactoring
Testing and debugging
Program Slicing
Program compre-hension
Dependence cluster analysis
Impact analysis

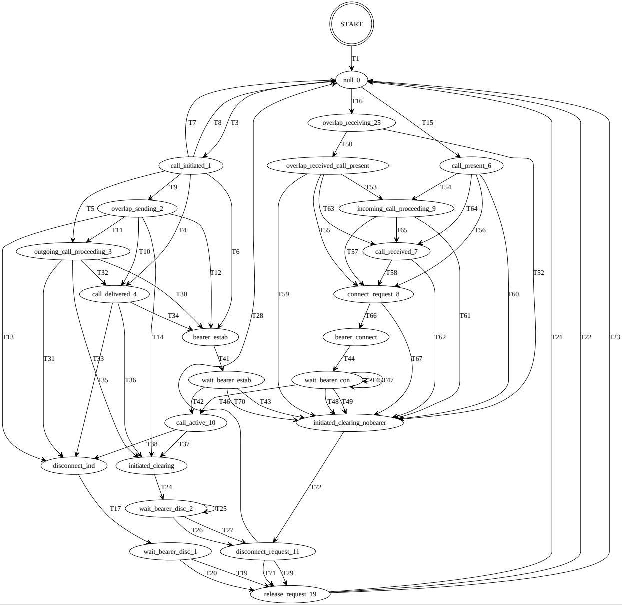

Program Slicing
Refactoring
Reverse engineering
Testing and debugging
Program compre- hension
Dependence cluster analysis
Impact analysis

Can slicing be applied to model level?

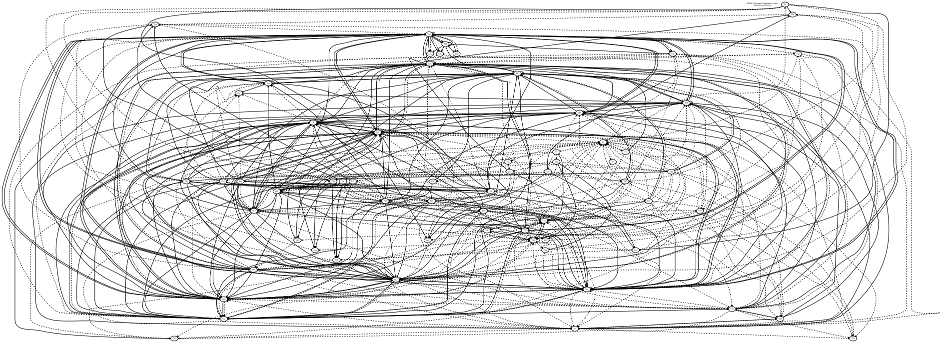
Model


START
T1
wait
T2
T3
closeDoor
T4
T6
doorIsClosed
T7
T5
T8
T12
openDoor
T9
T10
doorIsOpen
T11

Model


START
T1
null_0
T16
T15
T7
T8
T3
overlap_receiving_25
T50
call_initiated_1
overlap_received_call_present
call_present_6
T9
T53
T54
T5
overlap_sending_2
T63
incoming_call_proceeding_9
T64
T11
T4
T65
T55
T56
outgoing_call_proceeding_3
T10
T6
T57
call_received_7
T32
T12
T58
T52
call_delivered_4
T30
T59
connect_request_8
T60
T34
T28
T66
T61
T13
T14
bearer_estab
bearer_connect
T62
T21
T22
T23
T31
T33
T41
T44
T67
T35
T36
wait_bearer_estab
wait_bearer_con
T45
T47
T42
T46
T70
T43
T48
T49
call_active_10
initiated_clearing_nobearer
T38
T37
disconnect_ind
initiated_clearing
T24
T72
T17
wait_bearer_disc_2
T25
T26
T27
wait_bearer_disc_1
disconnect_request_11
T20
T19
T71
T29
release_request_19

If the model like this?



Motivation

- Models tend to be larger and more complex.

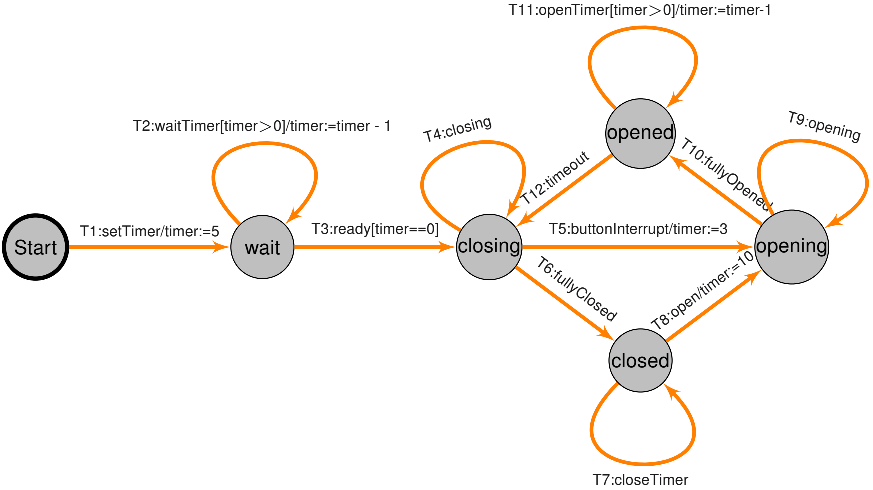
Motivation

- 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.

