Failure Diagnosis of Discrete Event Systems: A Temporal Logic Approach

Shengbing Jiang

Electrical & Controls Integration Lab
General Motors R&D

Outline

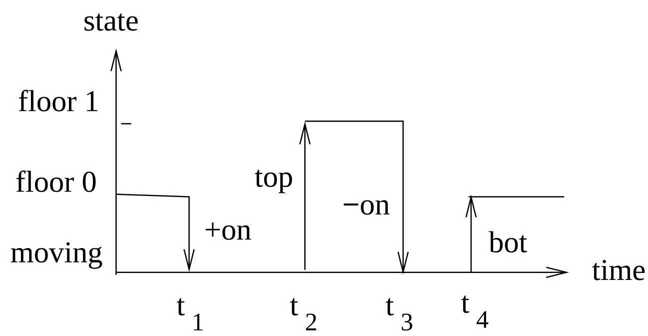
- Introduction
- Notion of Diagnosability in Temporal Logic Setting
- Algorithm for Diagnosis
- Example
- Conclusion

DES: Introduction

- Discrete states: driven by randomly occurring events
- Events: discrete qualitative changes
  - arrival of part in a manufacturing system
  - loss of message packet in a communication network
  - termination of a program in an operating system
  - execution of operation in database system
  - arrival of sensor packet in embedded control system
- Examples of discrete event systems:
  - computer and communication networks
  - robotics and manufacturing systems
  - computer programs
  - automated traffic systems

DES: Example

- States change in response to randomly occurring events
  $\Rightarrow$ piecewise-constant state trajectory



typical state−trajectory of an elevator

- State trajectory a sequence of triples:
  $(x_0, \sigma_1, t_1)(x_1, \sigma_2, t_2) \ldots$

- Untimed trajectory (for untimed specification):
  $(x_0, \sigma_1)(x_1, \sigma_1) \ldots$

- Under determinism this is equivalent to:
  x_0 and $\sigma_1 \sigma_2 \ldots$

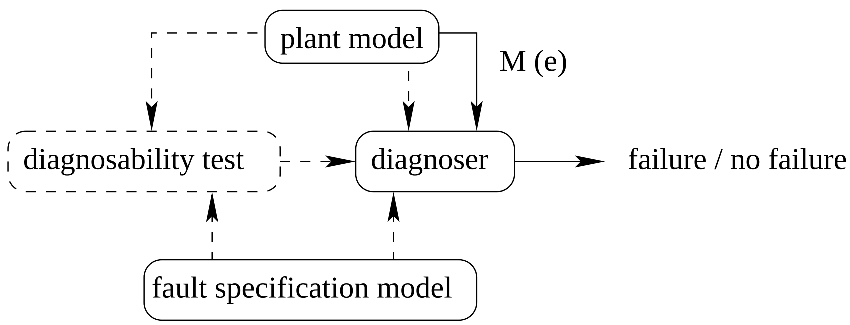
- Collection of all event traces, *language*, $L = pr(L) \neq \emptyset$
  Collection of "final" traces, *marked language*, $L_m \subseteq L$

- *Language model*, (L, L_m), also modeled as automaton:
  $G := (X, \Sigma, \alpha, x_0, X_m)$; $\quad (L(G), L_m(G))$ language model

Failure Diagnosis of DESs

- Failure: deviation from normal or required behavior
  - occurrence of a failure event
  - visiting a failed state
  - reaching a deadlock or livelock

- Failure Diagnosis: detecting and identifying failures



e: event generated; $M(e)$: event observed/sensored
$M(e) = \epsilon$: no sensor for the event e
$M(e_1) = M(e_2) \neq \epsilon$: e.g. motion sensor, detect the movement of a part, not the moving direction and the part type

- Diagnoser: observes the sequence of generated events, and determines (possibly with a delay that is bounded) whether or not a failure occurred

Prior Results:

- Formal language / automaton fault specification

- Fault spec.: only “safety properties”

Our Contribution:

- A temporal logic approach for failure diagnosis of DESs

  Temporal logic has a syntax similar to natural language and has a formal semantics

