Diagnosis in Infinite-State Probabilistic Systems

Nathalie Bertrand[1], Serge Haddad[2], Engel Lefaucheux[1,2]

1 Inria Rennes, France
2 LSV, ENS Cachan & CNRS & Inria, France

CONCUR 2016, Québec City

Why Diagnosis?

Faults and/or failures are unavoidable for some systems:

- Components have a finite lifetime;
- Reactive systems suffer pathological behaviour of the environment.

Why Diagnosis?

Faults and/or failures are unavoidable for some systems:

- Components have a finite lifetime;
- Reactive systems suffer pathological behaviour of the environment.

Consequences of unhandled faults may be critical:

- Human casualties (e.g. pacemaker);
- Financial losses (e.g. mission to Mars).

Why Diagnosis?

Faults and/or failures are unavoidable for some systems:

- Components have a finite lifetime;
- Reactive systems suffer pathological behaviour of the environment.

Consequences of unhandled faults may be critical:

- Human casualties (e.g. pacemaker);
- Financial losses (e.g. mission to Mars).

Necessity of a reactive and sound diagnoser.

Which Features for System Models?

Probabilistic Behaviour

- Obtained by analysing a set of observations from a partially observable system;
- Already defined by design (e.g. Ethernet protocol).

Which Features for System Models?

Probabilistic Behaviour

- Obtained by analysing a set of observations from a partially observable system;
- Already defined by design (e.g. Ethernet protocol).

Infinite number of states

- Open systems (requests, threads, etc.);
- Dynamic data structures (stack, queue, etc.).

Outline

Diagnosability specifications

Characterising diagnosability for infinite-state systems

Deciding diagnosability of visibly pushdown models

Outline

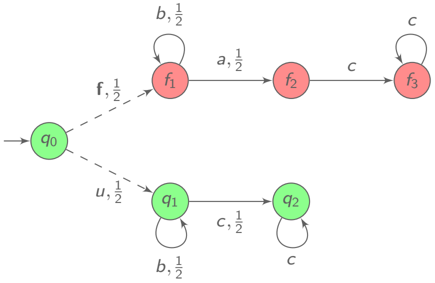
Diagnosability specifications

Characterising diagnosability for infinite-state systems

Deciding diagnosability of visibly pushdown models

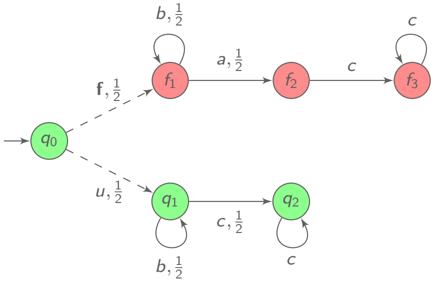
Fault Diagnosis in Probabilistic systems

Diagnoser: must tell whether a fault **f** occurred, based on observations.



Fault Diagnosis in Probabilistic systems

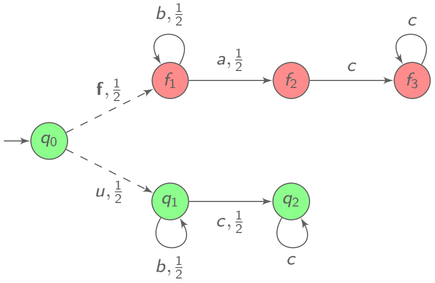
Diagnoser: must tell whether a fault $\mathbf{f}$ occurred, based on observations.



✓ c is *surely correct* since $\mathcal{P}^{-1}(c) = \{q_0 u q_1 c q_2\}$.

Fault Diagnosis in Probabilistic systems

Diagnoser: must tell whether a fault $\mathbf{f}$ occurred, based on observations.

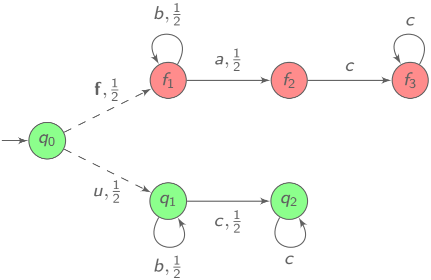


✓ c is *surely correct* since $\mathcal{P}^{-1}(c) = \{q_0 u q_1 c q_2\}$.

