DERIVING EXHAUSTIVE TEST SUITES FOR NONDETERMINISTIC FSMs W.R.T. NON-SEPARABILITY RELATION

Ekaterina Akenshina,
Natalia Shabaldina,
Nina Yevtushenko

Tomsk State University,
36 Lenin str., Tomsk, 634050, Russia

OUTLINE

- Introduction
- Relations between NFSMs and ‘all weather conditions’ assumption
- Separating two NFSMs
- Test suite derivation w.r.t. a mutation machine
- Test suite derivation w.r.t. “black box”
- Future work

WHY NONDETERMINISTIC?

- FSMs serve as an underlying model for SDL specifications as well as for UML state charts and those FSMs are nondeterministic
- PROMELA that is used in the model checker SPIN also allows a nondeterministic behavior
- software mutation testing

Nondeterministic descriptions usually are more compact than the corresponding deterministic models

FINITE STATE MACHINE

$$\mathcal{S} = \langle S, I, O, h_S, s_1 \rangle$$

S **– set of states**

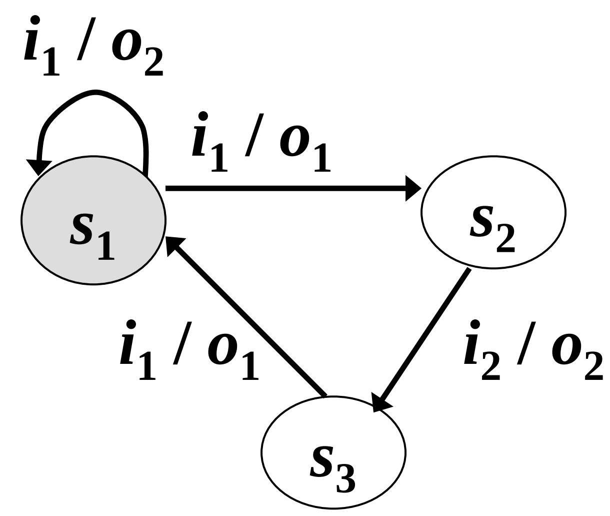
I **– input alphabet**

O **– output alphabet**

h_S **– behavior relation,**

$$h_S \subseteq S \times I \times O \times S$$

s_1 **– the initial state**



$$out_S(s, \alpha) = \{ \beta \mid \exists\, s' \in S \ (s, \alpha, \beta, s') \in h_S \}$$

RELATIONS BETWEEN NFSMs

- **FSMs *T* and *S* are *equivalent* if**

  $$\forall \alpha \in I^* \, (out_T(t_1,\alpha) = out_S(s_1,\alpha))$$

- **FSM *T* is a *reduction* of *S* if**

  $$\forall \alpha \in I^* \, (out_T(t_1,\alpha) \subseteq out_S(s_1,\alpha))$$

- **FSMs *T* and *S* are *non-separable* if**

  $$\forall \alpha \in I^* \, (out_T(t_1,\alpha) \cap out_S(s_1,\alpha) \neq \varnothing)$$

- **FSMs *T* and *S* are *r-compatible if T* and *S* have a common reduction**

separability ⇨ non-reduction ⇨ *r*-distingushability ⇨ non-equivalence

TRADITIONAL FAULT MODEL

$$< S,\ \sim,\ FD >$$

specification

conformance relation

fault domain

An implementation FSM *Imp* is *conforming* if *Imp* $\sim$ *S*

The objective of MBT testing – to detect each nonconforming *Imp* $\in$ *FD*

'ALL WEATHER CONDITIONS' ASSUMPTION

holds

does not hold

+ there exist test suite derivation methods
- sometimes this assumption is not realistic
- each test case is applied several times

+ realistic
+ each test case is applied only once
- existing test derivation methods do not guarantee the fault coverage

DETECTING NONCONFORMING IUT WHEN 'ALL WEATHER CONDITIONS' ASSUMPTION DOES NOT HOLD

- **Preset tests (experiments)**

Guarantee to detect an implementation that is separable with the specification

- **Adaptive tests (experiments)**

Guarantee to detect an implementation that is *r*-distinguishable with the specification

