University of Campinas - UNICAMP
Institute of Computing - IC
São Paulo - Brazil

A new Method for Incremental Testing of Finite State Machines [1]

Lehilton Lelis Chaves Pedrosa
lehilton.pedrosa@students.ic.unicamp.br

Arnaldo Vieira Moura
arnaldo@ic.unicamp.br

15-april-2010

[1] Supported by FAPESP

Testing of Finite State Machines (FSM)

Why test using FSMs?

- Black-box testing
- Detect flaws in specification
- Formal verification of a system's implementations
- May model programs, protocols, hardware, ...

In the literature

- The W-method (Chow 1978)
- Several derivations: Wp-method, HSI-method, G-method, ...

FSMs

Definition

Formally, a FSM M is a tuple

$$M = (X, Y, S, s_0, \delta, \lambda), \quad \text{where}$$

- X is the input alphabet,
- Y is the output alphabet,
- S is the set of states,
- $s_0 \in S$ is the initial state,
- $\delta : X \times S \to S$ is the state transition function,
- $\lambda : X \times S \to Y$ is the output function.

Example

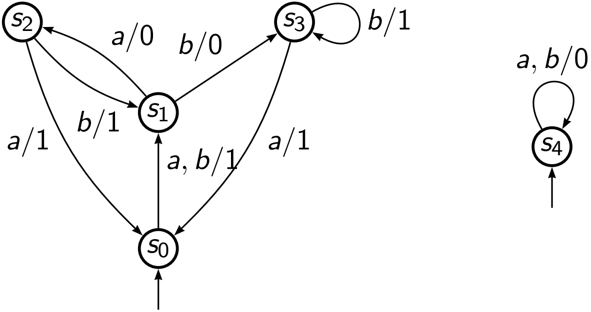


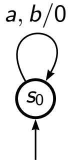
Figure: Finite state machines

Comparing FSMs

State equivalence

Given: M (specification) and M' (implementation):

- $s \approx_\rho s'$ if $\widehat{\lambda}(\rho, s) = \widehat{\lambda}'(\rho, s')$
- $s \approx_R s'$ if $\widehat{\lambda}(\rho, s) = \widehat{\lambda}'(\rho, s')$, $\forall \rho \in R$
- $s \approx s'$ if $\widehat{\lambda}(\rho, s) = \widehat{\lambda}'(\rho, s')$, $\forall \rho \in X^*$

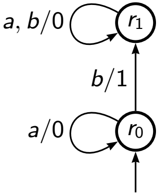


Target: find a *set* π such that

$$s_0 \approx s_0' \quad \text{iff} \quad s_0 \approx_\pi s_0'$$

Examples:

- $s_0 \approx_a r_0$
- $s_0 \not\approx_b r_0$
- $s_0 \approx r_1$



Review - W-method

Definitions

- n is the number of states in S
- m is an estimated upper bound to the number of states in S',
- P is a cover set of S: *for each* $s \in S$, *there exists* $\rho \in P$, *with* $\widehat{\delta}(\rho, s_0) = s$, *and* $\rho a \in P$, *for all* $a \in X$
- W is a characterization set of S: *for each pair* $r, s \in S$, *there exists* $\rho \in W$, *with* $r \not\approx_\rho s$

Test suite: $\pi = PZ$

- $Z = X_{m-n}W$
- X_{m-n} is the set of input words with length up to $m - n$

Incremental testing

Why incremental testing?

- Current methods are monolithic
- Generated test suites are exponential
  (depends on the number of states)
- Testing FSMs with a large number of states is impractical
- Retesting modified systems

Main idea

- Break a system model into a set of subsystems
- Test each subsystem (*submachine*) independently
- Test the integrated system (*combined FSM*) at low cost

Combined FSM - Definitions

Submachine of M: $\dot{N} = (\dot{X}, \dot{Y}, \dot{S}, \dot{s}_0, \dot{\delta}, \dot{\lambda})$

- $\dot{X} = X, \quad \dot{Y} \subset Y, \quad \dot{S} \subset S$
- $\dot{\delta}(a, s) = \delta(a, s), \quad \dot{\lambda}(a, s) = \lambda(a, s), \quad$ for every $a \in X$

N-combined FSM

- N is a set of submachines of M
- S_N is the set of all submachine states
- $S_M = S \setminus S_N$ is the set of additional states
- I_N as the set of all submachines entry points

