Random generator of $k$-diagnosable discrete event systems

Yannick Pencolé

DISCO Team (Diagnosis and Supervisory COntrol)
CNRS-LAAS, Université de Toulouse, FRANCE

August 31, 2015
Introduction

- Classical problem of fault diagnosis of discrete event systems
- Need of independent discrete event system benchmarks
  - For testing diagnosis algorithms
  - For testing diagnosability algorithms
- Two types of benchmarks (see SAT community)
  - Real-world benchmarks
    - Pros: validate an algorithm on a real case
    - Cons: are they really real? hard to get (industry)? hard to show (confidentiality)?
  - Random benchmarks
    - Pros: as many as we like, parametrized
    - Cons: how far it is from real case? any uncontrolled bias? does not valid any real-world solution
- Question: is it better to have an algorithm that solves one real case (and only one) or to have an algorithm that solves many random non-real problems?
Motivation for this work

- Creation of random DES benchmarks with the help of a specific software
- The generated DES are \textit{diagnosable}.
- Why diagnosability:
  - Worst-case of a diagnosability checker is usually when a system is diagnosable
  - Some diagnosis algorithms require that the system is diagnosable
- Identification of parameters of systems that have influence on the complexity of the algorithms to validate
- Looking at diagnosability from a different viewpoint:
  - usually, a checker looks for the reason why a system is not diagnosable
  - here, we look at the reasons why a system is diagnosable.
DES Model: finite state automaton

SD = (Q, Σ, T, q₀) where:
- Q is a finite set of states;
- Σ is a finite set of events;
- T ⊆ Q × T × Q is a finite set of transitions;
- q₀ is the initial state of the system.
Diagnosis problem and solution

- Observable events $\Sigma_o$, Unobservable events $\Sigma_{uo}$, $\Sigma = \Sigma_o \oplus \Sigma_{uo}$
- Fault: a non-observable event $f \in \Sigma$
- Observations OBS: a sequence $\sigma$ of observable events
- Diagnosis problem: $(SD, OBS, FAULTS)$
  Find the set of active faults $F \subseteq FAULTS$ that could have occurred in the system based on the model SD and the observations OBS.

\[
\begin{align*}
\Delta(eee) &= \{ \emptyset \} \\
\Delta(ec) &= \{ \emptyset \} \\
\Delta(cdc) &= \{ \{ f \}, \emptyset \}
\end{align*}
\]
Diagnosability

- $f$ is diagnosable in a system $S$ if :

\[ \exists n \in \mathbb{N}^+, \text{Diagnosable}(n) \]

where \text{Diagnosable}(n) stands for :

\[ \forall \tau_1.f \in \mathcal{L}(S), \forall \tau_2 : \tau_1.f.\tau_2 \in \mathcal{L}(S) \]
\[ |P_{\Sigma_o}(\tau_2)| \geq n \Rightarrow \]
\[ (\forall \tau \in \mathcal{L}(S), (P_{\Sigma_o}(\tau) = P_{\Sigma_o}(\tau_1.f.\tau_2) \Rightarrow f \in \tau)). \]

Intuition: once $f$ has occurred, the next $n$ observable events are always sufficient to diagnose $f$ with certainty:

\[ \Delta(P_{\Sigma_o}(\tau_1.f)P_{\Sigma_o}(\tau_2)) = \{F_1, F_2, \ldots, F_n\}, f \in F_i \]
A fault $f$ is $k$-diagnosable if:

$$\text{Diagnosable}(k) \land \neg \text{Diagnosable}(k - 1)$$

$k$ is the minimal number of observable events after the occurrence of a fault that are always sufficient to diagnose $f$ with certainty.

- $k$-diagnosable $\Rightarrow$ diagnosable
- diagnosable $\Rightarrow \exists k$, $k$-diagnosable

This $k$ is a property of the system.
Objectives of the proposed generator

- Random generation of $k$-diagnosable models
  - for a given $k$.
  - for a given number of states $n$.
  - for a given maximal state output degree $deg$
  - deterministic models

Generation based on the notion of fault signature.
Signatures

- The signature of an event \( f \) into a system \( S \) is the language \( \text{Sig}(f) \subseteq \Sigma^*_o \) such that

\[
\text{Sig}(f) = \{ \sigma \tau | \tau = \tau_1.o.\tau_2 \in L(S), f \in \tau_1, o \in \Sigma_o, \tau_2 \in \Sigma^*, \sigma = P_{\Sigma_o}(\tau) \}
\]

- The signature of the absence of \( f \) into a system \( S \) is the language \( \text{Sig}(\neg f) \subseteq \Sigma^*_o \) such that

\[
\text{Sig}(\neg f) = \{ \sigma \tau | \tau = \tau_1.o \in L(S), f \not\in \tau_1, o \in \Sigma_o, \sigma = P_{\Sigma_o}(\tau) \}
\]

\[
\text{Sig}(f) = \{ c, cde^*, cde^*c, cd(e^*cd)^+, cd(e^*cd)^+c \}
\]

\[
\text{Sig}(\neg f) = \{ e, e^*c, (e^*cd)^+, (e^*cd)^+e^*c \}
\]
Ambiguous signatures

- Any observable trace $\sigma$ that belongs to $\text{Sig}(f)$ AND $\text{Sig}(-f)$ is ambiguous

$$\forall \sigma \in \text{Sig}(f) \cap \text{Sig}(-f), \Delta(\sigma) \text{ is ambiguous.}$$

- To be diagnosable, the number of observable continuations of an observable ambiguous trace must be finite

- Twin-plant method: checking whether there exists in $\text{Sig}(f) \cap \text{Sig}(-f)$ an unbounded continuation of an observable trace $\sigma \in \text{Sig}(f) \cap \text{Sig}(-f)$.