NON-SEPARABILITY RELATION

Complete FSMs T and S are *non-separable* if for each input sequence α

$$[out_T(t_0, \ \alpha) = out_S(s_0, \ \alpha)]$$

If there exists an input sequence α $[out_T(t_0, \ \alpha) \neq out_S(s_0, \ \alpha)]$, then FSMs T and S are *separable*

α - a *separating* sequence of FSMs T and S

TESTING W.R.T. THE NON-SEPARABILITY RELATION

- Fault model <*S*, ~, *FD*> where

S - the specification FSM

~ - the non-separability relation

FD – the fault domain

All FSMs are complete

EXHAUSTIVE AND SOUND TESTS

- A *test suite* - a finite set of finite input sequences

- A test suite is *exhaustive* w.r.t. $<S, \sim, FD>$ if the test detects each $Imp \in FD$ that is separable with S

- A test suite is *sound* w.r.t. $<S, \sim, FD>$ if each $Imp \in FD$ that is non-separable with S *passes* the test suite

- A test suite is *complete* w.r.t. $<S, \sim, FD>$ if the test suite is sound and complete

We derive an exhaustive test suite

FAULT MODELS

Fault models

$\boldsymbol{<S, \sim, Sub_{nd}(MM)>}$

and

$\boldsymbol{<S, \sim, \Re_m>}$

have finite fault domains

$\Downarrow$

A test suite can be derived by explicit enumeration

INTERSECTION OF FSMs

Intersection $S \cap T$ is the largest connected submachine of the FSM

$<S \times T, I, O, h, s_1t_1>$ where

$$(st, i, o, s't') \in h$$

$$\Leftrightarrow$$

$$(s, i, o, s') \in h_S \;\&\; (t, i, o, t') \in h_T$$

DERIVING A SEPARATING SEQUENCE OF TWO FSMs (1)

Input: Complete FSMs *S* and *T*

Output: A shortest separating sequence of FSMs *S* and *T* (if it exists)

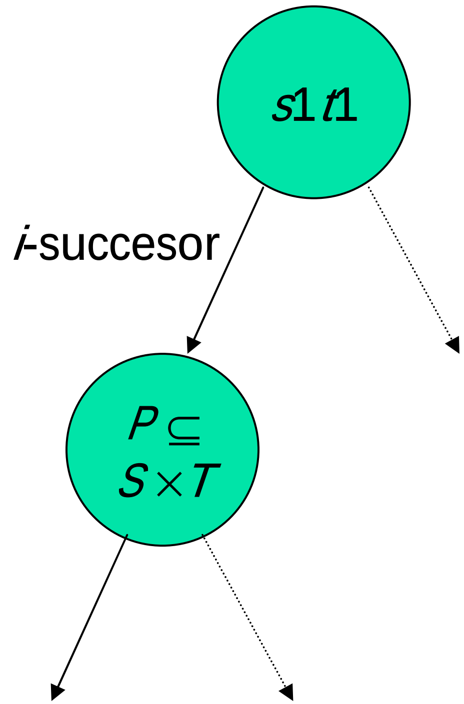
Step 1. Derive the intersection $S \cap T$

If the intersection is complete then the FSMs *S* and *T* are non-separable

DERIVING A SEPARATING SEQUENCE OF TWO FSMs (2)

