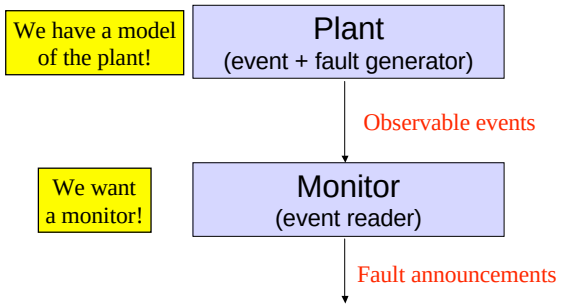
Fault Detection for Real-time Systems

Stavros Tripakis
VERIMAG

ARTIST2 Summer School, Sep 2005

1

Fault detection = Grey-box monitoring



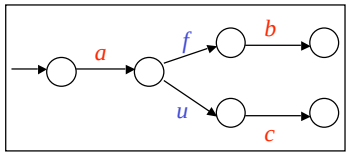
2

Requirements

- The monitor is deterministic.
- The monitor does not produce any false positives (announces a fault when no fault occurred).
- The monitor always announces a fault within a bounded delay after the fault occurred.
- Other sanity requirements (monitor is causal, does not change its mind, etc).

3

Example



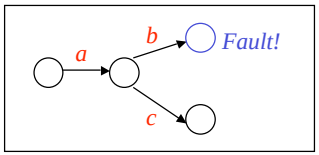
The plant model

Blue events are unobservable.

Red events are observable.

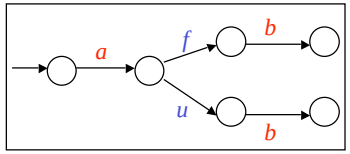
f is the fault event.

The monitor:



4

Not all faults are detectable!



We must FIRST check *detectability* …

and THEN synthesize the monitor.

5

Timed fault detection

The plant model is a timed automaton!

Plant
(event + fault generator)

Observable events + delays

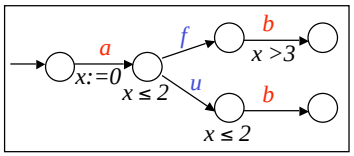
Monitor
(event reader)

The monitor can consult a clock!

Fault announcements

6

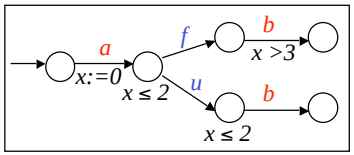
Example of timed fault detection



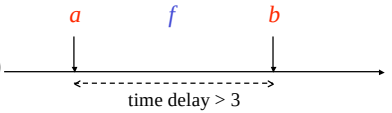
The plant model

7

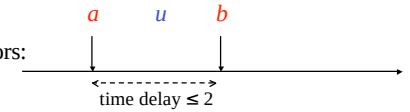
Example of timed fault detection



Faulty behaviors:
(case 1: *a* occurs)

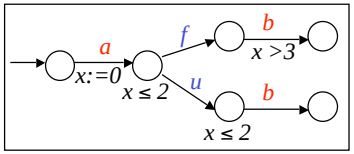


Correct behaviors:

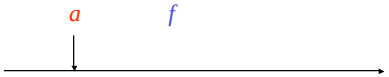


8

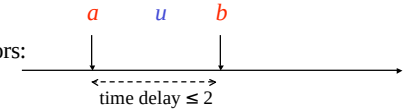
Example of timed fault detection



Faulty behaviors:
(case 2: no *a*)

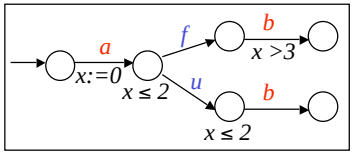


Correct behaviors:

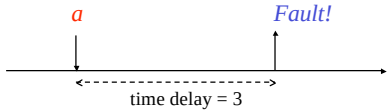


9

Example of timed fault detection



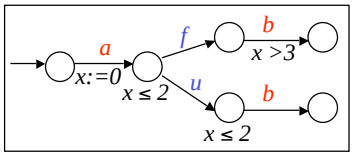
Faulty behaviors:



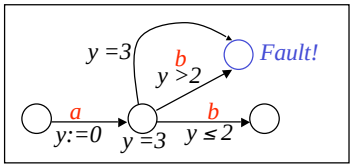
The monitor can report the fault within bounded delay: at most 3.