Definition

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 $source(t) \in S$, a target state $target(t) \in S$ and a label $lbl(t)$. 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



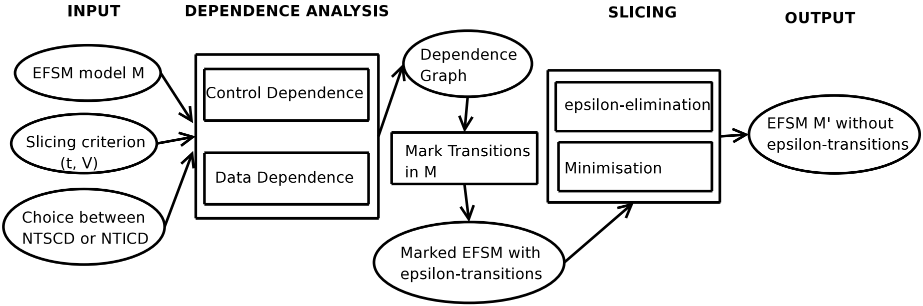
Definition (Slicing Criterion)

A slicing criterion for an EFSM is a pair (t, V) where transition $t \in T$ and variable set $V \subseteq Var$. It designates the point in the evaluation immediately after the execution of the action contain in transition t.

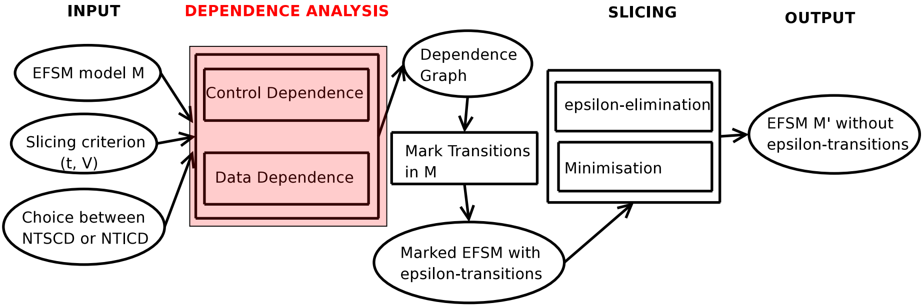
Definition (Slice)

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.

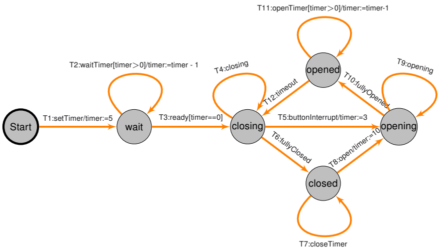
The CREST EFSM slicing tool



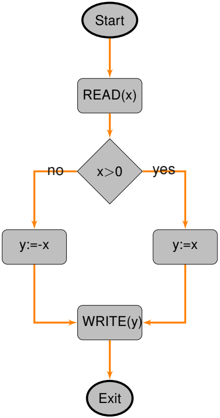
The CREST EFSM slicing tool



EFSM VS CFG

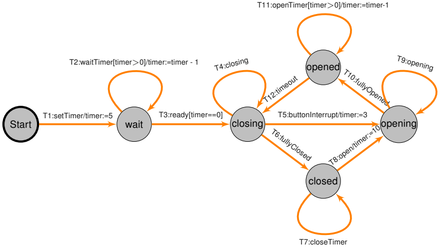


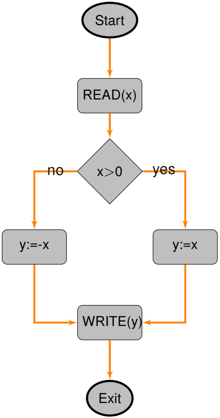
T11:openTimer[timer>0]/timer:=timer-1
T2:waitTimer[timer>0]/timer:=timer - 1
T4:closing
opened
T9:opening
T12:timeout
T10:fullyOpened
Start
T1:setTimer/timer:=5
wait
T3:ready[timer==0]
closing
T5:buttonInterrupt/timer:=3
opening
T6:fullyClosed
T8:open/timer:=10
closed
T7:closeTimer



Start
READ(x)
no
x>0
yes
y:=-x
y:=x
WRITE(y)
Exit

EFSM VS CFG

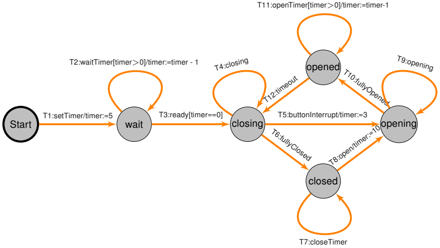


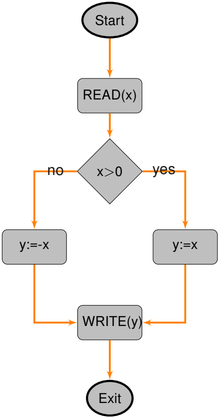


Difference

1. Transition in EFSM VS Node in CFG

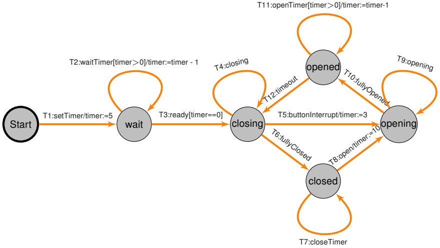
EFSM VS CFG

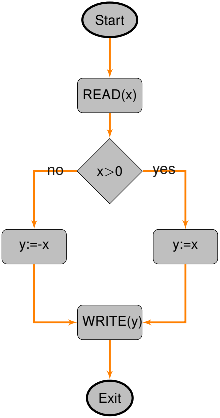




Difference

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





Difference

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

Dependence Analysis of EFSM

- Data Dependence
- Control Dependence

- Data Dependence
- Control Dependence
  - Traditional Control Dependence [Korel et al, ICSM 2003]

Dependence Analysis of EFSM

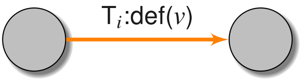
- Data Dependence
- Control Dependence
  - Traditional Control Dependence [Korel et al, ICSM 2003]
  - Non-Termination Insensitive Control Dependence (NTICD) [Ranganath et al. ESOP 2005]
  - Non-Termination Sensitive Control Dependence (NTSCD) [Ranganath et al. ESOP 2005]

- 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]

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;



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:

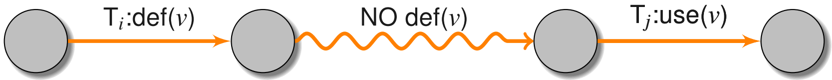
1. $v \in D(T_i)$, where $D(T_i)$ is a set of variables defined by transition T_i, i.e. variables defined by actions and by the event of T_i;
2. $v \in U(T_j)$, where $U(T_j)$ is a set of variables used in a condition and actions of transition T_j;

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;
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.

