NICTA

Conflict-Based Diagnosis of Discrete-Event Systems

Alban Grastien — Patrik Haslum — Sylvie Thiébaux

Australian Government
Department of Broadband, Communications and the Digital Economy
Australian Research Council

NICTA Funding and Supporting Members and Partners

Contribution

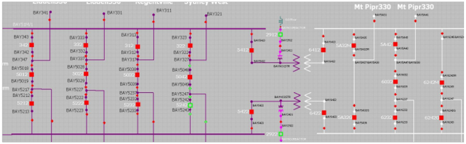
We define a conflict-based diagnosis theory for discrete event systems

- Compatible with the existing conflict-based diagnosis for circuits (Reiter theory)
- Efficient (solve many unsolved problems)
- Applicable to more frameworks (e.g. hybrid systems)

Example: System

TransGrid Network

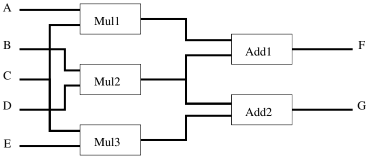
- 10k components



Example: Observation

Alarm Log (extract)

```
Date System_Time Event Text
2/07/2009 10:47:27 BAYSWTR PS  023 NO4 GEN       UNIT STATUS          OFF
2/07/2009 10:47:27 BAYSWTR330  330 SYD WEST    322 CB                 --OPENED--
2/07/2009 10:47:27 BAYSWTR330  330 NO4 BY/CUP 5042 CB                 --OPENED--
2/07/2009 10:47:27 BAYSWTR330  330 NO4 GEN TX 5242 CB                 --OPENED--
2/07/2009 10:47:27 BAYSWTR330   CONTROL SYSTEM LAN FAULT              ALARM
2/07/2009 10:47:27 BAYSWTR PS  023 NO4 GEN    2242 CB                 --OPENED--
2/07/2009 10:47:28 LIDDELL330  330 BAYSWTR330  332 CB                 --OPENED--
2/07/2009 10:47:28 LIDDELL330  330 BAYSWTR330  342 CB                 --OPENED--
2/07/2009 10:47:28 LIDDELL330  330 NO2 BY/CUP 5022 CB                 --OPENED--
2/07/2009 10:47:28 LIDDELL330  330 NO3 BY/CUP 5032 CB                 --OPENED--
2/07/2009 10:47:28 WANG330      FAULT RECORDER OPERATED               ALARM
2/07/2009 10:47:28 BAYSWTR330  330 MAIN BUS BAR   KV                  Limit 5 Low
2/07/2009 10:47:28 BAYSWTR330  330 GEN BUS BAR    KV                  Limit 5 Low
2/07/2009 10:47:28 WANG330      BU SUBSTATION MISC EQUIPMENT FAIL     ALARM
2/07/2009 10:47:28 SYD WEST    330 BAYSWTR330  322B B CB              --OPENED--
2/07/2009 10:47:28 SYD WEST    330 BAYSWTR330  322A A CB              --OPENED--
2/07/2009 10:47:28 MT PIPR330   330 FAULT RECORDER OPERATED           ALARM
2/07/2009 10:47:28 ERARING500   SUBSTATION MISC EQUIP FAIL            ALARM
2/07/2009 10:47:28 MT PIPR330  500 B BUS BAR      KV                  Limit 3 Low
2/07/2009 10:47:28 BAYSWTR330  330 NO3 BY/CUP 5032 CB                 --OPENED--
2/07/2009 10:47:28 BAYSWTR330  330 NO3 GEN TX 5232 CB                 --OPENED--
2/07/2009 10:47:28 BAYSWTR330  330 REGENTVILE  312 CB                 --OPENED--
2/07/2009 10:47:28 BAYSWTR PS  023 NO3 GEN    2232 CB                 --OPENED--
```



A = B = E = 3 F = 10

C = D = 2 G = 12

- **Model** Formula Φ_M involving *Ab* literals
- **Observation** Formula Φ_O
- **Possible behaviours** $\Phi_M \wedge \Phi_O$
- **Diagnosis** Projection on the *Ab* literals: $\exists X.\Phi_M \wedge \Phi_O$ where X are the non *Ab* literals, rewriten in prime implicants

$$Ab(Mul1) \vee Ab(Add1) \vee (Ab(Mul2) \wedge Ab(Mul3)) \vee (Ab(Mul2) \wedge Ab(Add2))$$