Step 2. Derive a truncated successor tree of the intersection $S \cap T$

Successor tree of $S \cap T$



TERMINATION RULES FOR A NODE WITH LABEL *P*

Termination rule 1

There exists an input i s.t. for each state of the set P the transition under i is undefined in the intersection $S \cap T$

successfully separated

Termination rule 2

There exists a node at a jth level, $j < k$, labeled with subset $R \subseteq P$

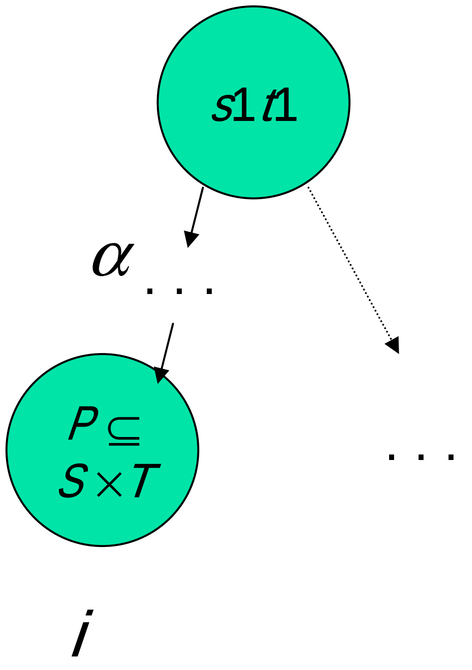
a shortest separating sequence cannot be derived using this path

DERIVING A SEPARATING SEQUENCE OF TWO FSMs (3)

Let there be a path labeled with α to a leaf node labeled with the subset P where a transition under i is undefined for each state of P

Then αi is a shortest separating sequence of S and T

Successor tree of $S \cap T$



UPPER BOUND ON SEPARATING SEQUENCE LENGTH

- Given FSMs S with n states and T with m states, the length of a shortest separating sequence is at most 2^{nm-1}
- The upper bound is reachable but possibly only for exponential number of inputs
- Never experimentally reached the upper bound

DERIVING A COMPLETE TEST SUITE W.R.T. $< S, \sim, Sub_{nd}(MM) >$ (1)

Input: FSMs S and MM

Output: A complete test suite TS w.r.t. $< S, \sim, Sub_{nd}(MM) >$

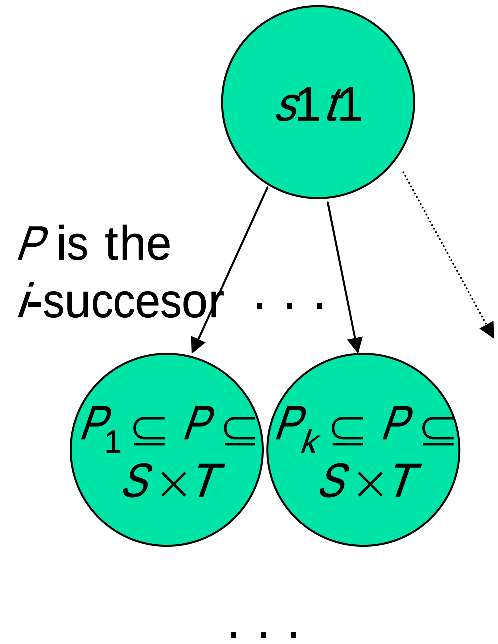
Step 1. Derive the intersection $S \cap MM$

DERIVING A COMPLETE TEST SUITE W.R.T. < S, ~, $Sub_{nd}(MM)$ > (2)

Step 2. Derive a truncated successor tree of the intersection $S \cap MM$

There is an outgoing edge from a non-leaf node labeled with i to the nodes labeled with **each non-empty subset** of the i-successor of the subset P

Successor tree of $S \cap MM$



TERMINATION RULES FOR THE NODE WITH LABEL *P*

Termination rule 1

There exists an input i s.t. for each state of the set P the transition under i is undefined in the intersection $S \cap T$

Termination rule 2

There exists a node of this path labeled with subset $R \subseteq P$

Termination rule 3

$P \supseteq \{(s, t)\}$ and the already derived part of the tree has a node labeled with $\{(s, t)\}$

and

an input sequence α that labels the path from the root to this node does not label any other path in the tree

DERIVING A COMPLETE TEST SUITE W.R.T. < S, ~, $Sub_{nd}(MM)$ > (3)