10

Example of timed fault detection



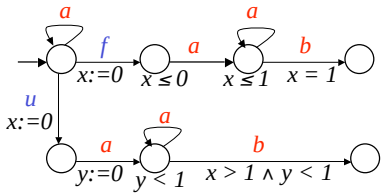
The monitor:



In this case, the monitor can be modeled as a timed automaton.
This is not always the case!

11

"Infinite-clock" monitor



- Faulty behaviors: there is some *a* exactly 1 time unit before the *b*.
- Correct behaviors: either no *b*, or no *a* exactly 1 time unit before the *b*.

12

Questions

- How to check detectability?
- How to synthesize a real-time monitor?
- When can the monitor be modeled as a timed automaton?

13

Formal definitions

- Timed behaviors over some alphabet Σ:

  $a\ 1.1\ u\ 0.4\ b\ 3\ f\ 2.2\ c$

- $\Sigma = \Sigma_o \cup \Sigma_u$

14

Formal definitions

- Observable behaviors:

$a\ 1.1\ u\ 0.4\ b\ 3\ f\ 2.2\ c$

Projection to observable events

$a\quad 1.5\quad b\quad 5.2\quad c$

15

Formal definitions

- Faulty behavior: contains a fault event.

$a\ 1.1\ u\ 0.4\ b\ 3\ f\ 2.2\ c$ faulty

$a\ 1.1\ u\ 0.4\ b\ 3\ u\ 2.2\ c$ non-faulty

16

Formal definitions

- T-faulty behavior (T is a delay):
  - faulty behavior,
  - at least T time elapses after the first occurrence of the fault.
- Examples:
  - 2-faulty (but not 3-faulty) behaviors:

$a\ 1.1\ u\ 0.4\ b\ 3\ f\ 2.5\ u$

$a\ 1.1\ u\ 0.4\ b\ 3\ f\ 1.9\ 0.1$

$a\ 1.1\ u\ 0.4\ b\ 3\ f\ 2.2\ u\ 0\ c\ 0.3\ u$

17

Real-time monitors

- A real-time monitor is a deterministic machine with an output:

$$Tr : S \times (\Sigma o \cup R_+)^* \rightarrow S$$

current state; observed event …; … or delay; next state

$$Out : S \rightarrow \{0,1\}$$

0: no fault, 1: fault

18

Real-time monitor requirements

- A real-time monitor defines a function:

$$M : (\Sigma o \cup R_+)^* \rightarrow \{0,1\}$$

$$M(\rho) = Out(Tr(s_0, \rho))$$

- The monitor is a *T-monitor* if, for every behavior ρ of A,

If ρ is not faulty, then M(Proj(ρ)) = 0

If ρ is T-faulty, then M(Proj(ρ)) = 1

19

Detectability questions

- Given T, does there exist a T-monitor?
- Is there a T-monitor for some T?
- Note: a T-monitor is also a (T+1)-monitor.

20

Necessary and sufficient condition for T-detectability

- There exists a T-monitor iff

$\nexists \rho, \rho' .$ ρ is T-faulty, ρ' is not faulty, and $Proj(\rho) = Proj(\rho')$

- or, equivalently

$\forall \rho, \rho' .$ if ρ is T-faulty and ρ' is not faulty, then $Proj(\rho) \neq Proj(\rho')$

21

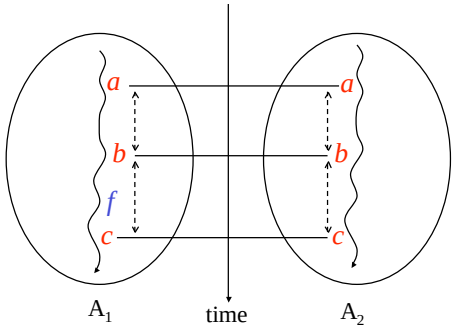
Checking detectability (for some T)

- Assumption: plant A is non-zeno.
- Make two copies of A, A1 and A2:
  - Copy/rename states, transitions, clocks, etc.
  - Copy/rename unobservable events.
  - Copy but do not rename observable events.
- Remove all faulty transitions from A2.
- Take the product B of A1 and A2: synchronize on common labels (i.e., observable events).
- Plant is detectable iff all faulty runs of B are zeno.

