

# Diagnosis of Pushdown Systems

Sophie Pinchinat, IRISA, university of Rennes

(jw with Christophe Morvan, Institut Gaspard Monge,  
university of Marne La Vallée)

January 7, 2009

# Discrete-event Systems

$$\mathcal{S} = \langle \Sigma, S, s^0, \delta, Prop, [\cdot] \rangle$$

- $\Sigma$  alphabet,
- $S$  states,  $s^0 \in S$
- $\delta : S \times \Sigma \rightarrow S$
- $[\cdot] : Prop \rightarrow 2^S$



$a_1 a_2 \dots \in \Sigma^\infty$  is an **execution** whenever  $s^0 = s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} \dots$  in  $\mathcal{S}$ .

Use  $u, u', \dots$  (resp.  $w, w', \dots$ ) for finite (resp. infinite) executions.

## Partial Observation Setting

- Partition  $\Sigma$ : **observables**,  $\Sigma_o$  and **unobservables**,  $\overline{\Sigma_o}$
- $\pi : \Sigma \rightarrow \Sigma_o$  canonical projection
- **Observation**  $\theta \in \Sigma_o^*$
- **Indistinguishable** executions **match** the same observation

$$\pi(u) = \theta = \pi(u')$$

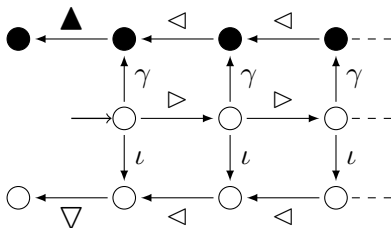
- **Information set**  $\mathcal{I} \subseteq S$  is a set of states reached from  $s^0$  by indistinguishable executions (in  $\Sigma^*\Sigma_o$ )

## Faulty (finite) executions

Distinguish  $f \in Prop$  and assume  $\llbracket f \rrbracket$  is a trap:

$$\delta(s, a) \in \llbracket f \rrbracket, \text{ for all } s \in \llbracket f \rrbracket, a \in \Sigma$$

$u$  is **faulty** if  $u$  leads to an  $f$ -marked state:  $\delta(s^0, u) \in \llbracket f \rrbracket$



$\llbracket f \rrbracket$  is the set of ●'s

## Equivocalness

An information set  $\mathcal{I}$  is

- **clear** if either all or none of its states are marked by  $f$

$$(\mathcal{I} \subseteq \llbracket f \rrbracket) \vee (\mathcal{I} \cap \llbracket f \rrbracket = \emptyset)$$

- **equivocal** otherwise

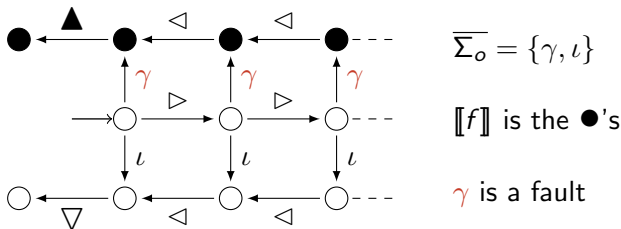
An observation  $\theta$  has type

- **clearly-faulty** if  $\delta(s^0, \theta) \subseteq \llbracket f \rrbracket$
- **clearly-nonfaulty** if  $\delta(s^0, \theta) \cap \llbracket f \rrbracket = \emptyset$
- **equivocal** if  $\delta(s^0, \theta)$  is equivocal

# Diagnosis and Diagnoser

**Diagnosis** is an informal notion that captures the ability to detect faulty sequences.

The **diagnoser** is a device that reads an observation and returns its type.



# Qualitative and quantitative aspects in diagnosis

Input:  $\mathcal{S}$ ,  $\Sigma_o$ , and  $f$

- The **Diagnosability** decision problem  
Output **YES** whenever equivocallness never lasts for ever
- The **Bounded-Latency** decision problem  
Output **YES** whenever there exists  $N$  (to be computed) such that equivocallness lasts no more than  $N$  units of time

# Diagnosability

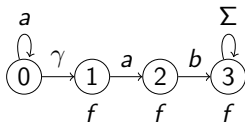
$\mathcal{S}$  is **diagnosable** (w.r.t.  $\Sigma_o$  and  $f$ ) if every infinite observation of an infinite faulty execution has a clear finite prefix.