- For each path terminated using Rule 1, include into *TS* an input sequence which labels the path appended with an input *i* s.t. a transition from each state of *P* under *i* is undefined in the intersection $S \cap MM$
- For each path of the tree terminated using Rule 2 or Rule 3, include into *TS* the longest prefix of an input sequence that labels the path, with the following property:

For the tail input *i* that labels the edge from the node labeled with the set *P*, there exists state $(s,m) \in P$ such that $out_{S \cap MM}((s,m), i) \subset out_{MM}(m, i)$

Example

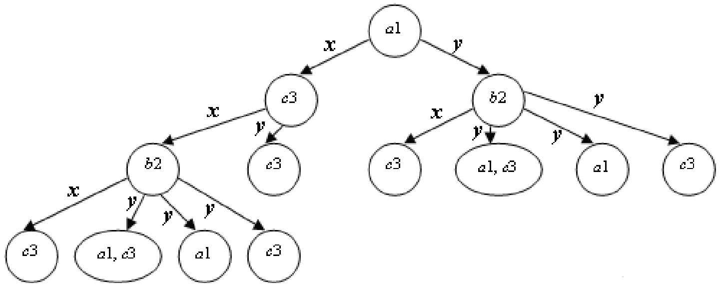
S	a	b	c	MM	1	2	3
x	c/1	c/0	b/0	x	3/1	3/0	2/0
y	b/0	a/0; c/1	c/1	y	2/0	1/0; 3/1	3/1; 1/0

FSMs *S* and *MM*

S∩M M	a1	c3	b2
x	c3/1	b2/0	c3/0
y	b2/0	c3/1	a1/0; c3/1

FSM *S*∩ *MM*

Tree of the previous exhaustive test



xy also is an exhaustive test suite w.r.t. *MM*

DERIVING A COMPLETE TEST SUITE w.r.t. $<S, \sim, \Re_m>$

Theorem. Given the specification FSM S with n states a test suite which contains each input sequence of length $2^{mn}-1$ is complete w.r.t. $<S, \sim, \Re_m>$

STATE COUNTING ALGORITHM (1)

Main ideas:

- to derive a truncated tree of S
- to terminate a path when we are sure that for each FSM T with m states the input sequence that labels the path traverses two equal subsets of the intersection $S \cap T$

STATE COUNTING ALGORITHM (2)

A current node labeled with the subset *P* of states of *S* is claimed as a leaf node if the path from the root to this node has $2^{|P| \cdot m}$ nodes labeled with subsets of *P*

MODIFICATION

The node *Current* at k^{th} level labeled with subset P is a leaf if:

1. A subset $K \subseteq P$ can be represented as the union of subsets P_j
2. For each P_j the prefix of the path to the node at the lth level, $l < k$, has $(2^{|P_j|\cdot m}-1)$ nodes labeled with the set P_j,
3. The suffix of the path from the node at the lth level to the node *Current* has $(2^{(|P|-|K|)\cdot m})$ nodes labeled with subsets of the set P

EXAMPLE (1)

- P = {1,2,3,4,5}, M = {a, b}

The number of different non-empty sets of the $P \times M$ = 2^{10} - 1

- K = {1,2} $\cup$ {3,4}

The number of different non-empty subsets of {1,2} $\times M$ = $2^4 - 1 = 15$

- The only sets which do not include non-empty subsets of {1,2} $\times M$ and {3,4} $\times M$ are subsets of {5} $\times M$

...

$\downarrow$

non-empty subsets of {1,2} $\times M$

...