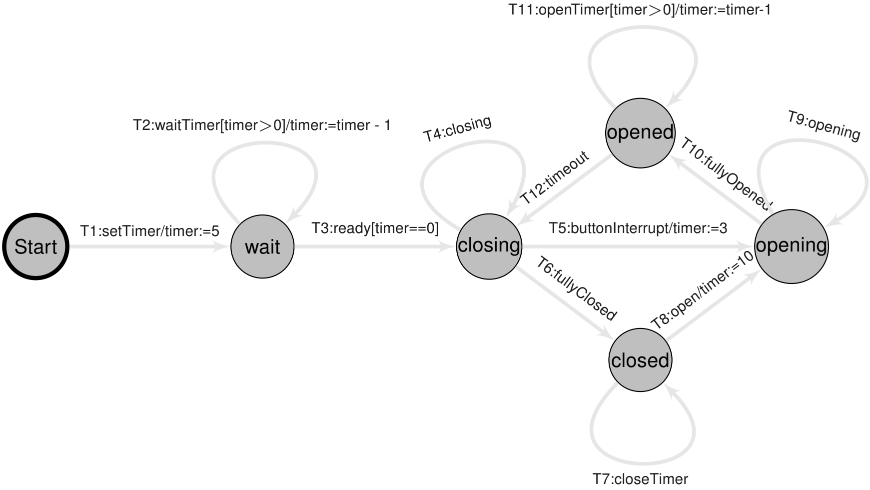


Types of Control Dependence and Paths in EFSM

Name		Path type
NTSCD	$\longrightarrow$	Maximal Path
NTICD	$\longrightarrow$	Sink-bounded Path
UNTICD	$\longrightarrow$	Unfair Sink-bounded Path

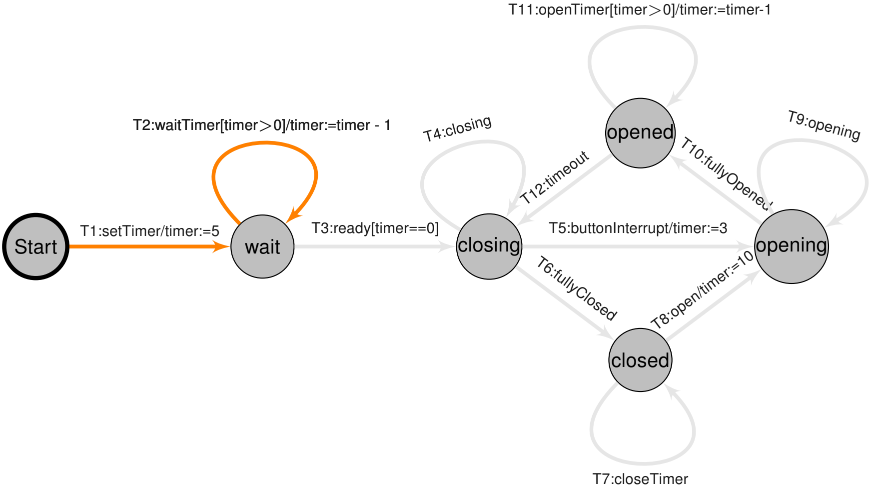
Definition (Maximal Path)

A maximal path is any path that terminates in a final transition, or is infinite.



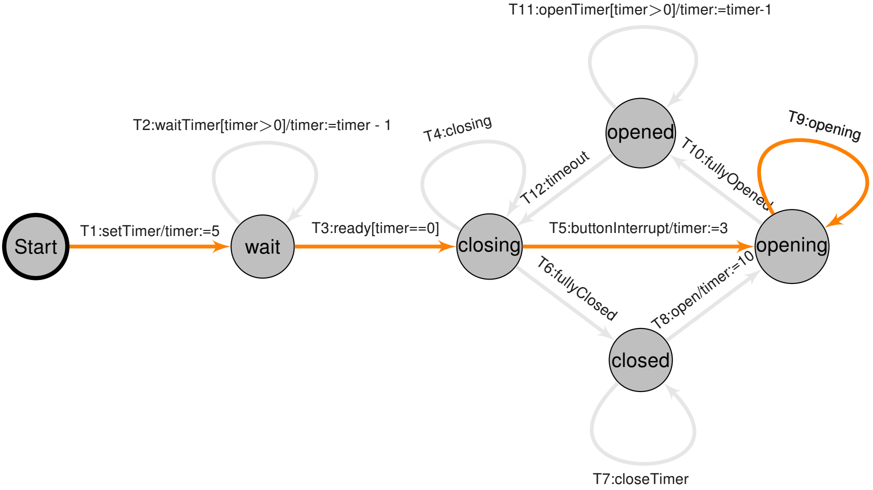
Definition (Maximal Path)

A maximal path is any path that terminates in a final transition, or is infinite.



