Détection de fautes dans les systèmes probabilistes

Engel Lefaucheux encadré par Serge Haddad et Nathalie Bertrand
Inria Rennes et LSV

Vendredi 22 Août 2014

Le contexte général

En 1995, Sampath et al [13] introduisirent le diagnostic pour les systèmes à évènements discrets. Diagnostiquer un système signifie détecter lorsque ce dernier commet une « faute ». En 2000, ce problème a été montré comme étant soluble en temps polynomial [9]. Cette question fut par la suite étendue aux systèmes probabilistes par Thorsley et Teneketsis [14] qui proposèrent deux notions de diagnostiquabilité : la A-diagnostiquabilité qui requiert de détecter la faute avec probabilité 1 et la AA-diagnostiquabilité qui, pour une probabilité d'erreur arbitrairement petite, consiste à détecter la faute avec probabilité 1. Chen et Kumar [4] ont adapté en 2013 les algorithmes utilisés dans le cas non probabiliste à ces nouvelles notions de diagnostiquabilité obtenant ainsi une complexité polynomiale (comme nous l'avons établi, ces algorithmes sont malheureusement erronés). Afin d'anticiper les fautes, on s'intéresse également à la prédiction [6] qui consiste à détecter qu'une faute aura nécessairement lieu.

Le problème étudié

Le sujet central de ce stage est l'étude de plusieurs problèmes liés à l'observation partielle de systèmes probabilistes : le diagnostic tout d'abord, mais aussi la prédiction, la détection d'état, etc. Un des objectifs était donc la construction d'un diagnostiqueur, une fonction qui déduit des observations du système l'occurence d'une faute. Quatre questions fondamentales se posaient alors :

- Quelle spécification choisir pour la diagnostiquabilité ?
- Quelle est la complexité du problème de décision associé ?
- Quelle est la complexité du problème de synthèse d'un diagnostiqueur et quelle est la taille minimale d'un diagnostiqueur ?
- Comment combiner prédiction et diagnostic afin d'obtenir un diagnostic au plus tôt ?

La contribution proposée

Initialement, nous avons démontré que le problème de la A-diagnostiquabilité était PSPACE-difficile ce qui nous a conduit à exhiber des contre-exemples aux algorithmes de Chen et Kumar [4]. D'autre part nous avons constaté que différentes notions pertinentes de diagnostiquabilité avaient été proposées. Aussi avons défini deux critères discriminants, la réactivité et la correction et établi les relations exactes entre les différentes notions à la fois dans le cas des systèmes infinis et finis. Nous avons ensuite démontré que le problème de décision de la diagnostiquabilité est PSPACE-complet quelle que soit la définition retenue. Nous avons proposé des algorithmes de synthèse de diagnostiqueurs de taille optimale à une constante près.

Nous avons également travaillé sur la prédiction en nous demandant une fois de plus quelles notions de prédiction sont intéressantes et avons résolu les problèmes de décision et de synthèse

correspondants. Afin de tirer partie des avantages du diagnostic et de la prédiction, nous avons introduit et étudié la notion de prédiagnostiquabilité.

Les arguments en faveur de sa validité

Les preuves que nous avons développées ont l'avantage d'être adaptables à de nombreux cas. Ainsi la preuve de la PSPACE-difficulté de la diagnostiquabilité est très similaire pour les notions nécessitant une réponse sûre. De même nous avons adapté cette preuve avec peu de changements pour une notion connexe appelée A-détection.

Nous avons caractérisé la complexité exacte des problèmes de décision et de synthèse. De manière surprenante les hiérarchies de complexités relatives à ces problèmes sont différentes : (1) la prédictibilité est NLOGSPACE-complète alors que la diagnostiquabilité est PSPACE-complète mais (2) les tailles optimales des prédicteurs et des diagnostiqueurs sont du même ordre de grandeur $2^{\Theta(n)}$.