Combined FSM - Example

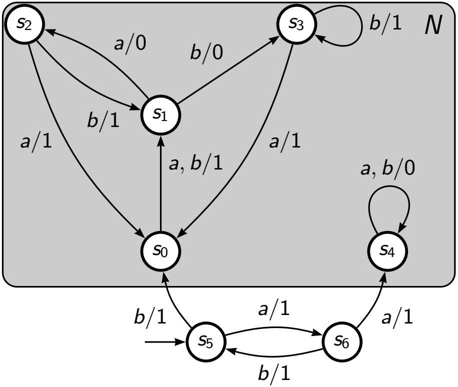


Figure: A combined FSM

- 7 states in total
- only 2 *additional* states

- $S_N = \{s_0, s_1, s_2, s_3, s_4\}$
- $I_N = \{s_0, s_4\}$
- $S_M = \{s_5, s_6\}$

Testing combined FSMs

Testing new combined machines

- Each submachine is implemented and tested previously
- Only additional states of the combined FSM need to be tested
- We may test FSMs with a large number of states

Retesting modified specifications

- Only affected submachines need to be retested
- Different combined FSM may share one submachine implementation

A new testing method

The C-method

- Based on the W-method and on the G-method
- Assumes that the specification and the implementation are combined FSMs

Introduction of new concepts

- Cover sets ⇒ Partial cover sets:
  - cover only a subset of states
  - used to test only additional states
- Characterization sets ⇒ Separator:
  - no need to distinguish every pair of states
  - generalizes the characterization sets

C-method - Concepts

Preliminary concepts: basic notion

- A set of input words R is an (A, B)-separator:
  R can distinguish states of A and B
- Neighborhood $\mathrm{nbh}(C, d)$:
  states reached from states C using at most d input symbols
- Partition $[C \diagup R]$:
  the set of equivalence classes induced by $\approx_R$ over states C
- Relative concatenation $A \underset{s}{\otimes} \mathcal{B}$:
  concatenates sequences according to the state reached from s

C-method - The algorithm

Input:

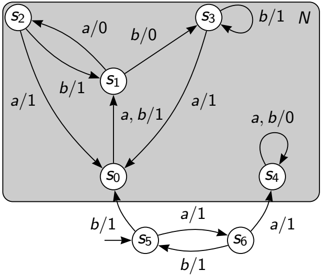
- M: a N-combined FSM
- m: bound on number of implementation aditional states ($S'_M \leq m$)

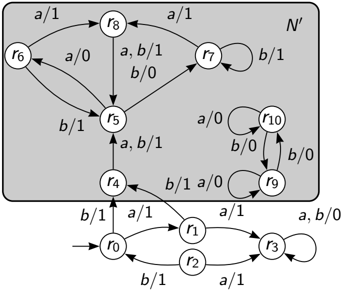
Algorithm

1. $P \longleftarrow$ obtain a partial cover set P for S_M
2. $R \longleftarrow$ obtain a $(S_M \cup I_N, S_N)$-separator
3. $n \longleftarrow |[S_M \diagup R]|$
4. $A \longleftarrow \mathrm{nbh}(I_N, m-n-1)$
5. $T \longleftarrow$ obtain a (A, S_N)-separator
6. $\mathcal{R}(S_M) \longleftarrow R, \quad \mathcal{R}(S_N) \longleftarrow R \cup T$
7. For each $s \in S, \quad \mathcal{Z}(s) \longleftarrow X_{m-n} \bigotimes_s \mathcal{R}$
8. Return $\pi \longleftarrow P \otimes \mathcal{Z}$

Example

Specification and a candidate implementation





Notes

- Specification M is a N-combined FSM
- A candidate implementation M' is any N'-combined FSM
- An implementation is a black-box, but bounds on the number of states may be estimated: $S'_N \leq 7, \;\; S'_M \leq 4$

Example - comparison with the W-method

W-method

- Characterization set
  $W = \{aaaa, bb\}$

C-method

- Smaller separators
  $R = T = \{aaaa\}$

Example - comparison with the W-method

W-method

- Characterization set
  $W = \{aaaa, bb\}$
- Complete cover set P:
  test all states

C-method

- Smaller separators
  $R = T = \{aaaa\}$
- Partial cover set P:
  test only additional states