- To be $k$-diagnosable, for any $\sigma \in \text{Sig}(f) \cap \text{Sig}(-f)$:
  - for any $\sigma \in \text{Sig}(f) \cap \text{Sig}(-f)$, any continuation $\sigma'$ of $\sigma$ such that $\sigma\sigma' \in \text{Sig}(f) \cap \text{Sig}(-f)$ must be such that:
    $$|\sigma\sigma'| - |\sigma| \leq k - 1$$
  - there exists at least a couple $\sigma, \sigma'$ such that
    $$|\sigma\sigma'| - |\sigma| = k - 1$$
Given $n$ a number of states, $k$ parameter, $deg$ allowed number of output transitions

1. Computation of a random ambiguous signature $AmbSig$
2. Parsing of the ambiguous signature and random generation of the system.
Example : Ambiguous signature

Very simple example :

\[ \text{Sig}(f) \cap \text{Sig}(-f) = \{ o_1o_2, \ o_1o_2o_2, \ o_1o_1o_1o_2, \ o_1o_1o_1o_2o_2, \ldots \} \]
Example: Generated system
Example: Generated system
Example: Generated system
Example: Generated system
Example: Generated system
Example : Generated system
Example: Generated system
Analysis of the generated system

- By construction, the signature of $f$ is:
  - $(o_1 o_1)^* o_3^+$
  - $o_1 (o_1 o_1)^* o_2$
  - $o_1 (o_1 o_1)^* o_2 o_2$
  - $o_1 (o_1 o_1)^* o_2 o_2 o_2^+$

- By construction, the signature of $\neg f$ is:
  - $o_1^+$
  - $o_1 (o_1 o_1)^* o_4^+$
  - $o_1 (o_1 o_1)^* o_2$
  - $o_1 (o_1 o_1)^* o_2 o_2$
  - $o_1 (o_1 o_1)^* o_2 o_2 (o_5 o_4^* + o_4^+)$
Analysis of the generated system

- By construction, the signature of \( f \) is:
  \[
  \begin{align*}
  &\cdot (o_1 o_1) o_3^+ \\
  &\cdot o_1 (o_1 o_1) o_2 \\
  &\cdot o_1 (o_1 o_1) o_2 o_2 \\
  &\cdot o_1 (o_1 o_1) o_2 o_2 o_2^+ \\
  \end{align*}
  \]

- By construction, the signature of \( \neg f \) is:
  \[
  \begin{align*}
  &\cdot o_1^+ \\
  &\cdot o_1 (o_1 o_1) o_4^+ \\
  &\cdot o_1 (o_1 o_1) o_2 \\
  &\cdot o_1 (o_1 o_1) o_2 o_2 \\
  &\cdot o_1 (o_1 o_1) o_2 o_2 (o_5 o_4^+ + o_4^+) \\
  \end{align*}
  \]

Ambiguous signature:

\[
Sig(f) \cap Sig(\neg f) = o_1 (o_1 o_1) o_2 + o_1 (o_1 o_1) o_2 o_2
\]
Ambiguous signature:

\[ \text{Sig}(f) \cap \text{Sig}(\neg f) = o_1(o_1o_1)^*o_2 + o_1(o_1o_1)^*o_2o_2 \]

- \( \neg f \)-certainty as long as we see \( o_1(o_1o_1)^* \).
- \( f \)-certainty as soon as \( o_3 \) occurs after \( o_1(o_1o_1)^* \).
- \( f \)-ambiguity as soon as \( o_2 \) occurs after \( o_1(o_1o_1)^* \).
- Still \( f \)-ambiguity if another \( o_2 \) occurs after \( o_1(o_1o_1)^*o_2 \).
- \( f \)-certainty as soon as a 3rd \( o_2 \) occurs after \( o_1(o_1o_1)^*o_2o_2 \).
- \( \neg f \)-certainty is any other case.

So \( f \) is 3-diagnosable.
• Generator, part of the toolset:
  DIADES: diagnosis of discrete event systems
• UNIX command line program: dd-diagnosable-system-generator
• Pure C++ implementation (C++11)
Command line

./dd-new-diagnosable-des-generate --help
Diades: generator of diagnosable discrete event systems

Allowed options:
--help
--states arg (still experimental)
--observables arg
--unobservables arg
--output_degree arg
--k arg
--min_observable_ambiguity arg
--ambiguity_ratio arg
--seed arg
Some available benchmarks

A set of benchmarks already generated:
http://homepages.laas.fr/ypencole/benchmarks

- From 100-states systems to 200 000-states systems
- From $k = 1$ to $k = 5000$
- From $deg = 3$ to $deg = 100$
- At present, only des_comp format
- More benchmarks will be generated
- Any idea of other parameters to control?
Perspectives

- Better control of the number of states
- Performance improvements
- Generation of component-based diagnosable system
- Automatic generation of diagnosis scenarios