- Fault spec.: both “safety” and “liveness” properties

Linear-time Temporal Logic (LTL): Introduction

Describing properties of sequence of proposition set traces

Notations

- AP: set of atomic propositions
- $\Sigma_{AP} = 2^{AP}$: power set of AP
- Σ_{AP}^{*}: set of all finite proposition set traces
- Σ_{AP}^{ω}: set of all infinite proposition set traces

Temporal operators & their interpretations

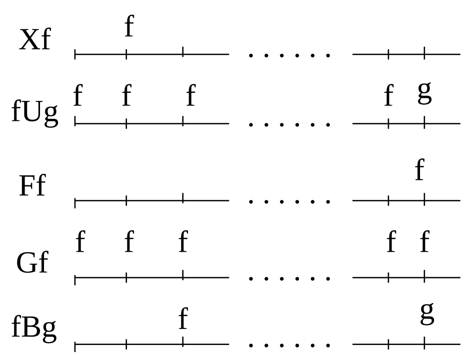
X : next,

U : until,

F : future or eventually, Ff=TrueUf

G : globally or always, Gf=~F~f

B : before, fBg=~(~fUg)



Syntax of LTL formulae

P1 $p \in AP \Rightarrow p$ is a LTL formula.

P2 f_1 and f_2 are LTL formulae $\Rightarrow$ so are $\neg f_1$, $f_1 \vee f_2$, and $f_1 \wedge f_2$.

P3 f_1 and f_2 are LTL formulae $\Rightarrow$ so are Xf_1, $f_1 U f_2$, Ff_1, Gf_1, and $f_1 B f_2$.

Semantics: proposition-trace $\pi = (L_0, L_1, \cdots) \in \Sigma_{AP}^{\omega}$
$\pi^i = (L_i, \cdots)$, $\forall i \geq 0$

1. $\forall p \in AP$, $\pi \models p \iff p \in L_0$.

2. $\pi \models \neg f_1 \iff \pi \not\models f_1$.

3. $\pi \models f_1 \vee f_2 \iff \pi \models f_1$ or $\pi \models f_2$.

4. $\pi \models f_1 \wedge f_2 \iff \pi \models f_1$ and $\pi \models f_2$.

5. $\pi \models Xf_1 \iff \pi^1 \models f_1$.

6. $\pi \models f_1 U f_2 \iff \exists k \geq 0$, $\pi^k \models f_2$
and $\forall j \in \{0, 1, \cdots, k-1\}$, $\pi^j \models f_1$.

7. $\pi \models Ff_1 \iff \exists k \geq 0$, $\pi^k \models f_1$.

8. $\pi \models Gf_1 \iff \forall k \geq 0$, $\pi^k \models f_1$.

9. $\pi \models f_1 B f_2 \iff \forall k \geq 0$ with $\pi^k \models f_2$,
$\exists j \in \{0, 1, \cdots, k-1\}$, $\pi^j \models f_1$.

Examples of LTL formulae

$G\neg R_1$: an invariance (a type of safety) property.

$G($**(message sent)** $\Rightarrow F$**(message received)**$)$:
a recurrence (a type of liveness) property.

FGp : a stability (a type of liveness) property.

Notion of Diagnosability in Temporal Logic Setting

System Model: $P = (X, \Sigma, R, X_0, AP, L)$

- X, a finite set of states;
- Σ, a finite set of event labels;
- $R : X \times \Sigma \cup \{\epsilon\} \times X$, a transition relation, $\forall x \in X$, $\exists \sigma \in \Sigma \cup \{\epsilon\}$, $\exists x' \in X$, $(x, \sigma, x') \in R$ (P is nondeterministic & nonterminating);
- $X_0 \subseteq X$, a set of initial states;
- AP, a finite set of atomic proposition symbols;
- $L : X \to 2^{AP}$, a labelling function.

$M : \Sigma \cup \{\epsilon\} \to \Delta \cup \{\epsilon\}$, an observation mask.

Fault specification: LTL formula f.