22

Checking detectability (for some T)

- Illustration:



23

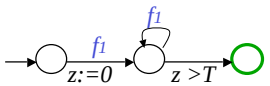
Checking detectability

- The proof is based on the following facts:
- Every run of B corresponds to a pair of runs ρ, ρ' of A which have the same projection.
- ρ' cannot be faulty.
- If a TA has a T-faulty run for all T, then it has a non-zeno faulty run.

24

Finding the minimum T for T-detectability

- Assumption: plant is detectable.
- Take the product of A_1, A_2 and the observer automaton below (where T is a parameter).



- Plant is T-detectable iff the final state of the observer cannot be reached.
- Perform a binary search on T : 0, 1, 2, 4, …etc. Complexity: log(T) reachability checks.

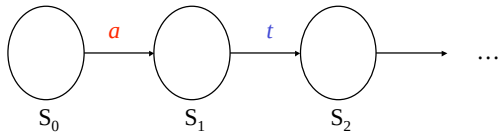
25

Questions

- How to check detectability? ✓
- How to synthesize a real-time monitor?
- When can the monitor be modeled as a timed automaton?

26

Synthesis of a real-time monitor: on-the-fly subset construction



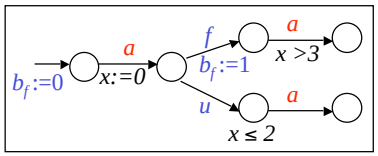
Monitor state = subset of timed automaton states = **all states plant can be in right now**

Monitor transition function = symbolic computation of successor states

Monitor output = announce fault if all plant states are **faulty**

27

Faulty states

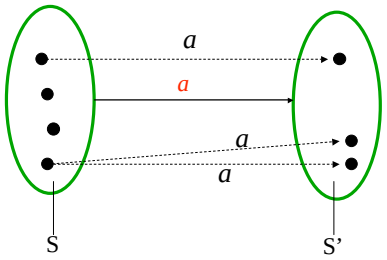


The plant model extended with boolean variable b_f

Faulty state = state where $b_f = 1$

28

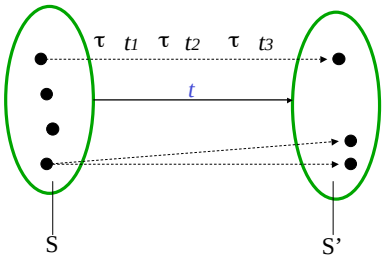
On-the-fly subset construction: event update



$S' = \{s' \mid \exists s \in S.\ s \xrightarrow{a} s'\}$

29

On-the-fly subset construction: delay update



$S' = \{s' \mid \exists s \in S.\ \exists \sigma \in UnobsSeq.\ s \rightarrow s' \wedge time(\sigma) = t\}$

Can be computed using an extra clock

30

Questions

- How to check detectability? ✓
- How to synthesize a real-time monitor? ✓
- When can the monitor be modeled as a timed automaton? ?

31

Going further

- Framework is more general:
  - Can detect state configurations
  - E.g. "state where j>2 and (b=true or p=nil)".
  - "Exists" vs. "For all" acceptance.
- Black-box monitoring:
  - Properties expressed in real-time logic (e.g. MITL)
  - Translate formula into timed automaton
  - Monitor timed automaton (OTF subset construction)
- Black-box testing (c.f. previous talks)
- Implementability:
  - Real-time monitors are "ideal": analog-clock.
  - Can use digital-clocks [SPIN'04,FORMATS'05].

