Detection method developed after a thorough study of the contest held in 1998

Ahmed Khoumsi Rodolphe Justin Bevelo

Department of Electrical and Computer Engineering

University of Sherbrooke, CANADA

Plan

1. Introduction : objective and motivation
2. Detection method developed previously
3. Contest organized in 1998
4. Transforming a Chisel specification into our model
5. About scenario selection
6. First category of interactions and their detection
7. Second category of interactions and their detection
8. Conclusion and future work

Introduction

Objective : to improve a detection method presented in FIW'97

Motivation : The detection method to be improved has following limitations :

- in most cases, only suspected interactions may be detected
- several interactions of the contest of 1998 are not detected even when they are known and suspected

Previous detection method

Feature description

Feature described by extended finite state automata (EFSA)

Transition of EFSA defined by : $\bigcirc \xrightarrow{\sigma,\ E,\ T} \bigcirc$

E = Enabling condition = function of variables

T = Transformation of variables

σ = event

σ occurs only if E = true

After occurrence of σ, T is applied

Examples of variables : $free_X$ $com_{X,Y}$ $subs_X^F$

Example of EC : $free_A \wedge (free_B \vee subs_X^{CW})$

Example of T : $\overline{free_A}$ & $\overline{free_B}$ & $com_{A,B}$

Previous detection method

Feature combination and interaction description

Operator *Comb* : Let F1 and F2 be two EFSAs describing two features

F = *Comb*(F1,F2) is a EFSA describing combination of the two features

Interaction description :

Given Property *P* not satisfied in F = *Comb*(F1,F2)

Requirements : Let V1 and V2 be sets of variables of F1 and F2

$V1 \cap V2 = \varnothing$

Property *P* expressed using :

- variables of *both* F1 and F2
- operators = (EQUAL) $\wedge$ (AND) $\vee$ (OR) $\neg$(NOT)

Previous detection method

Interaction detection

Steps of detection : Let F1 and F2 be two features to be combined

1. Express desired property *P*

2. Compute F = *Comb*(F1,F2)

3. Transform the EFSA F into an equivalent FSA f

4. Apply a model-checking technique to verify whether *P* holds in f

Interaction detected when *P* does not hold in at least one state of f

Contest organized in 1998

Features of the contest

Call Forwarding Busy Line (CFBL)	Three-Way Calling (TWC)
Calling Number Delivery (CND)	IN Call Forwarding (INCF)
IN Freephone Billing (INFB)	Call Waiting (CW)
IN Freephone Routing (INFR)	IN Charge Call (INCC)
IN Teen Line (INTL)	Return Call (RC)
Terminating Call Screening (TCS)	Cellular (CELL)

Contest organized in 1998

Interactions determined by the organizers of the contest

(Table 1)

Transforming a Chisel specification into our FSA model

Motivation and approach

In the present study, we describe every feature directly by FSA (denoted v-FSA) instead of describing it by EFSA and then transforming EFSA into FSA.

Motivation :

- Operator *Comb* becomes much simpler
- Translating a Chisel diagram into v-FSA is relatively simple

Approach of transformation :

- Each node *N* in a Chisel corresponds to a transition *N* of the FSA
- The set of outgoing arrows *A1, A2, …, An* from node *N* correspond to a state *S* of the FSA, which is reached by transition *N*
- Nodes reached by *A1, A2, …, An* correspond to the outgoing transitions of *S*

The twelve features of the contest have been described by v-FSAs.

Transforming a Chisel specification into a v-FSA

Examples

(Figure 1)

(Figure 2)

Ahmed Khoumsi and **Rodolphe Bevelo**

About scenario selection

Each feature specification is parameterized by the participants and their initial status

Example : specifications of previous page are parameterized by A and B, where A is initially idle

In the contest, each parameterization is illustrated by a Table which specifies for each participant, the features assignments, roles, and initial status

Example : (TWC-CW : 3) and (TWC-CW : 4)

(Table 4)

In general, a given parameterization of a feature F (or two combined F1 and F2) integrates several scenarios of F (or *Comb*(F1,F2))

Interesting thing : one or several scenarios (a priori unknown) may be automatically generated by a given parameterization

First category of interactions and their detection

Properties of first category

Let F1 and F2 be two features

Let *P* be a property of F1, i.e :

- *P* holds in F1 when F1 is used alone, **and**
- *P* depends on variables of V1

We consider the following two cases :

(1) *P* depends on variables in $V1 \cap V2$ $(\neq \emptyset)$

(2) *P* depends : - on constant variables in $V1 \setminus V2$, **and**
- possibly on variables in $V1 \cap V2$