( $\Leftrightarrow$  there is a case where equivocality lasts forever)

**Rmk1** we assume infinite executions do **diverge**: only fair behaviours of the system w.r.t. observability.

**Rmk2** non-faulty executions may yield arbitrarily long equivocal observations

$$\Sigma = \{a, b, \gamma\} \text{ and } \Sigma_o = \{a, b\}$$

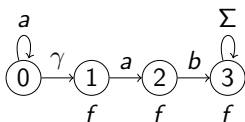


## Bounded-Latency

- A **just-clearly-faulty** observation is a clearly-faulty observation whose longest strict prefix is equivocal. Given an equivocal observation  $\theta$ , the **latency** for  $\theta$  is

$$\ell(\theta) := \max \{ |\vartheta|, \theta\vartheta \text{ is just-clearly-faulty} \}$$

- A system is **bounded-latency** if there exists  $N \in \mathbb{N}$  such that  $\ell(\theta) \leq N$ , for every equivocal observation  $\theta$ .



The **bounded-latency value** is the least such  $N$ ; here it is 1.

## Diagnosability and Bounded-Latency

only depend on  
the set of executions  
of the system

## The case of finite-state systems: the main theorems

[M. Sampath, R. Sengupta, S. Lafortune, K. Sinaamohideen, and D. Teneketzis, 1996]

Diagnosability is decidable.

[S. Jiang, Z. Huang, V. Chandra, and R. Kumar, 2001]

[T-S. Yoo and S. Lafortune]

Diagnosability is in PTIME (quadratic).

[J. Rintanen, 2007]

Diagnosability is NLOGSPACE-complete.

[T. Jéron, H. Marchand, S. Pinchinat, M-O. Cordier, 2006]

Diagnosability implies bounded-latency.

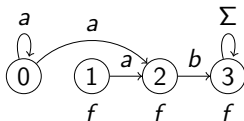
## The Algorithm Principle

**Proposition** The system is not diagnosable if, and only if, there exist two indistinguishable infinite executions  $w_1$  and  $w_2$  such that  $w_1$  is faulty but  $w_2$  is not.

### Construction

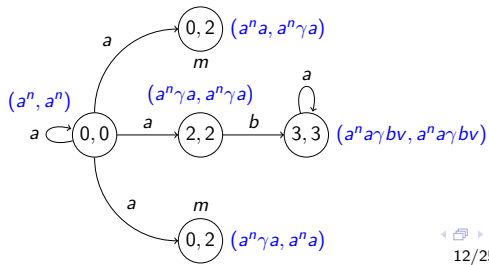
- $\pi(\mathcal{S})$  the projection onto  $\Sigma_o$ :  $p \xrightarrow{a} p'$  if  $\exists p'', p \xrightarrow{\Sigma_o^*} p'' \xrightarrow{a} p'$

$0 \xrightarrow{a} 2$  if  $0 \xrightarrow{\gamma} 1 \xrightarrow{a} 2$



- Take  $\pi(\mathcal{S}) \times \pi(\mathcal{S})$ : Each path correspond to a pair  $(u_1, u_2)$  of indistinguishable executions

Mark compound states by  $m$  whenever exactly one of the two is marked by  $f$ , and find an infinite path with globally the mark  $m$  at some point



## How about infinite-state systems?

We need a class of models with effective methods to compute/decide

- their **projection**, for observations
- their **product**, for indistinguishable pairs of executions
- the **existence of an infinite path**, for diagnosability
- the **finiteness of their set of executions**, for bounded-latency

In this talk, we focus on **(first-order) pushdown systems**.

# Pushdown automata

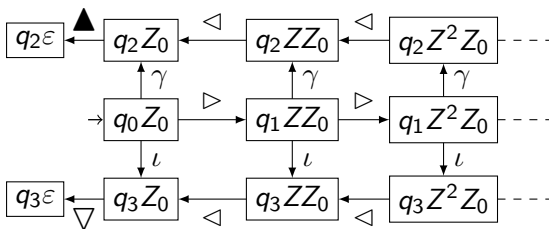
A **pushdown automaton PDA** is  $\mathcal{A} = (\Sigma, \Gamma, Q, q_0, F, \Delta)$