✗ ac is *surely faulty* since $\mathcal{P}^{-1}(ac) = \{q_0 \mathbf{f} f_1 a f_2 c f_3\}$.

Fault Diagnosis in Probabilistic systems

Diagnoser: must tell whether a fault $\mathbf{f}$ occurred, based on observations.



✓ c is *surely correct* since $\mathcal{P}^{-1}(c) = \{q_0 u q_1 c q_2\}$.

✗ ac is *surely faulty* since $\mathcal{P}^{-1}(ac) = \{q_0 \mathbf{f} f_1 a f_2 c f_3\}$.

? b is *ambiguous* since $\mathcal{P}^{-1}(b) = \{q_0 \mathbf{f} f_1 b f_1, q_0 u q_1 b q_1\}$.

Diagnosis of Probabilistic Systems

Diagnoser requirements:

- Soundness: if a fault is claimed, a fault occurred
- Reactivity: every fault is eventually almost surely detected

Diagnosis of Probabilistic Systems

Diagnoser requirements:

- Soundness: if a fault is claimed, a fault occurred
- Reactivity: every fault is eventually almost surely detected

A decision problem (*diagnosability*): does there exist a diagnoser?

A synthesis problem: how to build a diagnoser?

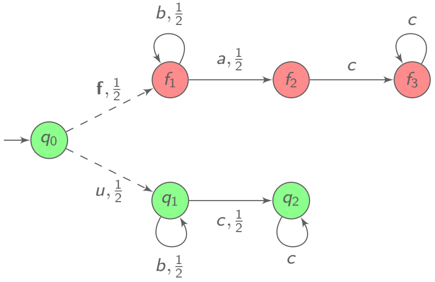
Diagnosis of Probabilistic Systems

Diagnoser requirements:

- Soundness: if a fault is claimed, a fault occurred
- Reactivity: every fault is eventually almost surely detected

A decision problem (*diagnosability*): does there exist a diagnoser?

A synthesis problem: how to build a diagnoser?



A sound and reactive diagnoser: claim a fault when *a* occurs.

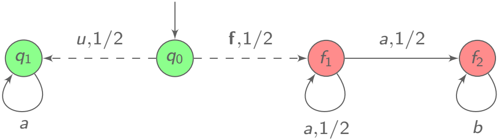
Specifying diagnosability for probabilistic systems

Two discriminating criteria:

Specifying diagnosability for probabilistic systems

Two discriminating criteria:

1. Detect faults, or tell whether a run is faulty or correct?



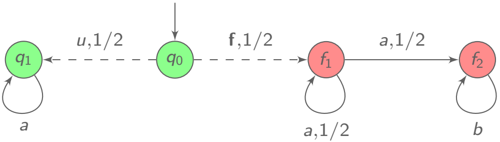
Fault is almost surely followed by occurrence of b.

Ambiguous sequences have probability $\frac{1}{2}$.

Specifying diagnosability for probabilistic systems

Two discriminating criteria:

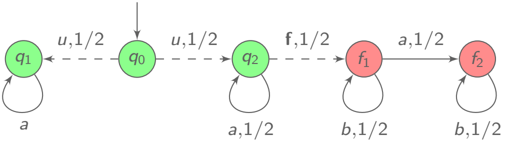
1. Detect faults, or tell whether a run is faulty or correct?



Fault is almost surely followed by occurrence of b.

Ambiguous sequences have probability $\frac{1}{2}$.

2. Consider infinite observed sequences or their finite prefixes?



Infinite sequence a^ω is surely correct.

For every n, a^n is ambiguous, and has probability greater than $\frac{1}{2}$.