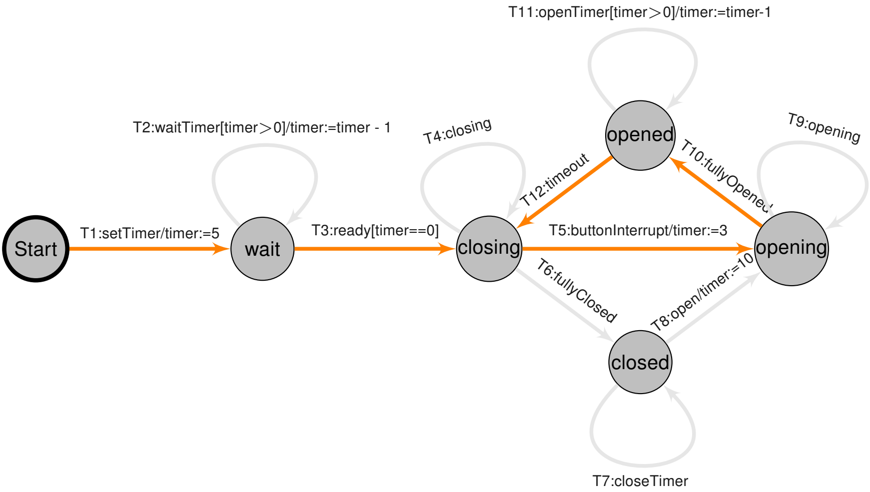
Definition (Maximal Path)

A maximal path is any path that terminates in a final transition, or is infinite.



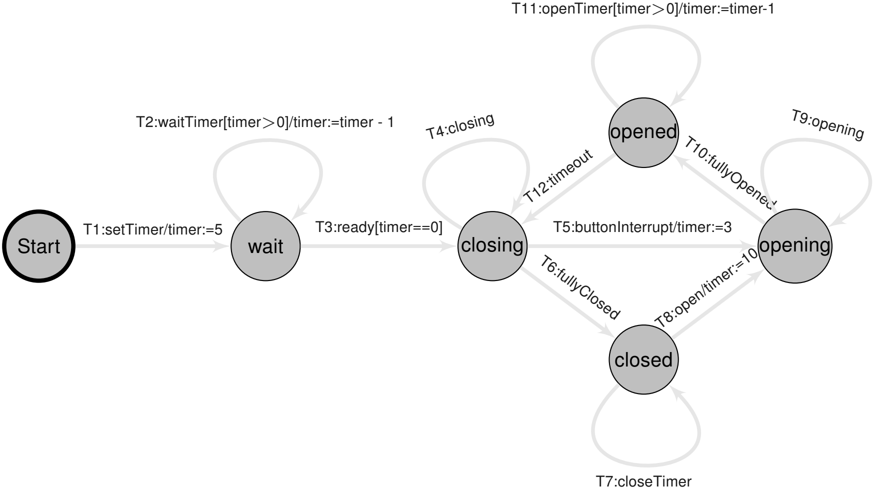
Definition (Maximal Path)

A maximal path is any path that terminates in a final transition, or is infinite.



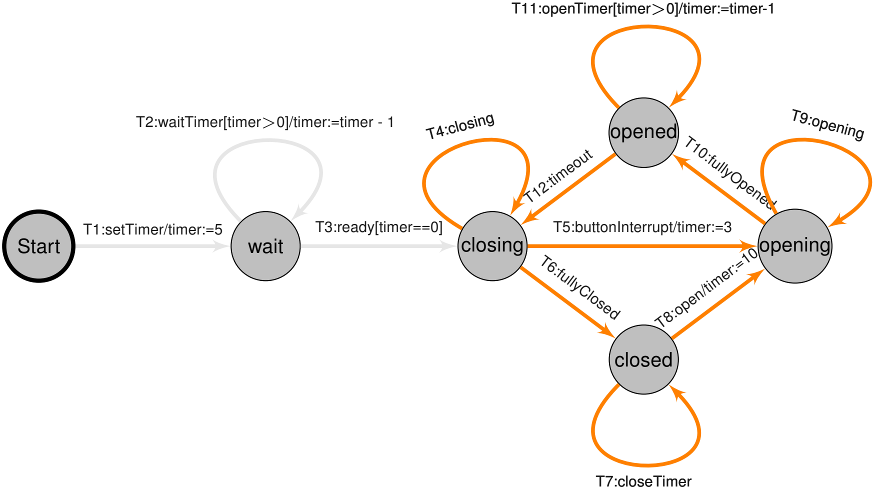
Definition (Control Sink)

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 $\mathcal{K}$.



Definition (Control Sink)

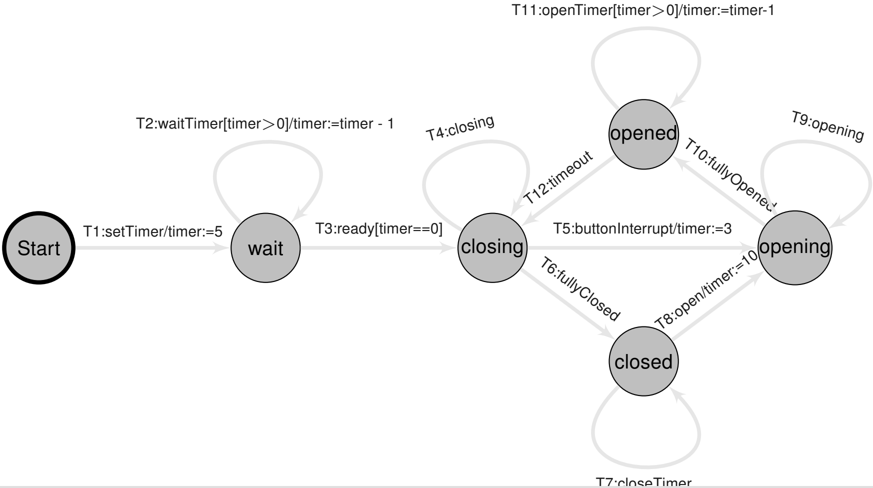
A control sink in an EFSM is a set of transitions $\mathcal{K}$ that form a strongly connected component (SCC) such that, for each transition t in $\mathcal{K}$ each successor of t is also in $\mathcal{K}$.



