Stubborn Sets for Reduced State Space Generation

Seminar Talk

Dominik Winterer

Albert-Ludwigs-Universität Freiburg

February 8, 2016

Transition Systems and Model Checking

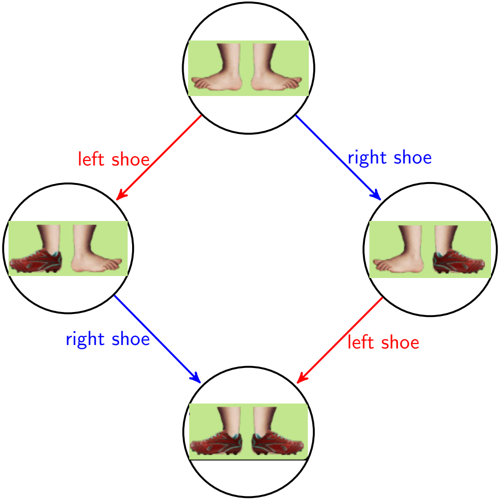
Abstraction

- Shrink transition system to tractable size
- "Solve" smaller transition system
- Use solution for regular transition system

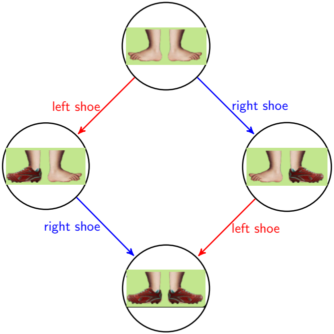
Partial Order Reduction

- Detect structural symmetries
- Fire only necessary transitions in each state

Partial Order Reduction - Example: Putting Shoes on



Partial Order Reduction ctd.



Observations:

- Commutative transitions
- Algorithms do not detect such symmetries without modifications

Example - Concurrent Program

Setting

- Three processes P1, P2, P3 share variables X, Y, Z, R
- Initially: All variables are zero, $X = Y = Z = R = 0$

P1	P2	P3
$X := 1$	Y := 2	Z := 1
$R := X \cdot Y \cdot Z$		

Example - Concurrent Program

Setting

- Three processes P1, P2, P3 share variables X, Y, Z, R
- Initially: All variables are zero, $X = Y = Z = R = 0$

P1	P2	P3
$X := 1$	$\mathrm{Y} := 2$	$\mathrm{Z} := 1$
$R := X \cdot Y \cdot Z$		

Observation: First statements of $P1, P2, P3$ independent

Variable/Transition Systems

Definition (variable/transition system)

A variable/transition system is a five-tuple $(V, T, type, next, ss_0)$, where

- V is a finite set of variables
- T is a finite set of transitions
- *type* is a function assigning a type to each variable
- *next* is the next state function
- ss_0 is the initial state

Variable/Transition Systems ctd.

Concurrent Program as v/t system $(V, T, type, next, ss_0)$ where

- $V = \{X, Y, Z, R\}$
- $T = \{t_1, \cdots, t_4\}$
- $type(v) = INT$ for all $v \in V$
- state encoding XYZR, $next = \{(0000, t_1, 1000), \cdots\}$
- $ss_0 = XYZR = 0000$

P1	P2	P3
t_1 $X := 1$	t_2 $Y := 2$	t_3 $Z := 1$
t_4 $R := X \cdot Y \cdot Z$		

Enabledness/Disabledness

- A transition t is enabled in state s if we can "fire" it
- If transition t is enabled in state s we denote this by $en(s, t)$,
- If transition t is not enabled in state s it is disabled, i.e. $next(s, t) = undefined$
- a state is terminal if there is no enabled transition

Enabledness/Disabledness- Example

Enabled Transitions in $ss_0 = 0000$: t_1, t_2, t_3
Disabled Transitions in $ss_0 = 0000$: t_4
Terminal state: $s = 1212$

P1	P2	P3
t_1 $X := 1$	t_2 $Y := 2$	t_3 $Z := 1$
t_4 $R := X \cdot Y \cdot Z$		

Enabled with respect to a Variable Set

Definition (enabled with respect to variable set)

Transition t is enabled with respect to a set of variables $U \subseteq V$ in state s iff there exist a state s' s.t for all $v \in U : s'(v) = s(v)$

Notation: $en(s, t, U)$

Enabled with Respect to Variable Set - Example

State $XYZR = 1000$: t_4 is enabled with respect to $U = \{X\}$

P1	P2	P3
t_1 $X := 1$	t_2 $Y := 2$	t_3 $Z := 1$
t_4 $R := X \cdot Y \cdot Z$		

Write Up Set

Definition (write up set)

A write up A set w.r.t t and U, $wrup(U, t)$ is a set of transitions that make t enabled w.r.t U in some state s.