Four diagnosability specifications [BHL 14]

[BHL 14] Bertrand, Haddad and Lefaucheux, *Foundation of Diagnosis and Predictability in Probabilistic Systems*, FSTTCS'14

Diagnosability	All runs		Faulty runs
Finite prefixes	FA	$\Rightarrow$ $\not\Leftarrow$	FF
	$\Downarrow$ $\not\Uparrow$		$\Downarrow\Uparrow^*$
Infinite sequences	IA	$\Rightarrow$ $\not\Leftarrow$	IF

* assuming finite-branching

Four diagnosability specifications [BHL 14]

[BHL 14] Bertrand, Haddad and Lefaucheux, *Foundation of Diagnosis and Predictability in Probabilistic Systems*, FSTTCS'14

Diagnosability	All runs		Faulty runs
Finite prefixes	FA	$\Rightarrow$ $\not\Leftarrow$	FF
	$\Downarrow \not\Uparrow$		$\Downarrow\Uparrow^*$
Infinite sequences	IA	$\Rightarrow$ $\not\Leftarrow$	IF

* assuming finite-branching

Complexity for finite-state models

All diagnosability problems are PSPACE-complete.

Diagnoser synthesis is in EXPTIME.

Four diagnosability specifications [BHL 14]

[BHL 14] Bertrand, Haddad and Lefaucheux, *Foundation of Diagnosis and Predictability in Probabilistic Systems*, FSTTCS'14

Diagnosability	All runs		Faulty runs
Finite prefixes	FA	$\Rightarrow$ $\not\Leftarrow$	FF
	$\Downarrow$ $\not\Uparrow$		$\Downarrow\Uparrow^*$
Infinite sequences	IA	$\Rightarrow$ $\not\Leftarrow$	IF

* assuming finite-branching

Complexity for finite-state models
All diagnosability problems are PSPACE-complete.
Diagnoser synthesis is in EXPTIME.

What about infinite-state probabilistic systems?

Outline

Diagnosability specifications

Characterising diagnosability for infinite-state systems

Deciding diagnosability of visibly pushdown models

Quest for a characterisation

Objective: a simple qualitative characterisation independent of probability values

$\mathcal{N}$ is diagnosable iff $\mathbb{P}_{\mathcal{N}}(B) \bowtie p$, where:

- $p \in \{0, 1\}$, $\bowtie \in \{<, =, >\}$;
- $(\star)$ B belongs to a low level of Borel hierarchy and $(\star)$ B only depends on the underlying LTS.

Quest for a characterisation

Objective: a simple qualitative characterisation
independent of probability values

$\mathcal{N}$ is diagnosable iff $\mathbb{P}_{\mathcal{N}}(B) \bowtie p$, where:

- $p \in \{0, 1\}$, $\bowtie \in \{<, =, >\}$;
- $(\star)$ B belongs to a low level of Borel hierarchy and
  $(\star)$ B only depends on the underlying LTS.

Definitions are not directly applicable:

- IA $\mathbb{P}(\text{Amb}_\infty) = 0$ Amb_∞ analytic set, a priori not Borel
- IF $\mathbb{P}(\text{FAmb}_\infty) = 0$ FAmb_∞ analytic set, a priori not Borel

Quest for a characterisation

Objective: a simple qualitative characterisation independent of probability values

$\mathcal{N}$ is diagnosable iff $\mathbb{P}_{\mathcal{N}}(B) \bowtie p$, where:

- $p \in \{0, 1\}$, $\bowtie \in \{<, =, >\}$;
- $(\star)$ B belongs to a low level of Borel hierarchy and $(\star)$ B only depends on the underlying LTS.

Definitions are not directly applicable:

•	IA	$\mathbb{P}(\text{Amb}_\infty) = 0$	Amb_∞ analytic set, a priori not Borel
•	IF	$\mathbb{P}(\text{FAmb}_\infty) = 0$	FAmb_∞ analytic set, a priori not Borel
•	FA	$\lim_{n\to\infty} \mathbb{P}(\text{Amb}_n) = 0$	$(\text{Amb}_n)_{n\in\mathbb{N}}$ family of Borel sets
•	FF	$\lim_{n\to\infty} \mathbb{P}(\text{FAmb}_n) = 0$	$(\text{FAmb}_n)_{n\in\mathbb{N}}$ family of Borel sets

Characterisation in pathL: expressive linear temporal logic

$\phi ::= \alpha \mid \neg\phi \mid \phi \wedge \phi \mid \Diamond\phi$ where α is a path formula