$\pi = (x_0, x_1, \cdots)$, $\pi_{AP} = (L(x_0), L(x_1), \cdots)$:
$\pi \models f$ if $\pi_{AP} \models f$

Faulty state-trace

An infinite state-trace π is *faulty* if $\pi \not\models f$.

Remark captures both safety and liveness failures

Indicator

A finite state-trace π is an *indicator* if all its infinite extensions in P are faulty.

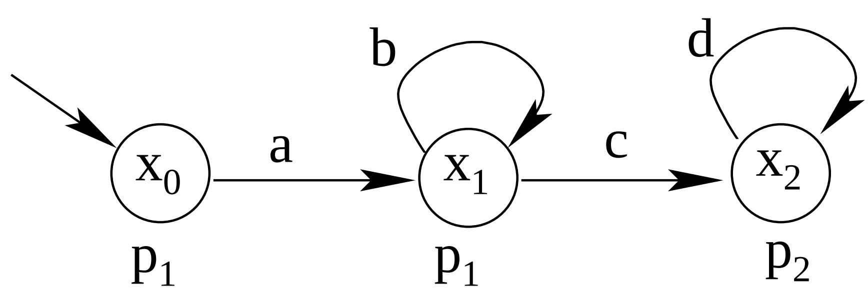
Remark: can only detect indicators through observation of finite length event-traces

Pre-diagnosability

P is *pre-diagnosable* w.r.t. f if every faulty state-trace in P possesses an indicator as its prefix.

Remark Needed for detecting all failures through observation of finite length event-traces

Example



$f = GFp_2$: not pre-diagnosable; no indicator for $x_0 x_1^\omega$

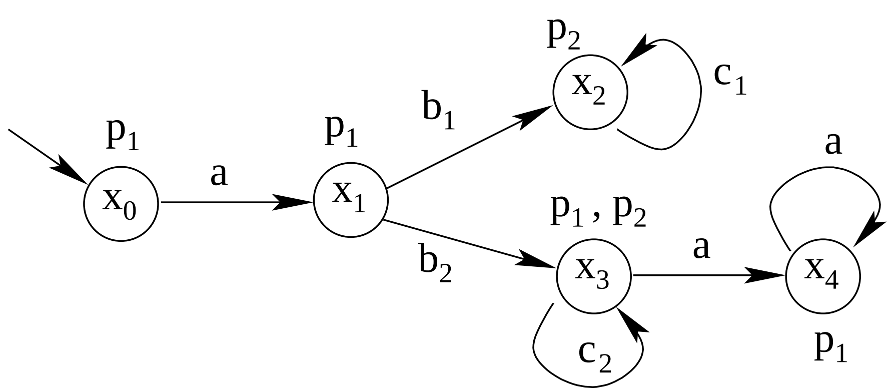
$f = GFp_1$: pre-diagnosable

Diagnosability: single specification

P is *diagnosable* w.r.t. M and f
if P is pre-diagnosable and

Exists a detection delay bound n such that
For all indicator trace π_0
For all extension-suffix π_1 of π_0, $|\pi_1| \geq n$
For all π' indistinguishable from $\pi_0\pi_1$
It holds that π' is an indicator trace

Example



$M(b_1) = M(b_2) = b$; $M(c_1) = M(c_2) = c$
$f = GFp_2$: diagnosable (no faulty trace looks like a non-faulty trace)
$f = Gp_1$: pre-diagnosable but not diagnosable

Diagnosability: multiple specifications

P is *diagnosable* w.r.t. M and $\{f_i, i = 1, 2, \cdots, m\}$
if P is diagnosable w.r.t. M and each f_i, $i = 1, 2, \cdots, m$.

Remark: Suffices to study the case of only one fault specification

Algorithm for Diagnosis with LTL Specifications

Problem of failure Diagnosis

Given P, M, f:

- Test the diagnosability of P w.r.t. M and f;
- If P is diagnosable, then construct a diagnoser for P.

Algorithm 1: diagnosis for single fault specification

1. Construct a tableau T_f for f that contains all infinite proposition-traces satisfying f, and let $\{F_i, 1 \le i \le r\}$ denote the generalized Büchi acceptance condition of f.