Definition (Sink-bounded Paths)

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

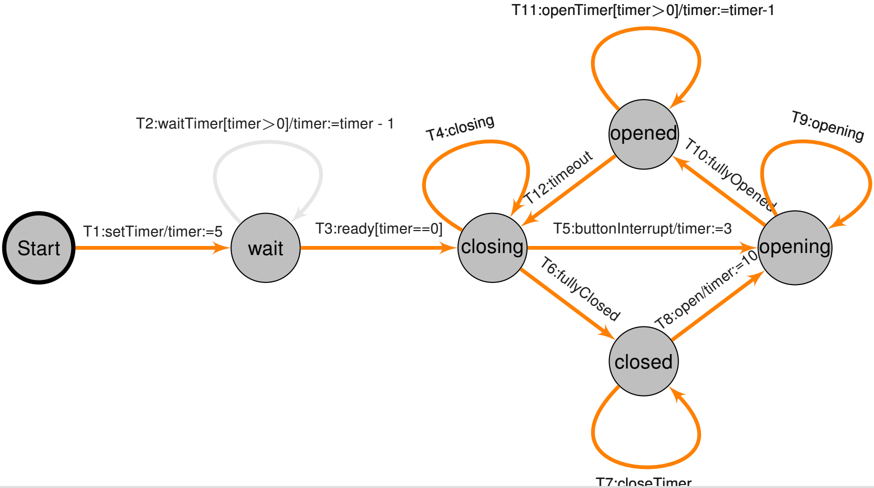
- π contains a transition from $\mathcal{K}$
- if π is infinite, then all transitions in $\mathcal{K}$ occur infinitely often.



Definition (Sink-bounded Paths)

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

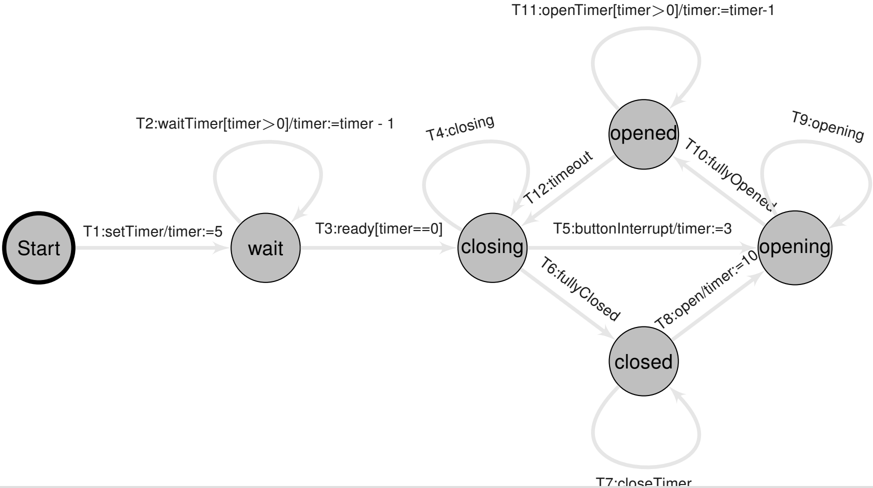
- π contains a transition from $\mathcal{K}$
- if π is infinite, then all transitions in $\mathcal{K}$ occur infinitely often.



Definition (Unfair Sink-bounded Paths)

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

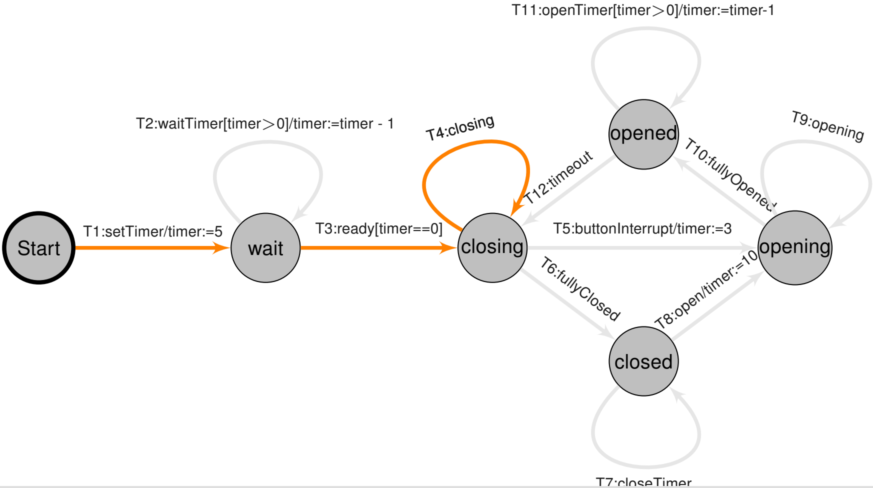
- π contains a transition from $\mathcal{K}$
- ~~if π is infinite, then all transitions in $\mathcal{K}$ occur infinitely often.~~



Definition (Unfair Sink-bounded Paths)

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

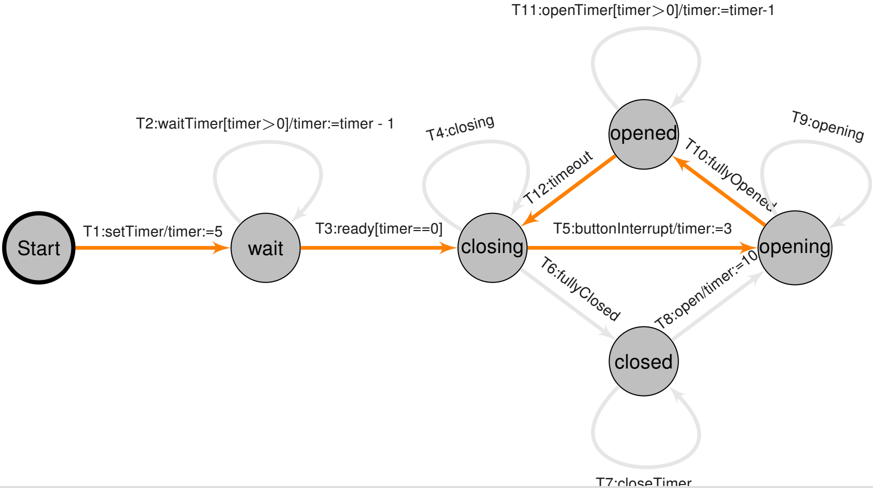
- π contains a transition from $\mathcal{K}$
- ~~if π is infinite, then all transitions in $\mathcal{K}$ occur infinitely often.~~