Characterisation in pathL: expressive linear temporal logic

$\phi ::= \alpha \mid \neg\phi \mid \phi \wedge \phi \mid \Diamond\phi$ where α is a path formula

pathL subsumes all ω-regular linear specification languages

Characterisation in pathL: expressive linear temporal logic

$$\phi ::= \alpha \mid \neg\phi \mid \phi \wedge \phi \mid \Diamond\phi \text{ where } \alpha \text{ is a path formula}$$

pathL subsumes all ω-regular linear specification languages

- $\mathfrak{f}(\rho) \equiv \rho$ faulty
- $\mathfrak{U}(\rho) \equiv \exists \rho'$ correct s.t. $\mathcal{P}(\rho) = \mathcal{P}(\rho')$

$\mathcal{N}$ is FF-diagnosable iff $\mathcal{N} \models \mathbb{P}^{=0}(\Diamond\Box(\mathfrak{U} \wedge \mathfrak{f}))$.

also valid for IF*-diagnosability if* $\mathcal{N}$ *is finitely-branching*

Characterisation in pathL: expressive linear temporal logic

$\phi ::= \alpha \mid \neg\phi \mid \phi \wedge \phi \mid \Diamond\phi$ where α is a path formula

pathL subsumes all ω-regular linear specification languages

- $\mathfrak{f}(\rho) \equiv \rho$ faulty
- $\mathfrak{U}(\rho) \equiv \exists\rho'$ correct s.t. $\mathcal{P}(\rho) = \mathcal{P}(\rho')$

$\mathcal{N}$ is FF-diagnosable iff $\mathcal{N} \models \mathbb{P}^{=0}(\Diamond\Box(\mathfrak{U} \wedge \mathfrak{f}))$.

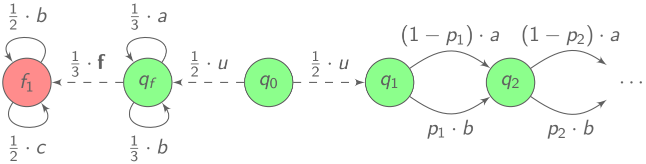
also valid for IF-*diagnosability if* $\mathcal{N}$ *is finitely-branching*

- $\mathfrak{W}(\rho) \equiv$ last observation does not change time of earliest possible fault

$\mathcal{N}$, finitely branching, is IA-diagnosable iff $\mathcal{N} \models \mathbb{P}^{=0}(\Diamond\Box(\mathfrak{U} \wedge \mathfrak{W}))$.

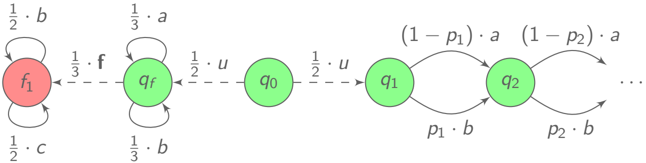
About expressiveness of the FA-diagnosability

There does not exist a F_σ set B only depending on the underlying LTS such that $\mathbb{P}(B) = 0$ characterises FA-diagnosability.

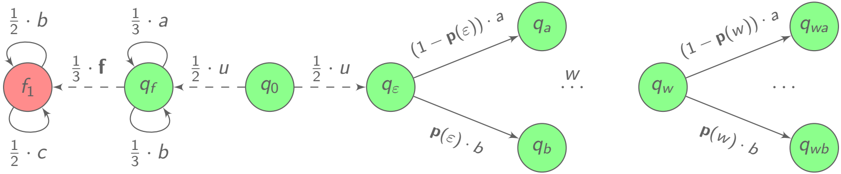


About expressiveness of the FA-diagnosability

There does not exist a F_σ set B only depending on the underlying LTS such that $\mathbb{P}(B) = 0$ characterises FA-diagnosability.



There does not exist a Borel set B only depending on the underlying LTS such that $\mathbb{P}(B) > 0$ characterises FA-diagnosability.



Outline

Diagnosability specifications

Characterising diagnosability for infinite-state systems

Deciding diagnosability of visibly pushdown models

Two issues to tackle

1. Find a decidable yet expressive model.

Two issues to tackle

1. Find a decidable yet expressive model.

   Choice of the probabilistic visibly pushdown automata.
   $\rightarrow$ Effective model checking procedures and good closure properties.