2. Test the pre-diagnosability of P.

   - Construct $T_1 = T_f||_{AP}P$ that generates traces that are accepted by P and are limits of traces satisfying f.

   - Check whether every infinite state-trace generated by T_1 satisfies f, i.e., whether every infinite state-trace generated by T_1 visits each F_i infinitely often: $T_1 \models \wedge_{i=1}^{r} GFF_i$, a **LTL model checking** problem.

     NO $\iff$ P is not pre-diagnosable.

3. Test the diagnosability of P.

   - Construct $T_2 = M^{-1}M(T_1)||_{\Sigma}P$ that accepts finite traces of P indistinguishable from finite traces of T_1, i.e., prefixes of non-faulty traces.

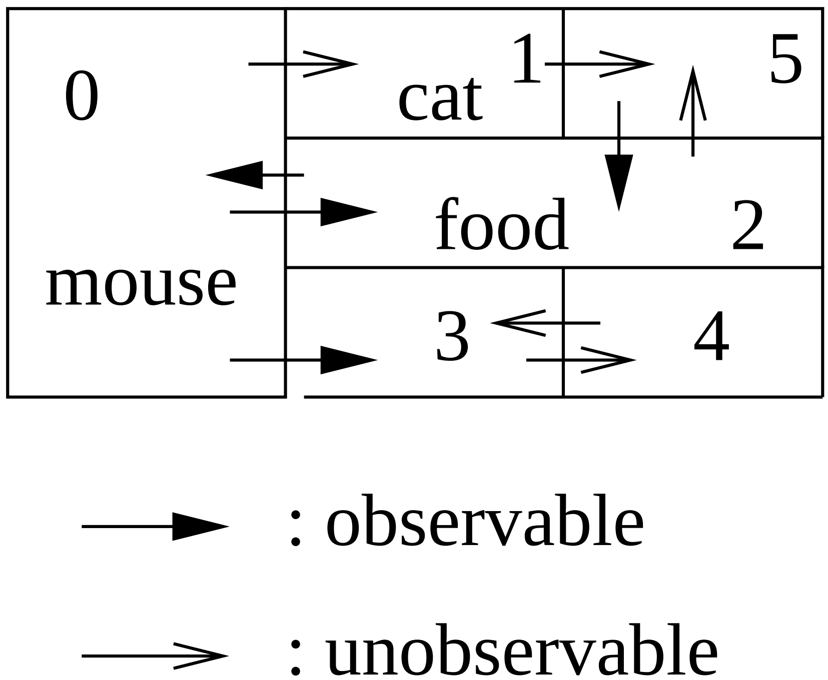
   - Check whether every infinite trace in T_2 satisfies f: $T_2 \models f$, a **LTL model checking** problem.

     NO $\iff$ P is not diagnosable.

4. Output $M(T_1)$ as the diagnoser D.

Complexity: $O(2^{|f|}|X|^4)$; size of D: $O(2^{|f|}|X|)$.

Illustrative Example: Mouse in a Maze



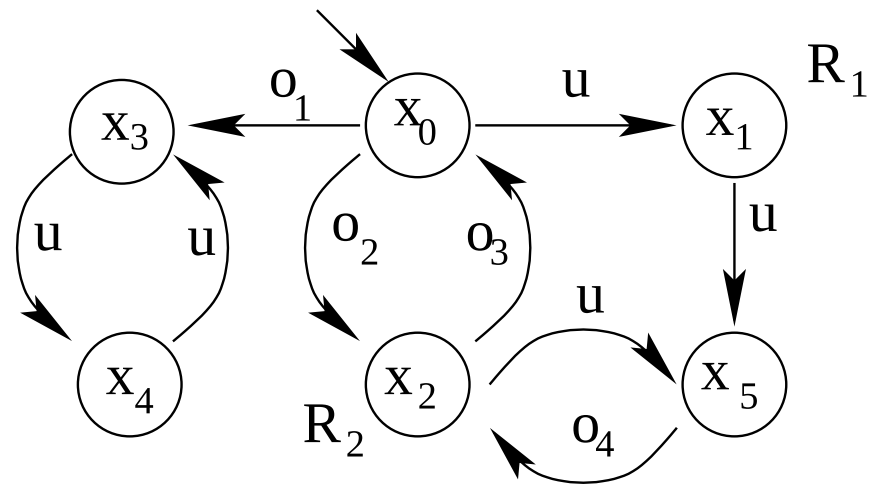
Spec 1 Never visit room 1 (an invariance, a type of safety, property).