Case (2) can be transformed into Case (1) if we add constant variables of $V1 \setminus V2$ into $V2$

Therefore, for simplicity we consider only case (1)

First category of interactions and their detection

Two propositions

Let *P* be a property of F1 of First category

Proposition 1 : If *P* does not hold in a minimal behaviour of F2 then :

at least one property of Second category does not hold in *Comb*(F1,F2)

Proposition 2 : If *P* holds in F2 then :

- *P* holds in Comb(F1,F2), **or**
- at least one property of Second category does not hold in *Comb*(F1,F2)

Proposition 1 implies that if *P* does not hold in a minimal behaviour of F2, then an interaction exists in *Comb*(F1,F2). Therefore, it is useless to compute *Comb*(F1,F2) and check the existence of interaction

Proposition 2 implies that if *P* holds in F2, then it is useless to check whether *P* holds in *Comb*(F1,F2)

First category of interactions and their detection

Interactions of the contest of 1998 : First category, Case 1

(Table 5)

First category of interactions and their detection

Interactions of the contest of 1998 : First category, Case 2

(Table 6)

Ahmed Khoumsi and **Rodolphe Bevelo**

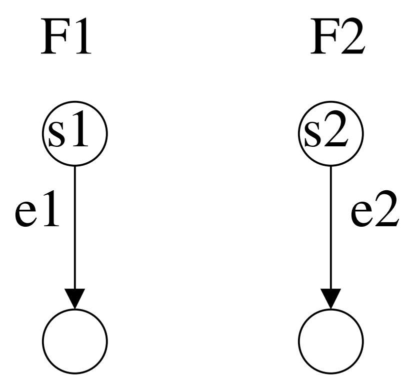
Second category of interactions and their detection

Incompatible events

Interactions of second category :

- Necessitate computation of *Comb*(F1,F2)
- To be detected only when no interaction of first category has been detected

Incompatible events :



Two incompatible events must not be allowed at the same time. Therefore :

If e1 and e2 are incompatible

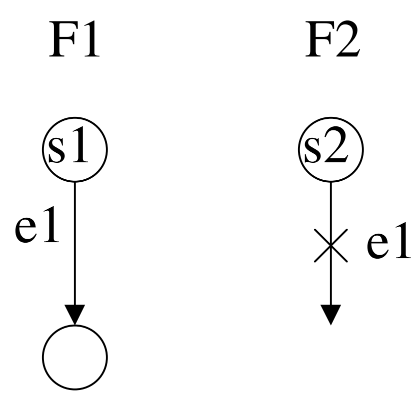
Then (s1,s2) must not be reachable in *Comb*(F1,F2)

Second category of interactions and their detection

Indelayable event

Undelayable event :

When an undelayable event is allowed in one feature, its occurrence cannot be delayed by the other feature to a future time.



If e1 is undelayable, and F1 and F2 are synchronized on e1

Then the state (s1,s2) must not be reachable in *Comb*(F1,F2)

Ahmed Khoumsi and **Rodolphe Bevelo**

Second category of interactions and their detection

Properties of second category

P1 : Property of termination of minimal behaviour

P2 : Property of events compatibility

P3 : Property of event delayability

P4 : Property of variable consistency

P5 : Property of a feature Fi (not of First category)

Second category of interactions and their detection

Interactions of the contest of 1998 : Second category

Property P1 not satisfied

(Table 7)

Second category of interactions and their detection

Interactions of the contest of 1998 : Second category

Property P2 not satisfied

(Table 8)

Ahmed Khoumsi and **Rodolphe Bevelo**

Second category of interactions and their detection

Interactions of the contest of 1998 : Second category

Property P3 not satisfied

(Table 9)

Second category of interactions and their detection

Interactions of the contest of 1998 : Second category

Property P4 not satisfied

(Table 10)

Conclusion

Contributions in comparison to our previous method

- Use of a model which facilitates translation from Chisel

- Problem of scenario selection transformed into a problem of parameterization

- Categorization of interactions :
  interactions of first catgory detected without computing *Comb*(F1,F2)

- Properties to be checked are general and do not depend necessarily on suspected interactions

Conclusion

Future work

- How to parameterize a pair of features

- To evaluate the detection method by applying it to other features
  Example : features of the contest held in 2000

- To evaluate and adapt the detection method to other areas than telecommunications
  Example : robotics

- To prove Propositions 1 and 2 (*in progress*)

- To determine other types of properties to be checked

- To implement the detection method (*in progress*)