NICTA

AUTOMATON

SEQUENCE OF OBSERVATIONS

- **Model** Language $\mathcal{L}_M$ involving Σ_f events
- **Observation** Language $\mathcal{L}_O$ involving only observable events Σ_O
- **Possible behaviours** $\mathcal{L}_M \cap \mathcal{L}_O$
- **Diagnosis** Projection on the Σ_f events and minimisation (removes non minimal words)

$$\mathcal{L}_\Delta = Minimisation(Proj_{\Sigma_f}(\mathcal{L}_M \cap \mathcal{L}_O))$$

NICTA

Static Systems

- **Model** Formula Φ_M
- **Observation** Formula Φ_O
- **Possible behaviours** $\Phi_M \wedge \Phi_O$
- **Diagnosis** Projection on the *Ab* literal + prime implicants

Discrete Event Systems

- **Model** Language $\mathcal{L}_M$
- **Observation** Language $\mathcal{L}_O$
- **Possible behaviours** $\mathcal{L}_M \cap \mathcal{L}_O$
- **Diagnosis** Projection on the Σ_f events and minimisation

MBD: Issue

Boum!

MBD: Issue

Static Systems

The size of the formula is exponential in the number of state variables
$\rightarrow$ Compilation Map (Darwiche et al.), BDD, sd-DNNF, Cone-based diagnoser, etc.

Boum!

MBD: Issue

Static Systems

The size of the formula is exponential in the number of state variables
→ Compilation Map (Darwiche et al.), BDD, sd-DNNF, Cone-based diagnoser, etc.

Boum!

DES

The size of the automata is exponential in the number of components
→ Decentralised / Distributed approach, Junction Trees, Specialised diagnosers, etc.

Consistency-Based MBD

Check carefully-chosen hypotheses until the diagnosis is found

- → We do not compute all diagnosis candidates
- → We compute only one representative of each candidate
- → For each test, we derive useful information from the hypothesis at hand

Testing if a Hypothesis is a Candidate

Static Systems

- Φ_h is a conjunct defined on all *Ab* literals
- *h* is a candidate iff

$$\Phi_M, \Phi_O, \Phi_h \not\models \bot$$

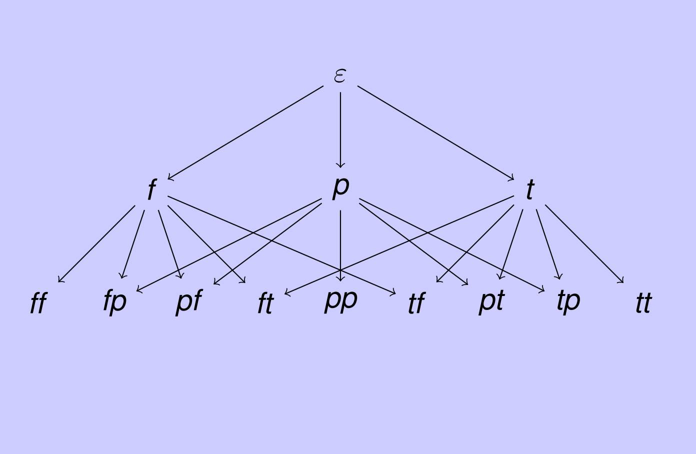
Discrete Event Systems

- $\mathcal{L}_h = \{\omega_h\}$ is a finite word defined on Σ_f
- *h* is a candidate iff