$G\neg R_1$: Globally (always) not in Room 1

Spec 2 Visit room 2 for food infinitely often (a recurrence, a type of liveness, property).

GFR_2: Globally (always) in future in Room 2

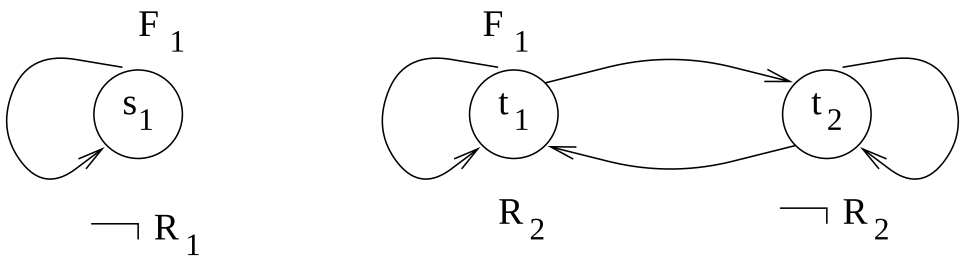
System model



$M(u) = \epsilon$, $M(o_i) = o_i$ for $0 \leq i \leq 4$; $AP = \{R_1, R_2\}$;
$L(x_i) = \emptyset$ for $i \notin \{1, 2\}$, $L(x_1) = \{R_1\}$, $L(x_2) = \{R_2\}$.

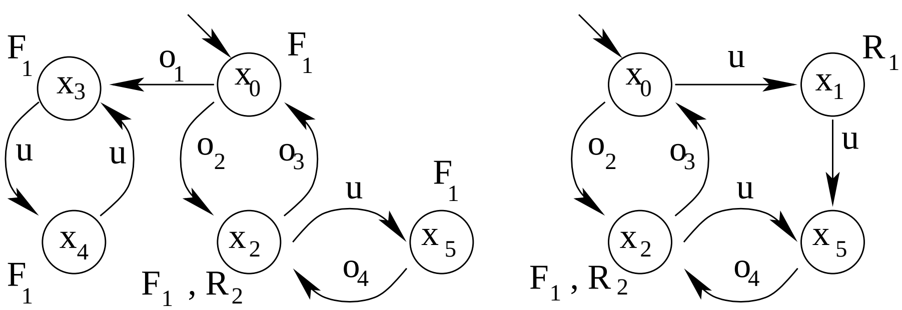
Specifications: $f_1 = G\neg R_1$, $f_2 = GFR_2$.

Tableau T_{f_i}, $i = 1, 2$.



(a) (b)

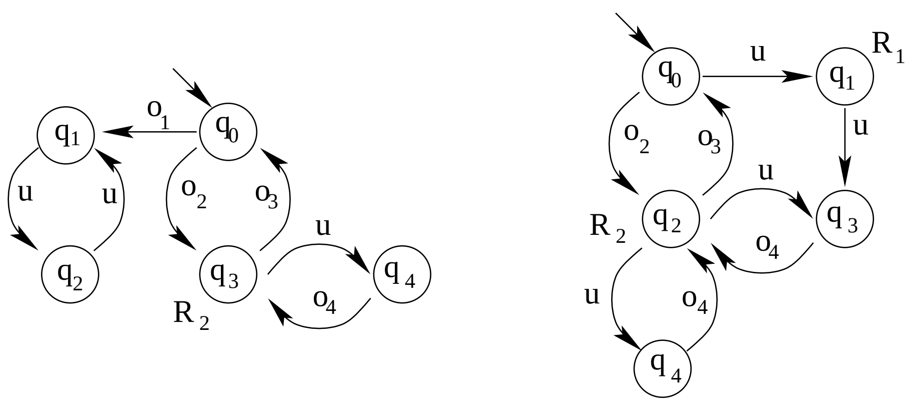
Checking pre-diagnosability: $T_1^{f_i} = T_{f_i} ||_{AP} P$, $i = 1, 2$; $T_1^{f_i} \models GFF_1$? $i = 1, 2$.