Two issues to tackle

1. Find a decidable yet expressive model.

   Choice of the probabilistic visibly pushdown automata.
   $\rightarrow$ Effective model checking procedures and good closure properties.

2. $\mathfrak{f}, \mathfrak{U}$ and $\mathfrak{W}$ are difficult to check.

Two issues to tackle

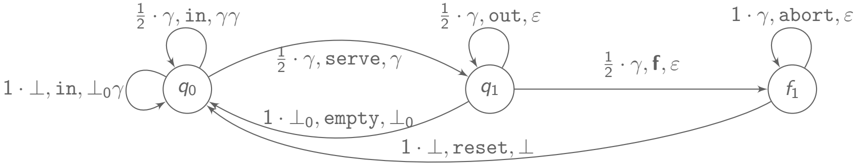
1. Find a decidable yet expressive model.

   Choice of the probabilistic visibly pushdown automata.
   $\rightarrow$ Effective model checking procedures and good closure properties.

2. $\mathfrak{f}, \mathfrak{U}$ and $\mathfrak{W}$ are difficult to check.

   Equip the model with atomic propositions reflecting the path formulae.

Probabilistic Visibly Pushdown Automata (pVPA)

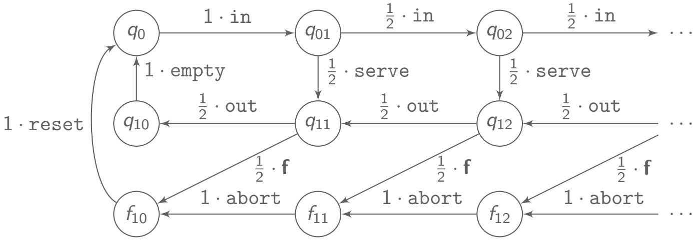


The action determines the operation on the stack.
i.e. the size of the stack is always known.

Iterative behaviour of a server.

1. A server takes an arbitrary list of requests.
2. It starts serving them until
   2.1 all of them are satisfied.
   2.2 or an error occurred then it drops all the following requests.

Semantics of pVPA



Observation of pop events: $\mathcal{P}(\texttt{out}) = \mathcal{P}(\mathbf{f}) = \mathcal{P}(\texttt{abort}) = \texttt{pop}$.

$$(q_0, |\bot_0|) \xrightarrow{\texttt{in}} (q_0, \begin{vmatrix}\gamma\\ \bot_0\end{vmatrix}) \xrightarrow{\texttt{in}} (q_0, \begin{vmatrix}\gamma\\ \gamma\\ \bot_0\end{vmatrix}) \xrightarrow{\texttt{serve}} (q_1, \begin{vmatrix}\gamma\\ \gamma\\ \bot_0\end{vmatrix}) \xrightarrow{\texttt{out}} (q_1, \begin{vmatrix}\gamma\\ \bot_0\end{vmatrix}) \xrightarrow{\texttt{out}} (q_1, |\bot_0|) \xrightarrow{\texttt{empty}} (q_0, |\bot_0|)$$

$$(q_1, \begin{vmatrix}\gamma\\ \gamma\\ \bot_0\end{vmatrix}) \xrightarrow{\mathbf{f}} (f_1, \begin{vmatrix}\gamma\\ \bot_0\end{vmatrix}) \xrightarrow{\texttt{abort}} (f_1, |\bot_0|) \xrightarrow{\texttt{reset}} (q_0, |\bot_0|)$$

Reducing to pLTL model checking on pVPA

Reduction in four steps:

- diagnosis-oriented determinisation of the pVPA into a VPA;

Reducing to pLTL model checking on pVPA

Reduction in four steps:

- diagnosis-oriented determinisation of the pVPA into a VPA;
- construction of the enlarged pVPA, a synchronized product of:
  - the deterministic VPA
  - and the original pVPA;

Reducing to pLTL model checking on pVPA

Reduction in four steps:

- diagnosis-oriented determinisation of the pVPA into a VPA;
- construction of the enlarged pVPA, a synchronized product of:
  - the deterministic VPA
  - and the original pVPA;
- translation of path formulae into atomic propositions;

Reducing to pLTL model checking on pVPA

Reduction in four steps:

