Slicing of State-Based Models

Bogdan Korel, Inderdeep Singh, Luay Tahat, Boris Vaysburg

ICSM'03

2006. 3. 7
Park, Seung-Hun
Contents

- Introduction
- EFSM modeling
- EFSM dependence graph
- EFSM slicing
  - Deterministic EFSM slicing
  - Non-deterministic EFSM slicing
- EFSM slicing tool
- Conclusion
- Discussion
Introduction (1/2)

- System modeling
  - Reduces ambiguity, misunderstanding, and misinterpretation of system specifications
  - Modern systems tend to be very large and complex
    - Hard to understand and difficult to modify and debug
    => Need to analyze a model with respect to particular system functionality only

- Slicing
  - Reduction technique to identify all statements that may affect the value of a variable of interest at some point in the program

```plaintext
begin
  read(X,Y);
  Total := 0.0;
  SUM := 0.0;
  if X <= 1 then
    SUM := Y;
  else
    begin
      read(Z);
      Total := X * Y;
      end;
  end;
  end
```

Slicing criteria (Total, 14)
Introduction (2/2)

- Approach of slicing state-based models
  - Focuses on slicing Extended Finite State Machine (EFSM) models
  - Automatically identifies the parts of the model that affect an element of interest using EFSM dependence analysis
  - Uses slice reduction techniques to reduce the size of the EFSM slice
  - Develops an EFSM slicing tool
EFSM modeling (1/2)

- Extended finite state machine (EFSM)
  - Combines the FSM and programming language approaches to specification
  - Transition is associated with an event, a condition, and a sequence of actions
  - Each event or action can have certain parameters
    - Three types of action
      - Input action, output action, assignment action
  - Each condition has syntax of predicates of conditional statements in C language
  - Input to an EFSM
    - A sequence of events with input values associated with these events

```
PIN(p)
[p == pin] /
Write("Select a language")
```
EFSM modeling (2/2)

Example: ATM system
EFSM dependence graph (1/4)

Notations

- $S_b(T)$: a state from which $T$ is outgoing
- $S_e(T)$: a state to which $T$ is incoming
- $U(T)$: a set of variables used in a condition and actions of $T$
- $D(T)$: a set of variables defined by actions and variables defined in the event of $T$ and not redefined by any action of $T$
- $C(T)$: an enabling condition associated with transition $T$
- $E(T)$: an event associated with transition $T$

U(T2) = \{p, pin, attempts\}
D(T2) = \{p, attempts\}
C(T2) = (p != pin) and (attempts < 3)
E(T2) = PIN(p)
Change impact analysis to support architectural evolution

EFSM dependence graph (2/4)

Data dependence
- Captures the notion that one transition defines a value to a variable and another transition may potentially use this value
- Data dependence bet. T1 and T2 with respect to variable v
  - \( v \in D(T1), \ v \in U(T2) \)
  - There exists a path in the EFSM model from T1 to T2 along which v is not modified
Control dependence

- Captures the notion that one transition may affect traversal of another transition
- Control dependence bet. T1 and T2
  - State $S_b(T2)$ does not post-dominate state $S_b(T1)$
  - State $S_b(T2)$ post-dominate transition T1

EFSM dependence graph (3/4)
Partial EFSM dependence graph for ATM system
Deterministic EFSM slicing (1/2)

- Slicing criterion

EFSM model M

- Input x

EFSM model M'

- Behavior preservation
Deterministic EFSM slicing (2/2)

Slicing algorithm

Start
T1
Card(pin, sb, cb)
write("Enter PIN")
| attempts = 0

S1

PIN(p)
| p == pin/
write("Select a Language English/Spanish")
T2

T4

T5

T17

T15

T7

T9

T10

T16

T14

T8

T23

T21

S1

S2

S3

S4

S5

S6

S7

T18

T19

T20

T22

Slicing algorithm

Non-contributing transition

contributing

Deterministic EFSM slicing (2/2)

Slicing algorithm

Non-contributing transition

contributing
Non-deterministic EFSM slicing (1/4)

- Non-deterministic EFSM slice
  - Reduce the slice by merging states
- Slicing criterion

\[ \text{EFSM model } M \xrightarrow{\text{Ti}(v)} \text{EFSM model } M' \]

\[ \text{Input } x \xrightarrow{\text{equal}} \text{Input } x' \]

\[ \text{Input } x \xrightarrow{\text{events}} \text{Input } x' \]
Non-deterministic EFSM slicing (2/4)

- Traversability property problem
  - Some traversals in the non-deterministic slices may be not possible in the original EFSM

Non-deterministic EFSM slice

Part of original EFSM
Non-deterministic EFSM slicing (3/4)

Rules for satisfying traversability property

- Rule 1

\[ S_i \rightarrow S_k \rightarrow S_i, S_k \]

- Rule 2

\[ S_j \leftrightarrow S_i, S_k \]
Non-deterministic EFSM slicing (4/4)

Non-deterministic slicing algorithm
**EFSM slicing tool**

- Consists of graphical editor, an EFSM executor, and an EFSM slicer
- Snapshot
Conclusion

- Approach of slicing EFSM models
  - Proposed deterministic slicing and non-deterministic slicing
    - Help in understanding how the existing system will interact with the changes
    - Facilitate in isolating the parts of the model that may contribute to faulty behavior
    - Used for test generation purposes

- Future work
  - Investigate slice minimization
  - Investigate EFSM forward slicing and dynamic model slicing
Discussion

Critique

- It needs to prove that the proposed rules preserve the traversability property
- Usage of the rules is different from the rule definition

Ex)

a. By applying reduction Rule 1, states S_2 and S_3 are merged into state “S_2,S_3”.
b. In step 10, transitions T_5 and T_6 are removed from the model.
c. By applying reduction Rule 1, states S_4 and S_5 are merged into state “S_4,S_5”.
d. In step 10, transitions T_{11}, T_{12}, T_{13}, T_{14}, T_{15}, and T_{16} are removed from the model.
e. By applying reduction Rule 1, states “S_2,S_3” and “S_4,S_5” are merged into state “S_2,S_3,S_4,S_5”.
f. In step 10, transitions T_7 and T_9 are removed from the model.
g. By applying reduction Rule 2, states S_6 and S_7 are merged into state “S_6,S_7”.

2006-04-07