$$\mathcal{L}_M \cap \mathcal{L}_O \cap \mathcal{L}_h \neq \emptyset$$

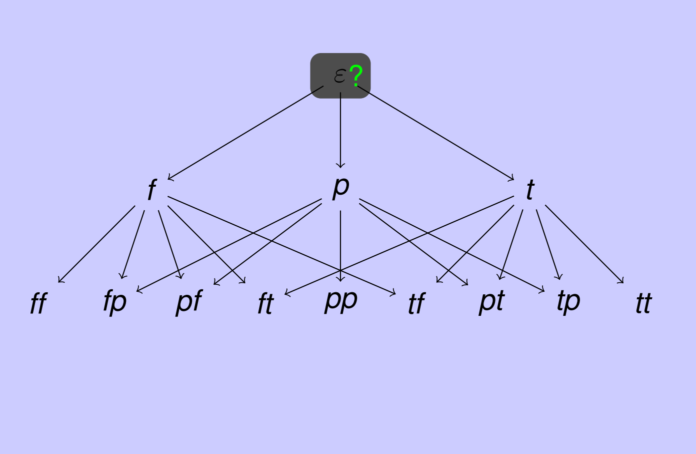
Consistency-Based MBD

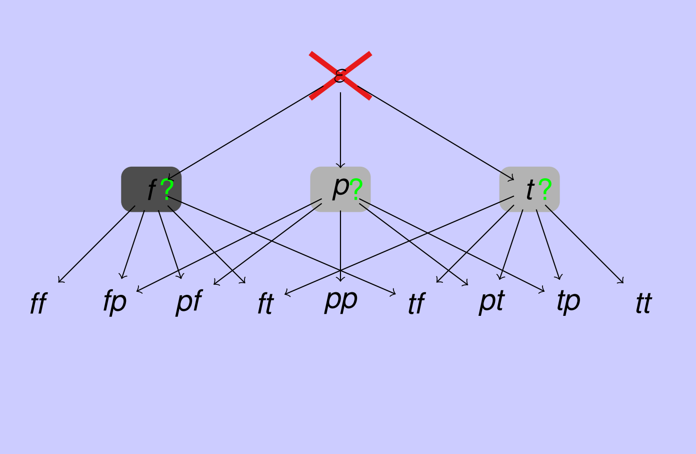
Preferred-First Strategy



Consistency-Based MBD

Preferred-First Strategy

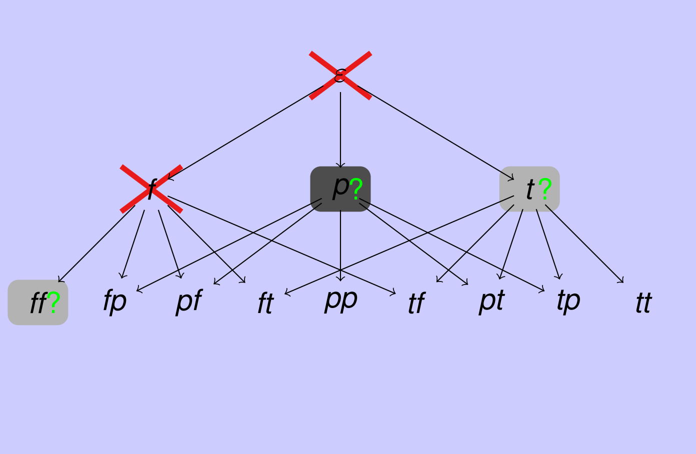




Successors of hypothesis *h* is all its children

Consistency-Based MBD

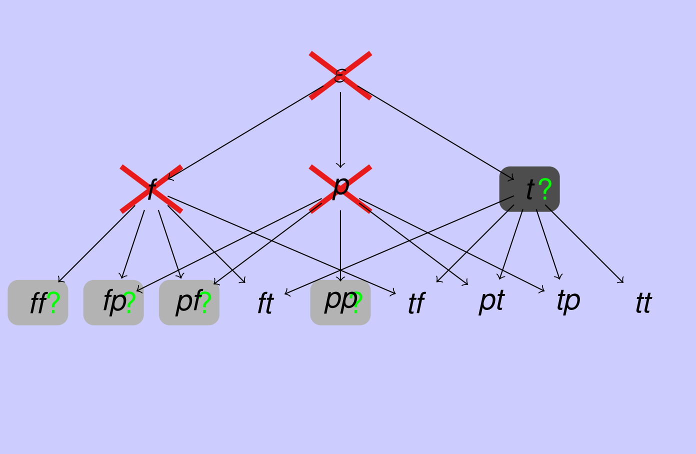
Preferred-First Strategy



But ignore successors that are covered by existing hypotheses

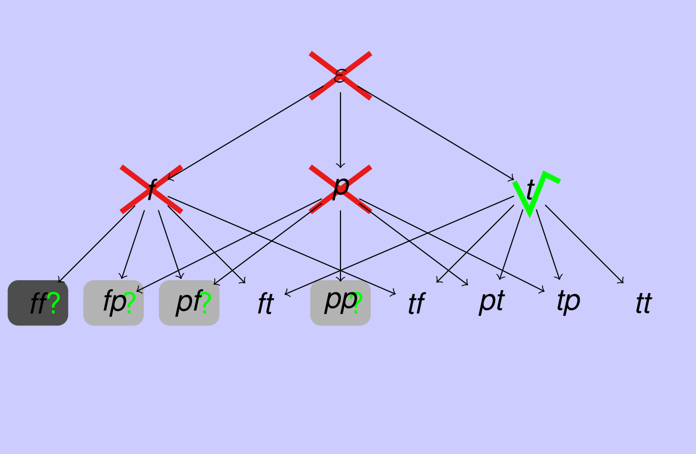
Consistency-Based MBD

Preferred-First Strategy



Consistency-Based MBD

Preferred-First Strategy



Also: termination issue (not discussed here)

Conflict

Principle

- If hypothesis *h* is not a candidate, the output is not very informative

A conflict is a generalisation of a test failure:

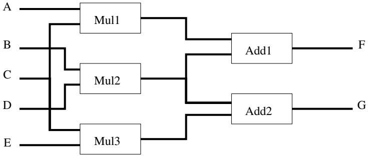
- Why did the test fail?

