Diagnosis (07)
Introduction to Discrete Event Systems

Alban Grastien
alban.grastien@rsise.anu.edu.au
1. Modelling the System by Finite State Machines
2. Other Formalisms
3. Diagnosis
1 Modelling the System by Finite State Machines

2 Other Formalisms

3 Diagnosis
A Finite State Machine is an oriented graph with a (set of) initial state(s).

Formally $\langle Q, E, T, I \rangle$:
- $Q$ is the set of nodes
- $E$ is the set of transition labels
- $T : Q \times E \times Q$ is the set of transitions
- $I \subseteq Q$ is the set of initial nodes (often, $|I| = 1$)

Equivalence: FSM = automaton
FSM to Model a System

A system can be modelled by an FSM → discrete event system.

- a node of the FSM represents a state of the system
- a transition between two nodes represents the evolution of the state of the system
- the label of a transition represents the event(s) that modified the state of the system (or that is/are consequence(s) or the modification of the state)
- the initial states represents the possible state at the beginning of the diagnosis

Dynamics

- Time Driven Systems
- Event Driven Systems
Observations

- Partial observation of the state ([Largouët & Cordier, DX 2001])
- Generally, observation of the transitions
  - Viewer [Lamperti & Zanella, 2003]: $T \rightarrow O \cup \{NonObs\}$
  - Generally, simplified: $O \subseteq E$
Faulty Behaviours

What we want to detect

- Is the current state faulty $s \in F \subseteq Q$?
- Did the faulty event $f \in F \subseteq E$ occur?
- Was the faulty transition $t \in F \subseteq T$ triggered?
- Lamperti & Zanella’s ruler
- Did the faulty behaviour represented by the specified automaton $A$ occur?
1. Modelling the System by Finite State Machines
2. Other Formalisms
3. Diagnosis
Definition

\[ \langle V, E, R, I \rangle \]

- \( V \) is a set of Boolean variables and \( I \) an assignment of the variables,
- \( E \) is a set of events, and
- \( R \) is a set of rules (precondition + effects).

A state is an assignment \( S : V \rightarrow \{0, 1\} \).

A STRIPS-like representation can be easily translated into an automaton.

Existing algorithm do not take benefit from such a representation.
Petri Nets

**Definition**

“Petri Nets: Properties, Analysis and Applications” [Tadao Murata, 89]

\[ PN = (P, T, F, W, M_0) : \]

- \( P \) is a set of places,
- \( T \) is a set of transition so that \( P \cap T = \emptyset \),
- \( F \subseteq (P \times T) \cup (T \times P) \) is the set of arcs,
- \( W : F \times \mathbb{N}^+ \) is a weighting function, and
- \( M_0 : P \rightarrow \mathbb{N} \) is the initial marking.

A state is a marking \( M : P \rightarrow \mathbb{N} \).
Petri Nets – Example
Advantages – Drawbacks

- Compact: the size of an equivalent automaton has an exponential number of states
- Same expressiveness as automata
- Very efficient to model flow of resources
- Dedicated algorithms (unfolding)

- However, it is often required to use methods equivalent to automata

- R. Boël, A. Benveniste
Languages

- Given an alphabet $\Sigma$, a language is a set of words: $L \subseteq \Sigma^*$. 
- A word $s \in L$ represents a possible evolution of the system.
- The language is prefix-closed.
- A language is more expressive than an automaton.

... but a language is actually generally represented by an automaton.
**Timed Automata [Alur, 1992]**

- A set of **clocks** is associated with the system.
- A state of the system is modelled by a state of the automaton + an assignment in $\mathbb{R}^+$ of all the clocks.
- Transitions and states are **guarded** by conditions on the clocks.
- Clocks can be reset on transitions.
- A (non empty) amount of time slip by between two transitions.
Manipulation Timed Automata

- Basically identical to classical automata (but more complex)
- Notion of clock regions
- Difference Bound Matrices
Automata with Parameters

- Similar to timed automata
- A set of variables is associated with the system.
- A state of the system is modelled by a state of the automaton + an assignment of all the variables.
- Transitions (not states) are guarded by conditions on the variables.
- The value of the variables can be modified by transitions.

Manipulating these Automata

- Identical to classical automata (this is only a compact representation).
1. Modelling the System by Finite State Machines

2. Other Formalisms

3. Diagnosis
The model is an automaton.
The transitions are labeled by a single event.
Some events are observable: $O \subseteq E$; the number of unobserved transitions triggered is not known.
Some events are faulty: $F = F_1 \cup \cdots \cup F_f \subseteq E$.
The observations are received in the order they are emitted.
Diagnosis

- Given the model
- Given a flow of observations
- What possible fault modes did occur?
## Sampath Diagnoser

**Sampath et al. 1996**

- **Off-line compilation of the model**
- **A state of the Sampath diagnoser is a set of pairs** \( \langle s, fm \rangle \)**
  - where:
    - \( s \) is a state of the system
    - \( fm \subseteq \{F_1, \ldots, F_f\} \) is a fault mode
  - **The semantics of**
    - \( \{\langle s_1, fm_1 \rangle, \langle s_2, fm_2 \rangle, \langle s_3, fm_3 \rangle, \langle s_4, fm_4 \rangle, \langle s_5, fm_5 \rangle\} \) **is that**
      - the state after the last observation is \( s_1 \) and the set of faults that occurred is \( fm_1 \), or
      - the state after the last observation is \( s_2 \) and the set of faults that occurred is \( fm_2 \), or
      - etc.
Using a Diagnoser

Construction of the Diagnoser

- The initial state of the Sampath diagnoser is $\{\langle s_0, \emptyset \rangle\}$
- For each state $s = \{\langle s_1, fm_1 \rangle, \ldots, \langle s_k, fm_k \rangle\}$
  - For each observable event $o$
    - Add a transition between $s$ and $s'$ labeled by $o$ where $s'$ contains the set of pairs $\langle s'_j, fm'_j \rangle$ so that
    - there exists a path $p$ label with unobservable events from a state $s_i$ to a state $s''$
    - $fm'_j = fm_i \oplus p$ (the fault mode is the previous fault mode added with the faults in the path),
    - there exists a transition from $s''$ to $s'_j$ labeled by $o$

Using the Diagnoser

Given the sequence of observation, simply follow the state in the diagnoser.
Example
## Discussion

### Advantages

- **Fast:** the complexity of the diagnosis task is linear in the number of observations and does not depend on the size of the system.

### Drawbacks

- The worst case size of the diagnoser is $2^{|Q| \times 2^f}$: for realistic real-world systems, this method cannot be applied.
- The observation must be totally ordered, or the size of the diagnoser is even worst.

### Improvements

- Specialised diagnosers (Y. Pencolé et al.)
- BDD (A. Schumann et al.)