32

Reading

- [Tripakis, FTRTFT'02]:
  - Original paper, on-the-fly subset construction technique
- [D'Souza-Madhusudan STACS'02, Bouyer et al CAV'03, Bouyer-Chevalier, FOSSACS'05]:
  - Resource-bounded synthesis (region-graph based)
- [Krichen-Tripakis, SPIN'04, FORMATS'04]:
  - Black-box monitoring and testing
  - Resource-bounded synthesis (zone based)
- [Altisen-Tripakis, FORMATS'05]:
  - Implementability with digital-clock and other models

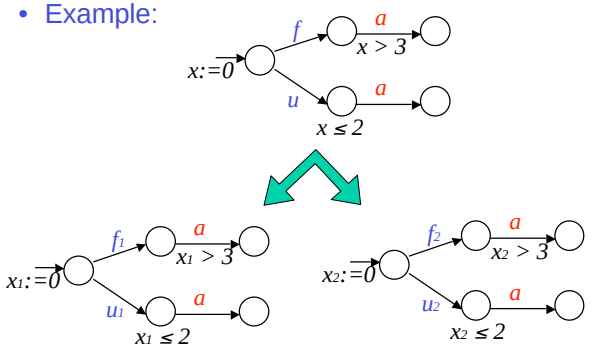
33

Appendix

34

Checking detectability

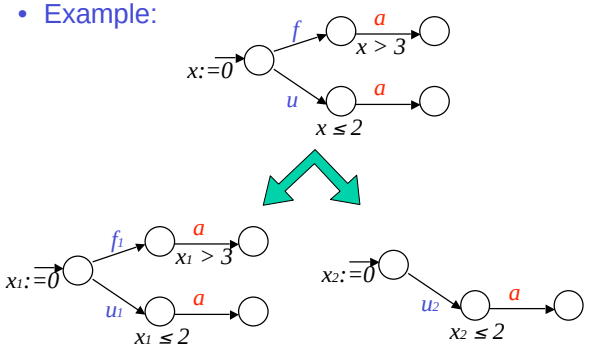
- Example:



35

Checking detectability

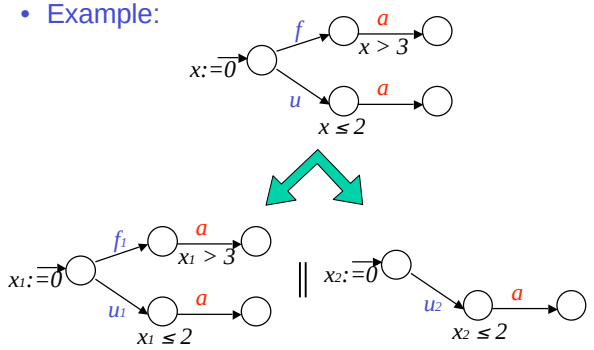
- Example:



36

Checking detectability

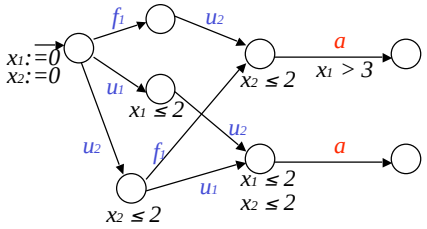
- Example:



37

Checking detectability

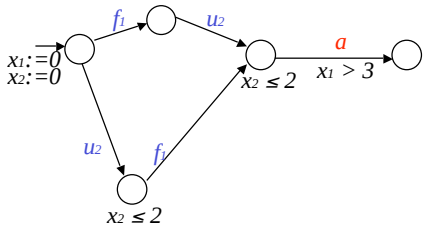
- Example:



38

Checking detectability

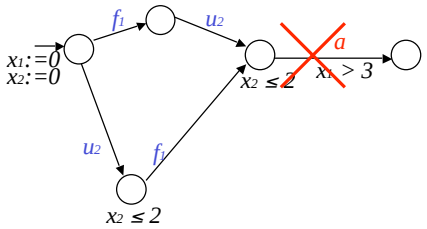
- Example:



39

Checking detectability

- Example:



40