Definition (Unfair Sink-bounded Paths)

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

- π contains a transition from $\mathcal{K}$
- ~~if π is infinite, then all transitions in $\mathcal{K}$ occur infinitely often.~~



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 $\pi \in$ PATHs$(target(T_i))$, the $source(T_j)$ belongs to π;
2. there exists a path $\pi \in$ PATHs$(source(T_k))$ such that the $source(T_j)$ does not belong to π.

Definition (Control Dependence)

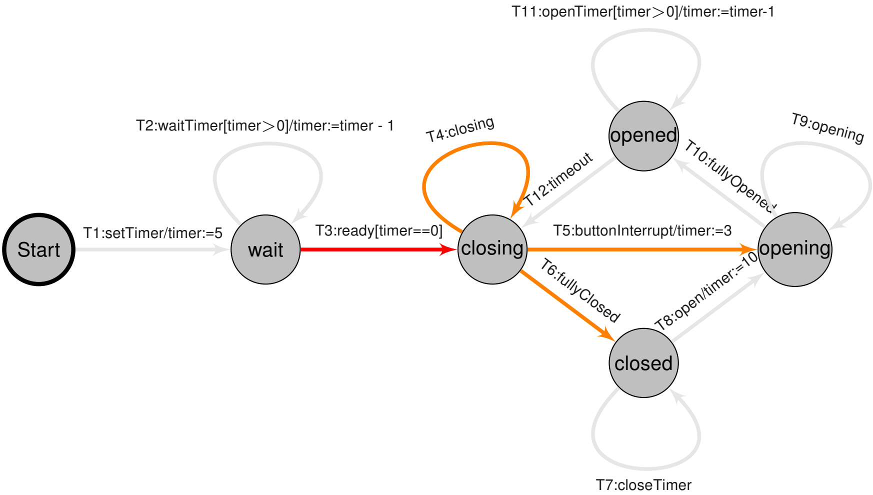
$T_i \xrightarrow{\text{CD}} T_j$ means that a transition T_j is control dependent on a transition T_i iff T_i has at least one sibling T_k such that:

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

CD		PATH type
NTSCD	$\longrightarrow$	Maximal Path
NTICD	$\longrightarrow$	Sink-bounded Path
UNTICD	$\longrightarrow$	Unfair Sink-bounded Path

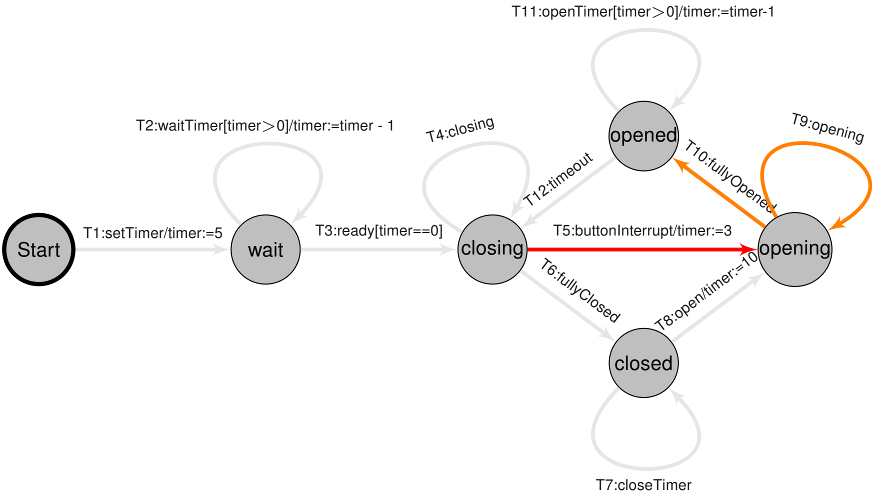
Example (NTSCD)

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



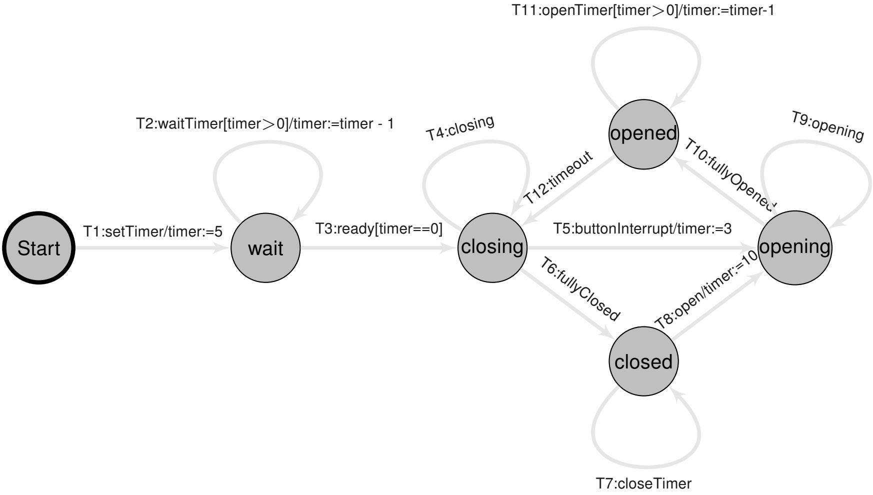
Example (NTSCD)