- $\Sigma$  **input** symbols and  $\Gamma$  **stack** symbols ( $Z_0 \in \Gamma$ )
- $Q$  finite set of states,  $q_0 \in Q$ ,  $F \subseteq Q$  final states,
- $\Delta \subseteq (Q \times \Gamma) \times (\Sigma \cup \{\epsilon\}) \times (Q \times \Gamma^*)$

# Pushdown Systems PDS

$$\Delta \subseteq Q \times \Gamma \times \Sigma_\varepsilon \times Q \times \Gamma^*$$

$$\text{Transitions } \left\{ \begin{array}{l} q_0 Z_0 \xrightarrow{\triangleright} q_1 Z Z_0, \\ q_1 Z \xrightarrow{\triangleright} q_1 Z Z, q_1 Z \xrightarrow{\gamma} q_2 Z, \dots \\ q_2 Z \xrightarrow{\triangleleft} q_2 \varepsilon, q_2 Z_0 \xrightarrow{\blacktriangle} q_2 \varepsilon \dots \end{array} \right.$$

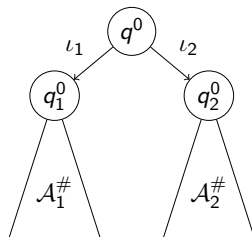


Executions are  $\triangleright^n \ell \triangleleft^n \nabla$ , and  $\triangleright^n \gamma \triangleleft^n \blacktriangle$  (faulty with  $q_2$  final)

# The diagnosability of PDS is undecidable

(Proof sketch)

- $\mathcal{A}_1$  and  $\mathcal{A}_2$  PDA over  $\Sigma_1$  and  $\Sigma_2$
- $\Sigma = \Sigma_1 \cup \Sigma_2 \cup \{\iota_1, \iota_2, \#\}$
- $\mathcal{A}_i^\#$  the PDA such that  $L(\mathcal{A}_i) \# \Sigma^*$
- Put  $f$  in the  $\mathcal{A}_1^\#$  part



The PDS is diagnosable w.r.t. observables  $\Sigma \setminus \{\iota_1, \iota_2\}$  and mark  $f$  if, and only if,

$$L(\mathcal{A}_1) \cap L(\mathcal{A}_2) = \emptyset \quad \text{UNDECIDABLE}$$

## Interpretation of this undecidability result

### Proposition (a classic!)

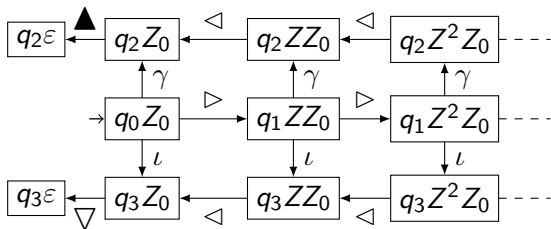
*Any PDA is equivalent to a PDA with no  $\varepsilon$ -transitions (+The construction is effective)*

- So projection is fine (replace unobservable by  $\varepsilon$  and proceed)
- Different for higher-order PDS
- Likely due to the inability to build the product of PDA

⇒ Visibly Pushdown Automata [R. Alur and P. Madhusudan, 2004]

# Visibly Pushdown Automata

- A **visibly pushdown automaton VPA** is a PDA with  $\Sigma := \Sigma_{push} \cup \Sigma_{pop} \cup \Sigma_{int}$  and transition types accordingly



$$\Sigma_{push} = \{\blacktriangleright\}, \Sigma_{pop} = \{\blacktriangleleft, \blacktriangle, \blacktriangledown\}, \Sigma_{int} = \{\gamma, \ell\}$$

- $[\Sigma_{push}, \Sigma_{pop}, \Sigma_{int}]$ -VPA's can be synchronized (product)

# The diagnosability of VPS is undecidable

- because

**Proposition** Any CF language is the projection of a VP language (i.e. accepted by a VPA) with  $\Sigma_{int} = \emptyset$ .

and we already have seen the undecidability of the diagnosability for PDS

- **Interpretation of this result:** Projection cannot be chosen arbitrarily

## A decidable case for diagnosability

Use  $[\Sigma_{int}]$ -VPS to emphasize  $\Sigma_{int}$

### Theorem

*Diagnosability w.r.t. observables  $\Sigma_o$  and proposition  $f$  is decidable over any class of  $[\Sigma_{int}]$ -VPS whenever*

$$\left\{ \begin{array}{l} \overline{\Sigma_o} \subseteq \Sigma_{int} \\ \text{and } f \text{ marks a regular set of configurations} \end{array} \right.$$

(Proof sketch)

- **Proposition**

The projection of a  $[\Sigma_{int}]$ -VPA onto  $\Sigma'$ , with  $\overline{\Sigma'} \subseteq \Sigma_{int}$ , is equivalent to a  $[\Sigma_{int} \cap \Sigma']$ -VPA (The construction is effective).