How to use conflicts:

- An earlier conflict may discard a new hypothesis
- Conflicts can reduce the set of successors

Conflict Example

Static System



A = B = E = 3   F = 10

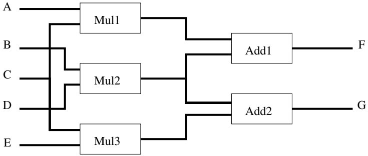
C = D = 2   G = 12

Testing if no component is abnormal:

$$\Phi_M, \Phi_O, (\neg Ab(Mul1) \wedge \neg Ab(Mul2) \wedge \neg Ab(Mul3) \wedge \neg Ab(Add1) \wedge \neg Ab(Add2)) \overset{?}{\models} \bot$$

Conflict Example

Static System



A = B = E = 3 F = 10

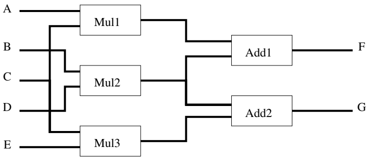
C = D = 2 G = 12

Testing if no component is abnormal:

$$\begin{array}{c} \Phi_M, \Phi_O, \\ \neg Ab(Mul1), \neg Ab(Mul2), \neg Ab(Mul3), \\ \neg Ab(Add1), \neg Ab(Add2) \end{array} \overset{?}{\models} \bot$$

Conflict Example

Static System



A = B = E = 3 F = 10

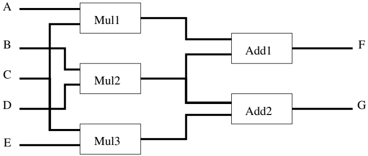
C = D = 2 G = 12

Testing if no component is abnormal:

$$\begin{array}{c} \Phi_M, \Phi_O, \\ \neg Ab(Mul1), \neg Ab(Mul2), \\ \neg Ab(Add1) \end{array} \models \bot$$

Conflict Example

Static System



A = B = E = 3 F = 10

C = D = 2 G = 12

Testing if no component is abnormal:

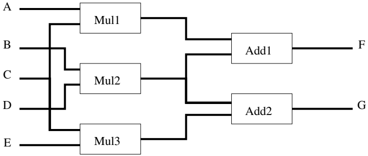
$$\left.\begin{array}{c}\Phi_M, \Phi_O, \\ \neg Ab(Mul1), \neg Ab(Mul2), \\ \neg Ab(Add1)\end{array}\right. \models \bot$$

Three successors:

- Only component *Mul*1 is abnormal
- Only component *Mul*2 is abnormal
- Only component *Add*1 is abnormal

Conflict Example

Static System



A = B = E = 3 F = 10

C = D = 2 G = 12

Testing if no component is abnormal:

$$\begin{array}{c} \Phi_M, \Phi_O, \\ \neg Ab(Mul1), \neg Ab(Mul2), \\ \neg Ab(Add1) \end{array} \models \bot$$

Three successors:

- Only component $Mul1$ is abnormal
- Only component $Mul2$ is abnormal
- Only component $Add1$ is abnormal

Conflict Generalisation to DES

If hypothesis h is not a candidate, then

$$\mathcal{L}_M \cap \mathcal{L}_O \cap \mathcal{L}_h = \emptyset \quad (1)$$

Conflict Generalisation to DES

If hypothesis h is not a candidate, then

$$\mathcal{L}_M \cap \mathcal{L}_O \cap \mathcal{L}_h = \emptyset \quad (1)$$

We reformulate $\mathcal{L}_h = \mathcal{L}_0 \cap \cdots \cap \mathcal{L}_k$

$$\mathcal{L}_M \cap \mathcal{L}_O \cap \mathcal{L}_0 \cap \cdots \cap \mathcal{L}_k = \emptyset \quad (2)$$