Le bilan et les perspectives

Nous avons atteint les deux objectifs principaux que nous nous étions fixés : clarifier les distinctions entre les différentes notions de diagnostic et caractériser la complexité des problèmes associés.

Nous pouvons distinguer deux types de perspectives, celles pour lesquelles nous avons déjà des résultats partiels et celles qui s'inscriraient plutôt dans le cadre d'une thèse.

Nous avons étudié comment nos résultats pouvaient s'étendre à d'autres questions proches comme la détection d'état. Comme l'algorithme de Chen et Kumar est erroné, il n'existe pas à l'heure actuelle d'algorithme de décision pour la AA-diagnostiquabilité. Nous avons établi que des problèmes proches étaient indécidables mais le statut de l'AA-diagnostiquabilité reste un problème ouvert. Nous avons également commencé des travaux relatifs à l'optimisation du diagnostic en donnant par exemple un coût à l'observation d'un évènement du système.

Le problème du diagnostic est issu de la communauté de l'automatique et du contrôle. Il serait intéressant de développer un prototype qui pourrait être un point de départ à une collaboration avec des chercheurs de cette communauté. Par ailleurs, nos algorithmes de décision et de synthèse s'appliquent uniquement aux systèmes finis. Que peut-on obtenir avec des systèmes infinis ? Quelles notions de diagnostiquabilité sont intéressantes dans ce cadre ? Un axe de recherche à entreprendre consisterait à étudier les systèmes infinis munis d'un attracteur fini [2].

Notes

La plupart des résultats de ce stage ont donné lieu à un rapport de recherche [1]. Ils font aussi l'objet d'une soumission à FSTTCS 2014. Durant ce stage j'ai présenté nos résultats à trois reprises : lors des journées annuelles du GT-Vérif, au séminaire de l'équipe SUMO d'Inria Rennes et au séminaire de l'équipe « Méthodes Formelles et Vérification » de l'Université Libre de Bruxelles.

Introduction

Le stage de deuxième année du Master MPRI permet d'approfondir nos compétences dans un domaine de l'informatique et potentiellement d'en découvrir un nouveau si nous ne sommes pas encore fixés sur nos domaines d'intérêt en recherche. Étant attiré par la théorie des jeux et la vérification dans des systèmes probabilistes, j'ai cherché un stage mêlant ces deux domaines. Après discussion avec Serge Haddad et Nathalie Bertrand, nous avons convenu d'étudier les jeux stochastiques à observation partielle et leurs applications au diagnostic de fautes. Ce stage a duré 4 mois et demi pour la première moitié dans l'équipe SUMO de l'INRIA Rennes et pour la seconde au LSV de l'ENS Cachan. J'ai eu l'occasion de présenter une partie du travail réalisé durant ce stage à trois reprises, notamment aux journées annuelles du GT-Vérif et au séminaire de l'équipe « Méthodes Formelles et Vérification » de l'Université Libre de Bruxelles.

En informatique, le terme diagnostic correspond à plusieurs notions différentes. Par exemple, en intelligence artificielle le diagnostic peut représenter l'identification d'une maladie à partir de ses symptômes, comme le fait le système expert MYCIN [3]. Dans ce qui suit, nous étudions le diagnostic de fautes tel qu'il est vu en théorie du contrôle où il est appliqué à des systèmes partiellement observables susceptibles de faire des fautes. Une séquence d'observation d'un tel système est dite surement correcte (resp. surement fautive) si toutes les exécutions possibles du système correspondant à cette séquence sont correctes (resp. fautives). Les autres séquences sont dites ambiguës. Le but d'un diagnostiqueur est de détecter à partir des observations si l'exécution courante est fautive. Le problème de l'existence d'un diagnostiqueur est le problème dit de diagnostiquabilité [13]. Afin d'anticiper les fautes, on s'intéresse aussi aux prédicteurs dont le but est d'annoncer si une faute arrivera certainement ; l'existence d'un prédicteur est le problème de prédictibilité [6].