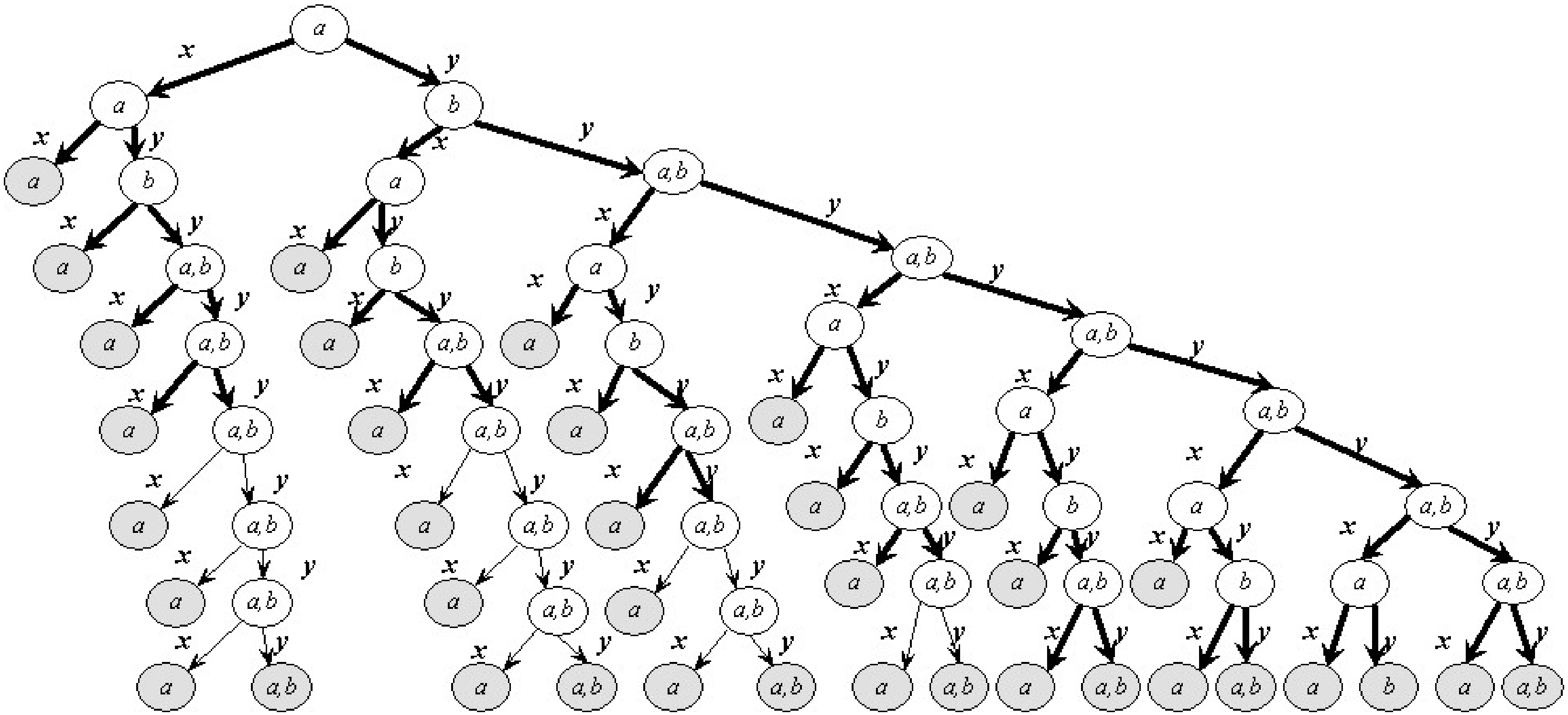
$\downarrow$

non-empty subsets of {1,2} $\times M$ (l^{th} level)

$\downarrow$

non-empty subsets of {5} $\times M$ (k^{th} level, $k > l$)

EXAMPLE (2)



EXPERIMENTAL RESULTS

- On average, tests w.r.t. a mutation machine *MM* using modified algorithm are twice shorter and this gain increases when *MM* is more deterministic
- On average, tests w.r.t. a “black box” are 1.5 times shorter

More rigorous analysis is needed to shorten tests w.r.t. a “black box”

FUTURE WORK

- Both algorithms do not return a shortest test suite w.r.t. $<S, \sim, Sub_{nd}(MM)>$ and w.r.t. $<S, \sim, \Re_m>$
- Adaptive tests using the *r*-distinguishability relation could be shorter
- More rigorous experimental results could be interesting

Thanks for your attention!