Conflict Generalisation to DES

If hypothesis h is not a candidate, then

$$\mathcal{L}_M \cap \mathcal{L}_O \cap \mathcal{L}_h = \emptyset \tag{1}$$

We reformulate $\mathcal{L}_h = \mathcal{L}_0 \cap \dots \cap \mathcal{L}_k$

$$\mathcal{L}_M \cap \mathcal{L}_O \cap \mathcal{L}_0 \cap \dots \cap \mathcal{L}_k = \emptyset \tag{2}$$

For some $C = \{C_0, \dots, C_{k'}\} \subseteq \{0, \dots, k\}$ (we prefer C as small as possible),

$$\mathcal{L}_M \cap \mathcal{L}_O \cap \mathcal{L}_{C_0} \cap \dots \cap \mathcal{L}_{C_{k'}} = \emptyset$$

Conflict Generalisation to DES

If hypothesis h is not a candidate, then

$$\mathcal{L}_M \cap \mathcal{L}_O \cap \mathcal{L}_h = \emptyset \tag{1}$$

We reformulate $\mathcal{L}_h = \mathcal{L}_0 \cap \cdots \cap \mathcal{L}_k$

$$\mathcal{L}_M \cap \mathcal{L}_O \cap \mathcal{L}_0 \cap \cdots \cap \mathcal{L}_k = \emptyset \tag{2}$$

For some $C = \{C_0, \ldots, C_{k'}\} \subseteq \{0, \ldots, k\}$ (we prefer C as small as possible),

$$\mathcal{L}_M \cap \mathcal{L}_O \cap \mathcal{L}_{C_0} \cap \cdots \cap \mathcal{L}_{C_{k'}} = \emptyset$$

$C =$ **conflicts**

Example

Discrete Event System

$\Sigma_f = \{a, b, c\}$ and $\mathcal{L}_h = \{a\}$

Example

Discrete Event System

$\Sigma_f = \{a, b, c\}$ and $\mathcal{L}_h = \{a\}$

$$\{a\} = \mathcal{L}_0 \cap \mathcal{L}_1 \cap \mathcal{L}_2 \cap \mathcal{L}_3 \cap \mathcal{L}_4 \cap \mathcal{L}_5$$

- $\mathcal{L}_0 = {\Sigma_f}^\star a {\Sigma_f}^\star$
- $\mathcal{L}_1 = ({\Sigma_f}^\star) \setminus ({\Sigma_f}^\star a {\Sigma_f}^\star a {\Sigma_f}^\star)$
- $\mathcal{L}_2 = ({\Sigma_f}^\star) \setminus ({\Sigma_f}^\star a {\Sigma_f}^\star b {\Sigma_f}^\star)$
- $\mathcal{L}_3 = ({\Sigma_f}^\star) \setminus ({\Sigma_f}^\star a {\Sigma_f}^\star c {\Sigma_f}^\star)$
- $\mathcal{L}_4 = ({\Sigma_f}^\star) \setminus ({\Sigma_f}^\star b {\Sigma_f}^\star a {\Sigma_f}^\star)$
- $\mathcal{L}_5 = ({\Sigma_f}^\star) \setminus ({\Sigma_f}^\star c {\Sigma_f}^\star a {\Sigma_f}^\star)$

Example

$\Sigma_f = \{a, b, c\}$ and $\mathcal{L}_h = \{a\}$

$$\{a\} = \mathcal{L}_0 \cap \mathcal{L}_1 \cap \mathcal{L}_2 \cap \mathcal{L}_3 \cap \mathcal{L}_4 \cap \mathcal{L}_5$$

- $\mathcal{L}_0 = {\Sigma_f}^\star a {\Sigma_f}^\star$
- $\mathcal{L}_1 = ({\Sigma_f}^\star) \setminus ({\Sigma_f}^\star a {\Sigma_f}^\star a {\Sigma_f}^\star)$

- $\mathcal{L}_3 = ({\Sigma_f}^\star) \setminus ({\Sigma_f}^\star a {\Sigma_f}^\star c {\Sigma_f}^\star)$
- $\mathcal{L}_4 = ({\Sigma_f}^\star) \setminus ({\Sigma_f}^\star b {\Sigma_f}^\star a {\Sigma_f}^\star)$

Conflict: $\{\mathcal{L}_0, \mathcal{L}_1, \mathcal{L}_3, \mathcal{L}_4\}$
Successors: *aa*, *ac*, and *ba*

NICTA

$\Sigma_f = \{a, b, c\}$ and $\mathcal{L}_h = \{a\}$

$$\{a\} = \mathcal{L}_0 \cap \mathcal{L}_1 \cap \mathcal{L}_2 \cap \mathcal{L}_3 \cap \mathcal{L}_4 \cap \mathcal{L}_5$$