La diagnostiquabilité et la prédictibilité ont d'abord été définies et étudiées pour des systèmes à évènements discrets modélisés par des systèmes à transitions étiquetées (labelled transition system en anglais, abrégé en LTS) pour lesquels ces problèmes sont solubles en PTIME [6] [9]. Les diagnostiqueurs/prédicteurs ont néanmoins une taille potentiellement exponentielle pour satisfaire les deux conditions qu'on leur impose, à savoir : (1) la correction, l'information fournie par le diagnostiqueur/prédicteur est avérée, et (2) la réactivité, assurant que toute faute sera détectée.

Une extension de la diagnostiquabilité des LTS a été proposée pour les chaînes de Markov à transitions étiquetées [14], aussi appelées systèmes de transitions probabilistes étiquetés (pLTS). Dans un contexte probabiliste, la condition de réactivité demande désormais que la faute soit détectée avec probabilité 1. Pour la condition de correction, deux spécifications ont été proposées : soit on garde la version originale demandant que toute information fournie soit correcte (A-diagnostiquabilité), soit on affaiblit cette condition en autorisant des erreurs dans l'information fournie, à condition que la probabilité d'erreur puisse être choisie arbitrairement petite (AA-diagnostiquabilité).

Contributions.

- Afin d'obtenir une classification sémantique robuste des notions de diagnostiquabilité, nous définissons des critères pour la diagnostiquabilité dans les systèmes probabilistes selon que (1) l'information produite par le diagnostiqueur soit relative aux exécutions fautives seulement ou à toutes les exécutions et (2) que l'ambiguité soit définie au niveau des exécutions infinies ou pour des préfixes finis de plus en plus longs. Ces deux dimensions décrivent quatre spécifications de la diagnostiquabilité. Nous présentons ces notions et étudions leurs liens dans la section 1.

– Pour des systèmes probabilistes finis nous établissons dans la section 2 une caractérisation de ces notions basée sur un automate déterministe (fini ou de Büchi) servant d'observateur et synchronisé avec le pLTS. Grâce à ces caractérisations nous montrons que le problème de la diagnostiquabilité pour ces spécifications est PSPACE-complet. Nous produisons également des algorithmes de synthèse de diagnostiqueur et prouvons que la taille de ces diagnostiqueurs, $2^{\Theta(n)}$ (où n est le nombre d'états du pLTS), est optimale.
– Comme la prédiction est une alternative intéressante au diagnostic, nous proposons dans la section 3, deux spécifications possibles pour la prédictibilité dans les systèmes probabilistes. Nous montrons que dans les deux cas le problème de la prédictibilité est NLOGSPACE-complet. Cependant, comme pour les diagnostiqueurs, la taille optimale des prédicteurs est en $2^{\Theta(n)}$.
– Dans cette même section nous présentons et étudions le prédiagnostic, une notion combinant les avantages de la prédiction et du diagnostic. Pour les deux notions de prédiagnostiquabilité que nous introduisons, nous prouvons que le problème de la prédiagnostiquabilité est PSPACE-complet et nous construisons des prédiagnostiqueurs de taille optimale. La hiérarchie entre les différentes notions est résumée dans une figure à la fin de cette section.
– Des travaux en cours sur la AA-diagnostiquabilité et l'optimisation du nombre d'observations nécessaires pour obtenir une information sont présentés dans la section 4.

Dans les annexes se trouvent les preuves les plus importantes. Toutes les preuves des trois premières parties et quelques résultats supplémentaires sont accessibles dans le rapport de recherche [1]. En annexe également, nous montrons que les algorithmes de [4] pour l'A-diagnostiquabilité et l'AA-diagnostiquabilité sont erronés.