- diagnosis-oriented determinisation of the pVPA into a VPA;
- construction of the enlarged pVPA, a synchronized product of:
  - the deterministic VPA
  - and the original pVPA;
- translation of path formulae into atomic propositions;
- model checking of qualitative pLTL formulae. [EY 12]

[EY 12] Etessami and Yannakakis, *Model checking recursive probabilistic systems*, ACMToCL 2012

Reducing to pLTL model checking on pVPA

Reduction in four steps:

- diagnosis-oriented determinisation of the pVPA into a VPA;
- construction of the enlarged pVPA, a synchronized product of:
  - the deterministic VPA
  - and the original pVPA;
- translation of path formulae into atomic propositions;
- model checking of qualitative pLTL formulae. [EY 12]

[EY 12] Etessami and Yannakakis, *Model checking recursive probabilistic systems*, ACMToCL 2012

FF-diagnosability, IF-diagnosability and IA-diagnosability are decidable in EXPSPACE for pVPA.

Details on the determinisation

- Inspired by original determinisation of [AM 04]
- With tags customized for diagnosis borrowed from [HHMS 13]

[AM 04] Alur and Madhusudan. *Visibly pushdown languages*, STOC'04

[HHMS 13] Haar, Haddad, Melliti and Schwoon. *Optimal constructions for active diagnosis*, FSTTCS'13.

Details on the determinisation

- Inspired by original determinisation of [AM 04]
- With tags customized for diagnosis borrowed from [HHMS 13]

stack symbol = set of tuples $\frac{\gamma, \mathrm{X}, q}{\gamma^-, \mathrm{X}^-, q^-}$ corresponding to possible runs:

• states q, q^-: q reached after the last action;
q^- reached after the last push;

• tags X, X^-: X status after last action
U = correct, V = recent fault, W = old fault;
X^- status after the last push

• original stack symbols γ, γ^-: γ the top stack symbol;
γ^- last but top stack symbol

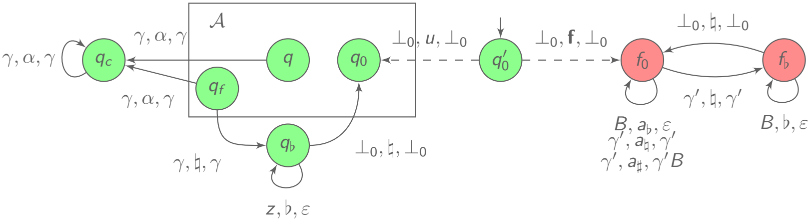
[AM 04] Alur and Madhusudan. *Visibly pushdown languages*, STOC'04

[HHMS 13] Haar, Haddad, Melliti and Schwoon. *Optimal constructions for active diagnosis*, FSTTCS'13.

Hardness of diagnosis

Diagnosability is EXPTIME-hard for pVPA.

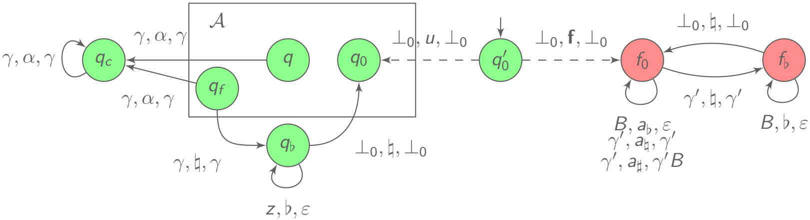
Reduction from the universality problem for VPA.



Hardness of diagnosis

Diagnosability is EXPTIME-hard for pVPA.

Reduction from the universality problem for VPA.



Diagnosability is undecidable for probabilistic pushdown automata.

Reduction from the Post Correspondence Problem.
Already holds for restricted classes of pPDA (two phases).

Conclusion

Summary of contributions

- Characterisation of diagnosability via qualitative probabilistic formulae;
- Lower and upper bounds for diagnosis of visibly pushdown systems.

Conclusion

Summary of contributions

- Characterisation of diagnosability via qualitative probabilistic formulae;
- Lower and upper bounds for diagnosis of visibly pushdown systems.