Example - comparison with the W-method

W-method

- Characterization set
  $W = \{aaaa, bb\}$
- Complete cover set P:
  test all states
- Parameters $m = 11$, $n = 7$

C-method

- Smaller separators
  $R = T = \{aaaa\}$
- Partial cover set P:
  test only additional states
- Parameters $m = 4$, $n = 2$

Example - comparison with the W-method

W-method

- Characterization set
  $W = \{aaaa, bb\}$
- Complete cover set P:
  test all states
- Parameters $m = 11$, $n = 7$
- Calculates set X_{11-7},
  with 31 words

C-method

- Smaller separators
  $R = T = \{aaaa\}$
- Partial cover set P:
  test only additional states
- Parameters $m = 4$, $n = 2$
- Calculates set X_{4-2},
  with 7 words

Example - comparison with the W-method

W-method

- Characterization set
  $W = \{aaaa, bb\}$
- Complete cover set P:
  test all states
- Parameters $m = 11$, $n = 7$
- Calculates set X_{11-7},
  with 31 words
- Test suite $\pi = PZ$,
  with 256 prefix-free words

C-method

- Smaller separators
  $R = T = \{aaaa\}$
- Partial cover set P:
  test only additional states
- Parameters $m = 4$, $n = 2$
- Calculates set X_{4-2},
  with 7 words
- Test suite $\pi = P \otimes \mathcal{Z}$,
  with 20 prefix-free words

Example - comparison with the W-method

W-method

- Characterization set
  $W = \{aaaa, bb\}$
- Complete cover set P:
  test all states
- Parameters $m = 11$, $n = 7$
- Calculates set X_{11-7},
  with 31 words
- Test suite $\pi = PZ$,
  with 256 prefix-free words

C-method

- Smaller separators
  $R = T = \{aaaa\}$
- Partial cover set P:
  test only additional states
- Parameters $m = 4$, $n = 2$
- Calculates set X_{4-2},
  with 7 words
- Test suite $\pi = P \otimes \mathcal{Z}$,
  with 20 prefix-free words
- Additionally, the submachines
  may be tested with 24 test cases,
  for a total of 44 words

Comparison with W-method - Results

Suppose

- $|S_M| = \ell, \quad |S_N| = j, \quad |S'_M| = m, \quad |S'_N| = k$
- P_W, P_C: complete and partial cover sets, respectively
- π_W, π_C: test suites generated by W-method and C-method, respectively

Then

1. $\frac{|P_W|}{|P_C|} \geq 1 + \frac{|X|}{|X|+1}\frac{j}{\ell}$
2. $|\pi_C| \in O(l(j+\ell)^2|X|^{m-\ell+1})$
3. $|\pi_W| \in O((j+\ell)^3|X|^{m-\ell+k-j+1})$

Conclusion

Final remarks

- We introduced a new method to test FSMs
- The C-method may be used for:
  - incremental testing of FSMs
  - retesting modified systems with previously working implementation
- The results indicate that the C-method is scalable: it is possible to test combined FSM with a large number of states

Future works

- Extend the C-method to nondeterministic and partially specified FSMs.

Questions...

Lehilton Pedrosa (lehilton@gmail.com)

Arnaldo Moura (arnaldo@ic.unicamp.br)

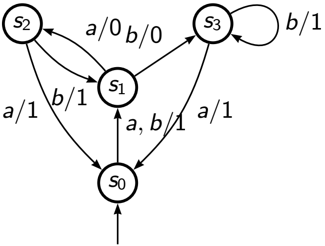
Basic concepts

Extended functions

- transition function $\widehat{\delta} : X^* \times S \to S$
- output function $\widehat{\lambda} : X^* \times S \to Y^*$

Examples:

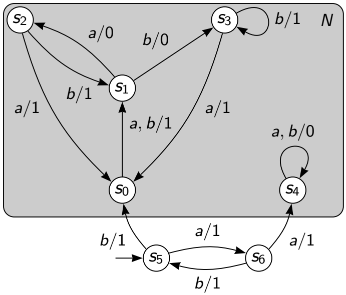
- $\widehat{\delta}(aabb, s_0) = s_3$
- $\widehat{\lambda}(aabb, s_0) = 1010$



C-method - Separators

Separators

- $R \subset X^*, \quad A, B \subset S$
- R is a (A, B)-separator iff for every $r \in A$, $s \in B$, such that $r \not\approx s$, we have $s \not\approx_R r$.