(a) (b)

P is pre-diagnosable w.r.t. f_1 and f_2 respectively.

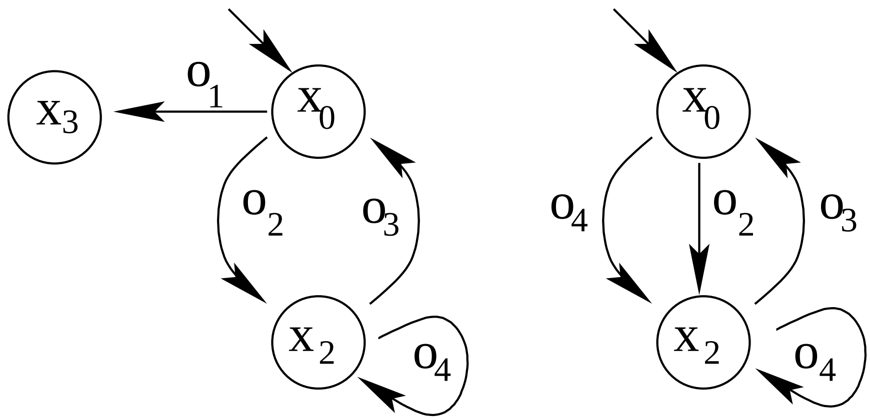
Checking diagnosability: $T_2^{f_i} = M^{-1}M(T_1^{f_i})||_\Sigma P$, $i = 1, 2$; $T_2^{f_1} \models G\neg R_1$? $T_2^{f_2} \models GFR_2$?



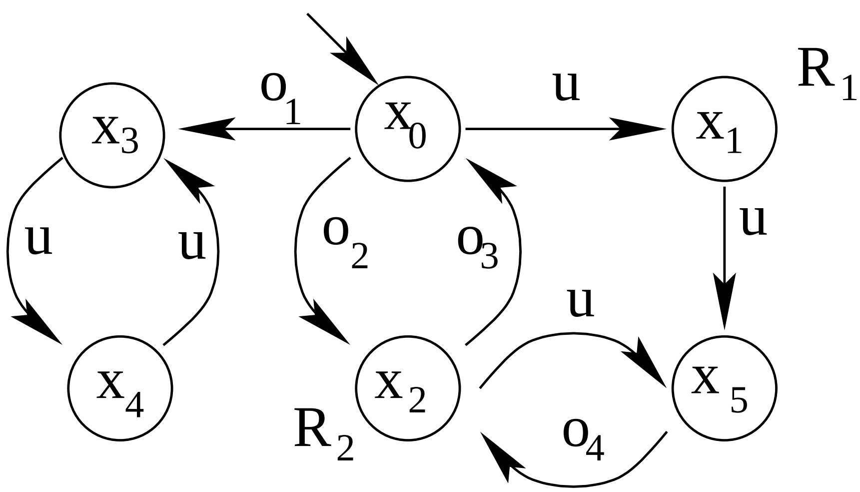
(a) (b)

P is diagnosable w.r.t. f_1 and f_2 respectively.

Diagnoser $D = \{M(T_1^{f_1}), M(T_1^{f_2})\}$



(a) (b)



Specifications: $f_1 = G\neg R_1$, $f_2 = GFR_2$.

Conclusion

- A framework for failure diagnosis in LTL setting
- Notions of indicator, pre-diagnosability, and diagnosability
- Algorithms for checking pre-diagnosability & diagnosability in proposed framework
- Construction of diagnoser for on-line diagnosis in proposed framework
- Complexity analysis (polynomial in the plant size, exponential in the formula length)