Future work

- Reduction of the complexity gap between lower and upper bounds;
- Diagnosis of other infinite state stochastic systems;
- Diagnosis for continuous-time stochastic systems.

Enlarging the pVPA

Ad-hoc determinised VPA.

A stack symbol is a set of tuples $\frac{\gamma, X, q}{\gamma^-, X^-, q^-}$ corresponding to possible runs:

- $q, q^- \in Q$
q (resp q^-) is the state reached by the run (resp. after the last push);

- $X, X^- \in \{U, V, W\}$
X (resp. X^-) is the status of the run (resp. after the last push):
U for a correct run, V for a young faulty run and W for an old faulty run;

- $\gamma, \gamma^- \in \Gamma$
γ (resp. γ^-) is the symbol on (resp. under) the top of the stack.

Enlarging the pVPA

Ad-hoc determinised VPA.

A stack symbol is a set of tuples $\frac{\gamma, \mathsf{X}, q}{\gamma^-, \mathsf{X}^-, q^-}$ corresponding to possible runs:

- $q, q^- \in Q$
q (resp q^-) is the state reached by the run (resp. after the last push);

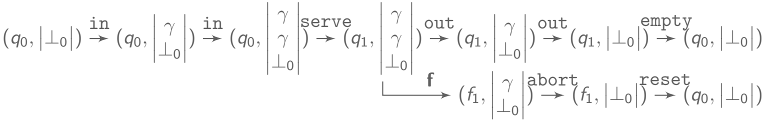
- $\mathsf{X}, \mathsf{X}^- \in \{\mathsf{U}, \mathsf{V}, \mathsf{W}\}$
X (resp. X^-) is the status of the run (resp. after the last push):
U for a correct run, V for a young faulty run and W for an old faulty run;

- $\gamma, \gamma^- \in \Gamma$
γ (resp. γ^-) is the symbol on (resp. under) the top of the stack.

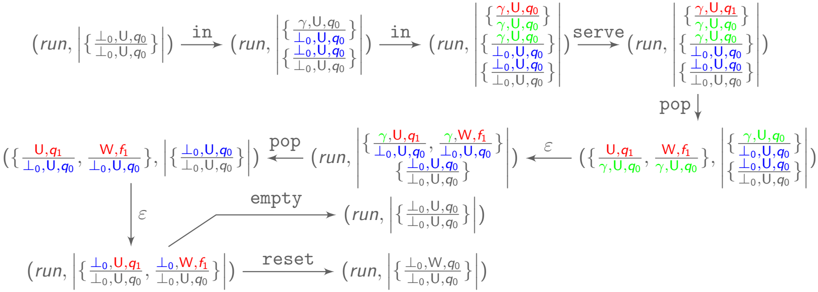
The enlarged pVPA is a synchronized product of:

- the determinised VPA and
- the original pVPA with fault memory.

From runs to observation



with $\mathcal{P}(out) = \mathcal{P}(\mathbf{f}) = \mathcal{P}(abort) = pop$.



Deciding diagnosability

The atomic proposition U and W allows to express the path formulae $\mathfrak{f}$, $\mathfrak{U}$ and $\mathfrak{W}$.

- $\mathfrak{f}$ is already an atomic proposition.

Deciding diagnosability

The atomic proposition U and W allows to express the path formulae $\mathfrak{f}$, $\mathfrak{U}$ and $\mathfrak{W}$.

- $\mathfrak{f}$ is already an atomic proposition.
- U and W encode $\mathfrak{U}$ and respectively $\mathfrak{W}$ with:
  - U occurs if and only if $\mathfrak{U}$ holds;
  - $\Box\Diamond$W occurs if and only if $\Box\Diamond\mathfrak{W}$ holds.

Deciding diagnosability

The atomic proposition U and W allows to express the path formulae $\mathfrak{f}$, $\mathfrak{U}$ and $\mathfrak{W}$.

- $\mathfrak{f}$ is already an atomic proposition.
- U and W encode $\mathfrak{U}$ and respectively $\mathfrak{W}$ with:
  - U occurs if and only if $\mathfrak{U}$ holds;
  - $\Box\Diamond$W occurs if and only if $\Box\Diamond\mathfrak{W}$ holds.