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



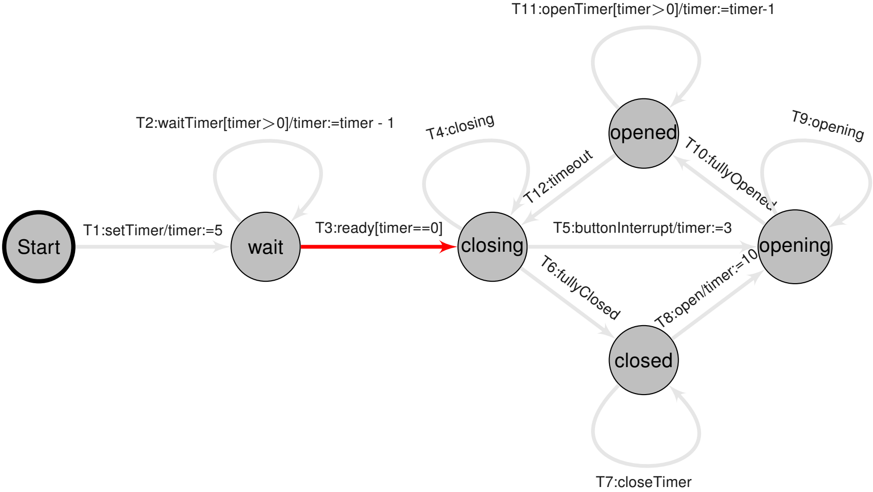
Example (NTICD)

NO NTICD in this example



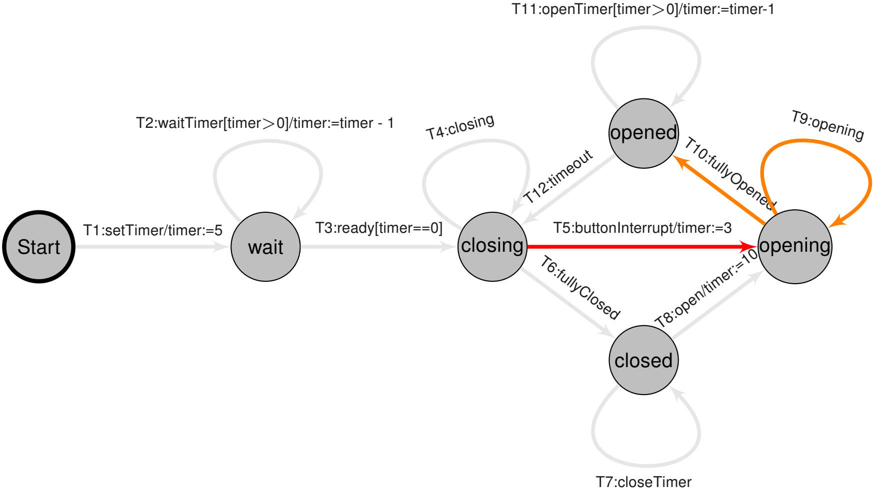
Example (UNTICD)

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



Example (UNTICD)

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



Definition (Slice Size)

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

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

Definition (Slice Size)

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

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

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 $\forall t \in NT, |\mathcal{S}(M,t)| > 0$) .Thus, the average slice size of M is:

$$\mathsf{Avg}(M) = \frac{\sum\limits_{t \in NT} |\mathcal{S}(M,t)|}{|NT|}$$

Subjects

Models	#S	#T	#V	EXIT	Description
ATM	9	23	8	Yes	Automated Teller Machine
Cashier	12	21	10	Yes	Cashier Machine
CruiseControl	5	17	18	Yes	Cruise Control System
FuelPump	13	25	12	Yes	Fuel Pump System
PrinTok	11	89	5	Yes	Print Token
VendingMachine	7	28	7	Yes	Vending Machine system
INRES	8	18	8	No	INRES protocol
TCP	12	57	31	No	TCP Standard(RFC793)
TCSbin	24	65	61	No	Telephony Control Protocol
Lift	6	12	1	No	Lift System
Total	107	355	161		

	Forward Slices		Backward Slices	
Dependence	# T	Avg	# T	Avg
DD+NTSCD				
DD+NTICD				
DD+UNTICD				
DD				
NTSCD				
NTICD				
UNTICD				

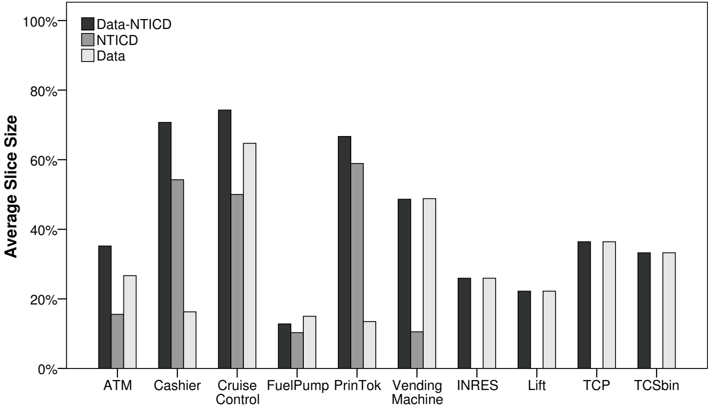
Slice Size

	Forward Slices		Backward Slices	
Dependence	# T	Avg	# T	Avg
DD+NTSCD	276	87.45%		
DD+NTICD	220	61.99%		
DD+UNTICD	267	83.20%		
DD	161	35.67%		
NTSCD	205	86.10%		
NTICD	92	78.67%		
UNTICD	190	82.21%		