Examples:

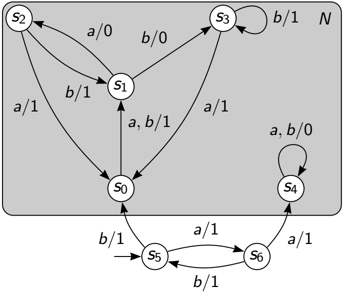
- $S_N = \{s_0, s_1, s_2, s_3, s_4\}$
- $S_M = \{s_5, s_6\}$
- $R = \{aaaa\}$

- R is a (S_N, S_M)-separator
- R is not a (S_N, S_N)-separator

C-method - Neighborhood

Neighborhood of states

- An auxiliary concept
- notation $\mathrm{nbh}(C, d)$:
  states that can be reached from a state of C through a word with length of at most d



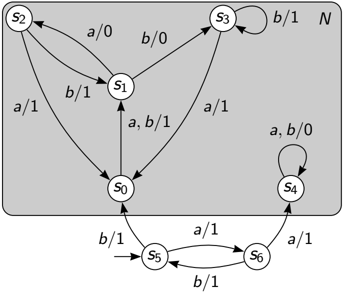
Example:

- $I_N = \{s_0, s_4\}$
- $\mathrm{nbh}(I_N, 1) = \{s_0, s_1, s_4\}$

C-method - Induced partitions

Partitions induced by a set of words

- The equivalence relation $\approx_R$ induces partition over a set of states C
- notation $[C \diagup R]$: *the set of equivalence classes induced by* $\approx_R$ *over* C



Examples:

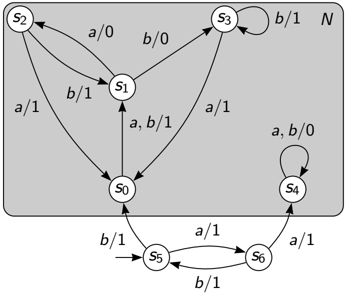
- $S_N = \{s_0, s_1, s_2, s_3, s_4\}$
- $S_M = \{s_5, s_6\}$
- $R = \{aaaa\}$

- $[S_N \diagup R] = \{\{s_0\}, \{s_1\}, \{s_2, s_3\}, \{s_4\}\}$
- $[S_M \diagup R] = \{\{s_5\}, \{s_6\}\}$

C-method - Relative concatenation

Relative concatenation

- Each state (or subset of states) is tested with different test cases
- State attribution: $\mathcal{B} : S \to \mathcal{P}(X^*)$
- notation
  $A \underset{s}{\otimes} \mathcal{B} = \{\alpha\beta | \alpha \in A, \beta \in \mathcal{B}(\widehat{\delta}(\alpha, s))\}$



Examples:

- $S_N = \{s_0, s_1, s_2, s_3, s_4\}$
- $S_M = \{s_5, s_6\}$
- $\mathcal{R}(S_N) = \{a\}, \quad \mathcal{R}(S_M) = \{b\}$
- $A = \{a, b, bb\}$
- $A \otimes \mathcal{R} = \{ab, ba, bba\}$

Comparison example - test suite construction

Using W-method

- $|S'| \leq m = 11$
- $W = \{aaaa, bb\}$
- $n = |S| = 7$
- $P = \{\varepsilon a, b, aa, ab, aaa, aab, ba, bb, baa, bab, baaa, baab, baba, babb\}$
- $Z = X_{11-7} W$
- $\pi = PZ$
  π contains 256 prefix-free words

Using C-method

- $|S'_M| \leq m = 4$
- $R = \{aaaa\} \quad (S_M \cup I_N, S_N)$*-separator*
- $n = |[S_M / R]| = 2$
- $P = \{\varepsilon, a, b, aa, ab, aaa, aab, ba, bb\}$
- $A = \mathrm{nbh}(I_N, 4-2-1) = \{s_0, s_1, s_4\}$
- $T = \{aaaa\} \quad (A, S_N)$*-separator*
- $\mathcal{R}(S_N) = \mathcal{R}(S_M) = R$
- $\mathcal{Z}(s) = X_{4-2} R$, for every $s \in S$
- $\pi = P \otimes \mathcal{Z}$
  π contains 20 prefix-free words
- Additionally, we may test the submachines with 24 test cases, for a total of 44 words