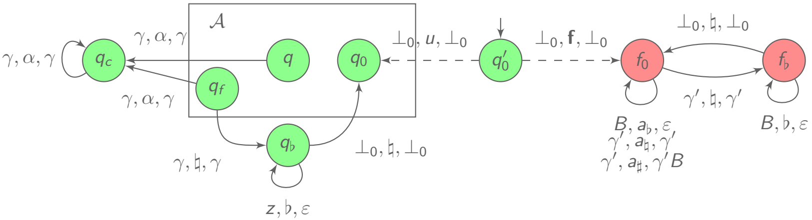
FF-diagnosability, IF-diagnosability and IA-diagnosability are decidable in EXPSPACE for pVPA.

- Boils down to pLTL model checking; [EY-TOCL12]
- Enlarged pVPA exponential in size.

Hardness of diagnosability for pushdown systems

Diagnosability is EXPTIME-hard for pVPA.

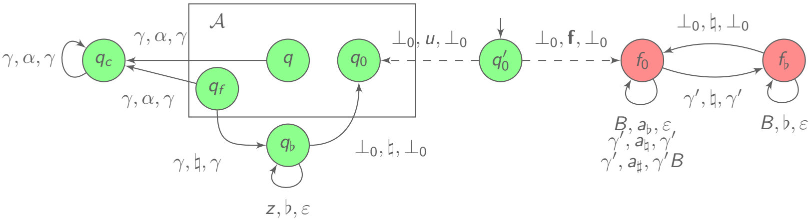
Reduction from the universality problem for VPA.



Hardness of diagnosability for pushdown systems

Diagnosability is EXPTIME-hard for pVPA.

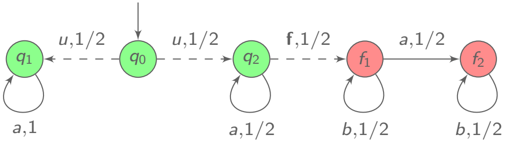
Reduction from the universality problem for VPA.



Diagnosability is undecidable for probabilistic pushdown automata.

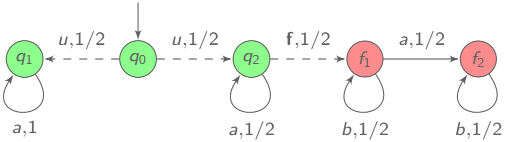
Reduction from the Post Correspondence Problem to restricted classes of probabilistic pushdown automata.

Some useful path formulae



$\mathfrak{f}(\rho) =$ **true** if ρ is faulty.
$\mathfrak{f}(\varepsilon) =$ **false**.

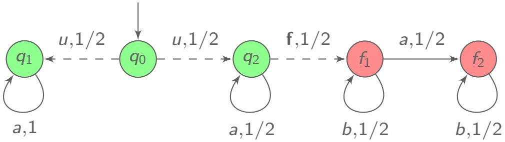
Some useful path formulae



$\mathfrak{f}(\rho) =$ **true** if ρ is faulty.
$\mathfrak{f}(\varepsilon) =$ **false**.

$\mathfrak{U}(\rho) =$ **true** if there exists a correct run ρ' with $\mathcal{P}(\rho') = \mathcal{P}(\rho)$.
$\mathfrak{U}(q_0 u q_2 \mathbf{f} f_1 a f_2) =$ **true**.

Some useful path formulae



$\mathfrak{f}(\rho) =$ **true** if ρ is faulty.
$\mathfrak{f}(\varepsilon) =$ **false**.

$\mathfrak{U}(\rho) =$ **true** if there exists a correct run ρ' with $\mathcal{P}(\rho') = \mathcal{P}(\rho)$.
$\mathfrak{U}(q_0 u q_2 \mathbf{f} f_1 a f_2) =$ **true**.

$\mathfrak{W}(\rho a q)$ is **true** if an oldest fault of $\mathcal{P}(\rho)$ is also a fault in $\mathcal{P}(\rho a)$.
The only faulty run of observed sequence a is $\rho = q_0 u q_2 \mathbf{f} f_1 a f_2$.
ρ is necessarily followed by a b.
So $\mathfrak{W}(q_0 u q_1 (a q_1)^2) =$ **false**.