Write up Set - Example

Example

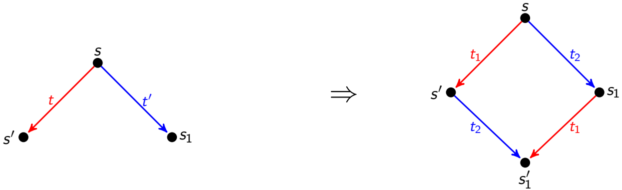
$A = \{t_1\}$ is a write up set w.r.t t_4 and $\{X\}$

P1	P2	P3
t_1 $X := 1$	t_2 $Y := 2$	t_3 $Z := 1$
t_4 $R := X \cdot Y \cdot Z$		

Commutativity - The Diamond Property

Definition (commutativity)

Transition t and t' are commutative iff for every s, s' and s_1 there is a state s_1' such that



Semistubborn Set

Definition (semistubborn set)

A set of transition $T_s \subseteq T$ is *semistubborn* in state s, if and only if for every $t \in T_s$

1. $\neg en(s,t) \implies \exists U \subseteq V : \neg en(s,t,U) \wedge wrup(t,U) \subseteq T_s$
2. $en(s,t) \implies \forall t' \notin T_s : t$ and t' are commutative

Semistubborn Set - Example

A Semistubborn Set in state $ss_0 = 0000$: $T_{ss_0} = \{t_1, t_4\}$
t_1 is enabled and commutative to t_2, t_3
t_4 has write up set $\{t_1\}$ w.r.t to $\{X\}$

P1	P2	P3
t_1 $X := 1$	t_2 $Y := 2$	t_3 $Z := 1$
t_4 $R := X \cdot Y \cdot Z$		

Semistubborn Set - Counterexample I

A Semistubborn Set in state $ss_0 = 0000$: $T_{ss_0} = \emptyset$

Empty $T_{ss_0} \rightarrow$ no conditions to be satisfied

P1

t_1 $X := 1$

t_4 $R := X \cdot Y \cdot Z$

P2

t_2 $Y := 2$

P3

t_3 $Z := 1$

Semistubborn Set - Counterexample II

A Semistubborn Set in state $ss_0 = 0000$: $T_{ss_0} = \{t_5\}$

P1

t_1 $X := 1$

t_4 $R := X \cdot Y \cdot Z$

P2

t_2 $Y := 2$

P3

t_3 $Z := 1$

t_5 $V := 1000$

Stubborn Sets

Definition (stubborn sets)

A set of transitions $T_s \subseteq T$ is stubborn in state s, iff

1. T_s is semistubborn in s
2. T_s contains an enabled transition in s (key transition)

Stubborn Set - Example

A Stubborn Set in state $ss_0 = 0000$: $T_{ss_0} = \{t_1, t_4\}$

P1

t_1 $X := 1$

t_4 $R := X \cdot Y \cdot Z$

P2

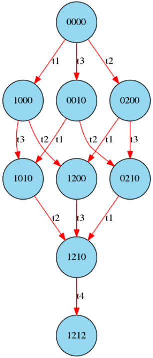
t_2 $Y := 2$

P3

t_3 $Z := 1$

State Space Reduction with Stubborn Sets

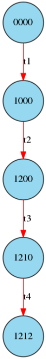
No reduction



States: $2^3 + 1$

Transitions: $3! + 1$

Stubborn Sets



States: $3 + 1$

Transitions: $(3 + 1) + 1$

Computation of Stubborn Sets

Stubborn Set	*Complexity*
non-trivial	NP-hard
minimal enabled	NP-hard
optimal	PSPACE-hard

Properties

- Any superset of a stubborn set is a stubborn set
- Therefore T is stubborn
- Tradeoff reduction/overhead of stubborn set computation

Conclusion

- Stubborn set method: State space reduction technique
- Valmari provided theoretical foundation
- State space reduction can increase the performance/decrease memory usage of verification
- Similar concepts: Ample Sets, Persistent Sets
- Various applications of partial order reduction