- Project the VPA, determinize it, build the self-product, and finally decide existence of an infinite “equivocal” path.

# Bounded-Latency

## Theorem

*Given a  $[\Sigma_{int}]$ -VPS  $\mathcal{S}$ , an observation alphabet  $\Sigma_o$  with  $\overline{\Sigma_o} \subseteq \Sigma_{int}$ , and a proposition  $f$  which marks a regular set of configurations, it is decidable whether  $\mathcal{S}$  is bounded-latency or not. Furthermore, the bound can be effectively computed.*

(Proof sketch)

- $L := \{\vartheta \in \Sigma_o^* \mid \exists \theta \text{ equivocal, } \theta\vartheta \text{ is a just-clearly-faulty}\}$  is finite if, and only if,  $\mathcal{S}$  is bounded-latency.
- **Lemma**  $L$  is a VP language
- If  $L$  is finite, the bounded-latency value is  $\max\{|\vartheta|, \vartheta \in L\}$ .
- Recall that [the finiteness of CF languages is decidable](#)

## Higher-order PDS

$\{\varepsilon\text{-CLOSURE OF HPDS}\} \stackrel{[CW03]}{=} \{\text{GRAPHS CAUCAL HIERARCHY}\}$

REACH SOME CONTEXT-SENSITIVE LANGUAGES, e.g.  $a^n b^n c^n$

- Diagnosability: consider **higher-order VPA** [Shaji Illias, 2005], and use [A. Carayol and S. Woerhle, 2003] to decide the existence of an infinite ambiguous path:

### Theorem

*For any class of  $k$ -order  $[\Sigma_{int}]$ -VPS, diagnosability w.r.t. the set of observables  $\Sigma_o$  and the proposition  $f$  is decidable, whenever  $\overline{\Sigma_o} \subseteq \Sigma_{int}$  and  $f$  marks a regular set of configurations.*

- Bounded-Latency is more involved

## Bounded-Latency

- Again, decide the finiteness of  $L = \{\vartheta \in \Sigma_o^* \mid \exists \theta \text{ equivocal, } \theta\vartheta \text{ is a just-clearly-faulty}\}$
- **Proposition**  
The finiteness of a **real-time** HPD language is decidable.
- An HVPA for  $L$ , but not real-time ( $\varepsilon$ -transitions)
- It is open whether HPDA are real-time (conj. neg. by Carayol)

THE FINITENESS OF HPD LANGUAGES  
IS A DIFFICULT QUESTION

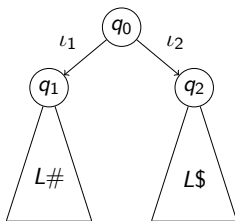
## Proposition

Let  $\mathcal{L}$  be a class of pushdown languages which contains finite languages, and which is closed under concatenation and union. If the language finiteness problem is undecidable on  $\mathcal{L}$ , so is the bounded-latency problem on the class of systems whose set of executions is in  $\mathcal{L}$ .

(Proof sketch)

$L \in \mathcal{L} \rightsquigarrow \mathcal{S}_L$  **diagnosable**,  $\mathcal{S}_L$  is bounded-latency  $\Leftrightarrow L$  is finite.

- $\Sigma \cup \{\iota_1, \iota_2, \#, \$\}$
- By construction,  $\text{Exec}(\mathcal{S}_L) \in \mathcal{L}$ .
- Unobservables  $\iota_1$  and  $\iota_2$
- $f$  which marks the configurations of the  $L\$$  component



# Perspectives

Extension to other classes of infinite systems where projections and products can be effectively computed

- Visibly transducers [Caucal06]
- Synchronization grammars [Caucal07]



M. Sampath, R. Sengupta, S. Lafortune, K. Sinaamohideen,  
and D. Teneketzis.

Diagnosability of discrete event systems.

*IEEE Transactions on Automatic Control*, 40(9):1555–1575,  
1995.