Slice Size

Dependence	Forward Slices # T	Forward Slices Avg	Backward Slices # T	Backward Slices Avg
DD+NTSCD	276	87.45%	345	70.46%
DD+NTICD	220	61.99%	278	49.48%
DD+UNTICD	267	83.20%	335	66.83%
DD	161	35.67%	174	33.15%
NTSCD	205	86.10%	336	53.63%
NTICD	92	78.67%	167	44.59%
UNTICD	190	82.21%	313	51.00%

Backward Slice size using NTICD



Correlation of Slice Size

Model	Dependence	Forward			Backward		
		NTICD	UNTICD	NTSCD	NTICD	UNTICD	NTSCD
ATM	NTICD	-	1.000	.652	-	1.000	.941
	UNTICD	1.000	-	.652	1.000	-	.941
	NTSCD	.652	652.	-	.941	.941	-
Cashier	NTICD	-	1.000	.898	-	1.000	1.000
	UNTICD	1.000	-	.898	1.000	-	1.000
	NTSCD	.898	.898	-	1.000	1.000	-
CruiseControl	NTICD	-	1.000	1.000	-	1.000	1.000
	UNTICD	1.000	-	1.000	1.000	-	1.000
	NTSCD	1.000	1.000	-	1.000	1.000	-
FuelPump	NTICD	-	1.000	.786	-	1.000	-.509
	UNTICD	1.000	-	.786	1.000	-	-.509
	NTSCD	.786	.786	-	-.509	-.509	-
PrinTok	NTICD	-	1.000	1.000	-	1.000	1.000
	UNTICD	1.000	-	1.000	1.000	-	1.000
	NTSCD	1.000	1.000	-	1.000	1.000	-
VendingMachine	NTICD	-	1.000	.360	-	1.000	.224
	UNTICD	1.000	-	.360	1.000	-	.224
	NTSCD	.360	.360	-	.224	.224	-
INRES	NTICD	-	x	x	-	x	x
	UNTICD	x	-	1.000	x	-	1.000
	NTSCD	x	1.000	-	x	1.000	-
Lift	NTICD	-	x	x	-	x	x
	UNTICD	x	-	.813	x	-	1.000
	NTSCD	x	.813	-	x	1.000	-
TCP	NTICD	-	x	x	-	x	x
	UNTICD	x	-	1.000	x	-	1.000
	NTSCD	x	.	-	x	1.000	-
TCSbin	NTICD	-	x	x	-	x	x
	UNTICD	x	-	1.000	x	-	1.000
	NTSCD	x	1.000	-	x	1.000	-

Correlation of Slice Size

Model	Dependence	Forward			Backward		
		NTICD	UNTICD	NTSCD	NTICD	UNTICD	NTSCD
ATM	NTICD	-	1.000	.652	-	1.000	.941
	UNTICD	1.000	-	.652	1.000	-	.941
	NTSCD	.652	652.	-	.941	.941	-
Cashier	NTICD	-	1.000	.898	-	1.000	1.000
	UNTICD	1.000	-	.898	1.000	-	1.000
	NTSCD	.898	.898	-	1.000	1.000	-
CruiseControl	NTICD	-	1.000	1.000	-	1.000	1.000
	UNTICD	1.000	-	1.000	1.000	-	1.000
	NTSCD	1.000	1.000	-	1.000	1.000	-
FuelPump	NTICD	-	1.000	.786	-	1.000	-.509
	UNTICD	1.000	-	.786	1.000	-	-.509
	NTSCD	.786	.786	-	-.509	-.509	-
PrinTok	NTICD	-	1.000	1.000	-	1.000	1.000
	UNTICD	1.000	-	1.000	1.000	-	1.000
	NTSCD	1.000	1.000	-	1.000	1.000	-
VendingMachine	NTICD	-	1.000	.360	-	1.000	.224
	UNTICD	1.000	-	.360	1.000	-	.224
	NTSCD	.360	.360	-	.224	.224	-
INRES	NTICD	-	x	x	-	x	x
	UNTICD	x	-	1.000	x	-	1.000
	NTSCD	x	1.000	-	x	1.000	-
Lift	NTICD	-	x	x	-	x	x
	UNTICD	x	-	.813	x	-	1.000
	NTSCD	x	.813	-	x	1.000	-
TCP	NTICD	-	x	x	-	x	x
	UNTICD	x	-	1.000	x	-	1.000
	NTSCD	x	.	-	x	1.000	-
TCSbin	NTICD	-	x	x	-	x	x
	UNTICD	x	-	1.000	x	-	1.000
	NTSCD	x	1.000	-	x	1.000	-

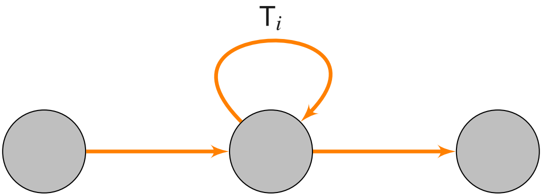
Properties of Control Dependence

- 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.

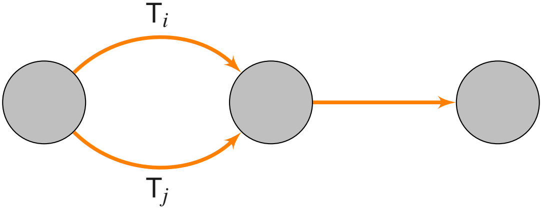
Proposition

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.



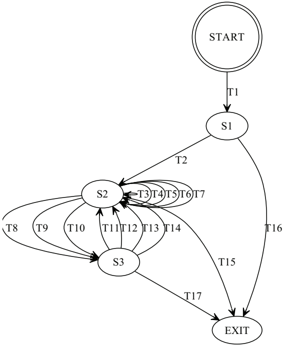
Proposition

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).



Proposition

For an EFSM M, if all states $s \in M$ where $s \neq$ START have a transition T_i where $source(T_i) = s$ and $target(T_i) =$ 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/