- $\mathcal{L}_0 = {\Sigma_f}^\star a {\Sigma_f}^\star$
- $\mathcal{L}_1 = ({\Sigma_f}^\star) \setminus ({\Sigma_f}^\star a {\Sigma_f}^\star a {\Sigma_f}^\star)$

- $\mathcal{L}_3 = ({\Sigma_f}^\star) \setminus ({\Sigma_f}^\star a {\Sigma_f}^\star c {\Sigma_f}^\star)$
- $\mathcal{L}_4 = ({\Sigma_f}^\star) \setminus ({\Sigma_f}^\star b {\Sigma_f}^\star a {\Sigma_f}^\star)$

Conflict: $\{\mathcal{L}_0, \mathcal{L}_1, \mathcal{L}_3, \mathcal{L}_4\}$
Successors: aa, ac, and ba

NICTA

$\Sigma_f = \{a, b, c\}$ and $\mathcal{L}_h = \{ab\}$

Conflict:

- $\mathcal{L}_i = (\Sigma_f{}^\star) \setminus (\Sigma_f{}^\star b \Sigma_f{}^\star b \Sigma_f{}^\star)$
- $\mathcal{L}_j = (\Sigma_f{}^\star) \setminus (\Sigma_f{}^\star c \Sigma_f{}^\star)$

Successors: *abb*, *bab*, *abc*, *acb*, and *cab*

NICTA

$\Sigma_f = \{a, b, c\}$ and $\mathcal{L}_h = \{ab\}$

Conflict:

- $\mathcal{L}_i = (\Sigma_f{}^\star) \setminus (\Sigma_f{}^\star b \Sigma_f{}^\star b \Sigma_f{}^\star)$
- $\mathcal{L}_j = (\Sigma_f{}^\star) \setminus (\Sigma_f{}^\star c \Sigma_f{}^\star)$

Successors: *abb*, *bab*, *abc*, *acb*, and *cab*

Technically. . .

- Given a hypothesis h, define properties
  - $p_{\text{desc}}(h)$: property satisfied by all hypotheses $h' \succeq h$
  - $p_{\cancel{\text{desc}}}(h)$: property satisfied by all hypotheses $h' \not\succeq h$
- A possible decomposition of $\{h\}$:
  - $p_{\text{desc}}(h)$
  - $\forall h' \in \text{children}(h),\ p_{\cancel{\text{desc}}}(h')$
- $C = \{p_1, \ldots, p_k\}$ is a conflict for h iff
  - $\forall h' : p_{\text{desc}}(h') \in C \Rightarrow h' \preceq h$
  - $\forall h' : p_{\cancel{\text{desc}}}(h') \in C \Rightarrow h' \not\preceq h$
- Successors of conflict $C = \{p_1, \ldots, p_k\}$
  - Let $\Omega = \{h' \mid p_{\cancel{\text{desc}}}(h') \in C\}$
  - Successors: $\bigcup_{h' \in \Omega}(h \otimes h')$

Experiments

NICTA

Diagnosis Problem

- Electricity transmission network
- Alarm log
- Hypothesis: a sequence of “unexplained” events

Problem Instances

Metrics

- Number of components: 3 to 105
- Component model:
  - 8 to $1,024$ (more often) states
  - 44 to $92,800$ transitions
- Number of minimal candidates: up to 27 and more

Results

	N	*M*	*C*	*A*	PF	JT
window-250	1	0	2	3	0.3	1.5
chunk-004	1	2	3	3	0.8	2
chunk-056	1	4	4	7	1.7	2.6
window-618	1	0	6	2	0.7	–time–
window-527	2	1	11	8	2.7	–time–
window-347	4	9	32	13	106.1	–time–
window-336	?	?	58	49	–time–	–time–
window-335	?	?	67	66	–time–	–time–
chunk-089	?	?	105	146	–time–	–memory–
window-410	?	?	19	13	–time–	5
window-409	?	?	22	14	–time–	5.3
Nb problems solved (/129)					**116**	35

N: number of minimal candidates,
M: maximum number of faults in a minimal candidate,
C: number of components in the problem,
A: number of alarms,
PF: runtime for PF running SAT, and
JT: runtime for automata-based approach (in seconds)

Conclusion

Contribution

A generalised perspective of conflicts for non trivial hypothesis search space.

Extensions

- Application to hybrid systems
- Conflicts = explanations