1 Spécification du diagnostic

1.1 Système de transition probabiliste étiqueté

Pour l'étude de la détection de faute dans des systèmes à événements discrets, on utilise généralement des systèmes de transitions étiquetés pour modéliser le système étudié. Dans un contexte probabiliste, ces systèmes s'apparentent à des chaînes de Markov étiquetées.

Définition 1. Un *système de transition probabiliste étiqueté* (pLTS) est un tuple $\mathcal{A} = \langle Q, q_0, \Sigma, T, \mathbf{P}\rangle$ où :
– Q est un ensemble d'états, $q_0 \in Q$ étant l'état initial ;
– Σ est un ensemble fini d'actions ;
– $T \subseteq Q \times \Sigma \times Q$ est un ensemble de transitions ;
– $\mathbf{P}$ est une fonction de T vers $\mathbb{Q}_{>0}$ satisfaisant pour tout $q \in Q$: $\sum_{(q,a,q')\in T} \mathbf{P}[q,a,q'] = 1$.

On définit la relation de transition du pLTS par $q \xrightarrow{a} q'$ quand $(q,a,q') \in T$; cette transition est alors dite *franchissable* en q. Un pLTS est vivant si dans tout état du pLTS, au moins une transition est franchissable. Nous supposons par la suite qu'en tout état du pLTS, seul un nombre dénombrable de transitions sont franchissables. Cela permet de s'assurer que la somme $\sum_{(q,a,q')\in T} \mathbf{P}[q,a,q']$ est bien définie.

Introduisons maintenant quelques notions et notations qui seront utilisées tout au long du rapport. Une *exécution* ρ d'un pLTS est un mot (fini ou infini) $\rho = q_0a_0q_1\ldots$ tel que pour tout i, $q_i \in Q$, $a_i \in \Sigma$ et quand q_{i+1} est défini, $q_i \xrightarrow{a_i} q_{i+1}$. La notion d'exécution peut être généralisée en partant depuis un état arbitraire q du pLTS. Nous appelons Ω l'ensemble des exécutions infinies de $\mathcal{A}$ commençant en q_0, en supposant que le pLTS étudié est clair de par le contexte. Si elle est finie, l'exécution s'arrête en un état q et sa *longueur*, noté $|\rho|$, est le nombre d'actions présentes dans l'exécution. Etant données une exécution finie $\rho = q_0a_0q_1\ldots q_n$ et une exécution $\rho' = q_na_nq_{n+1}\ldots$, la concaténation de ρ et ρ', $\rho\rho'$, est l'exécution $q_0a_0q_1\ldots q_na_nq_{n+1}\ldots$; ρ est alors un préfixe de

$\rho\rho'$, ce que l'on écrit $\rho \preceq \rho\rho'$. La séquence associée à une exécution $\rho = qa_0q_1\ldots$ est le mot $\sigma_\rho = a_0a_1\ldots$, et nous notons indifféremment $q \overset{\rho}{\Rightarrow}$ ou $q \overset{\sigma_\rho}{\Longrightarrow}$ (resp. $q \overset{\rho}{\Rightarrow} q'$ ou $q \overset{\sigma_\rho}{\Longrightarrow} q'$) pour une exécution infinie (resp. finie) ρ. Un état q est *accessible* (depuis q_0) s'il existe une exécution ρ telle que $q_0 \overset{\rho}{\Rightarrow} q$, ce que l'on pourra noter $q_0 \Rightarrow q$. Le langage d'un pLTS consiste en l'ensemble des mots infinis qui étiquettent des exécutions du pLTS : $\mathcal{L}^\omega(\mathcal{A}) = \{\, \sigma \in \Sigma^\omega \mid q_0 \overset{\sigma}{\Rightarrow} \}$.

Une mesure de probabilité peut être définie sur l'ensemble des exécutions infinies de la même façon que pour des chaînes de Markov à temps discret :

$$\mathbb{P}(C(q_0a_0q_1\ldots q_n)) = \mathbf{P}[q_0, a_1, q_1]\cdots\mathbf{P}[q_{n-1}, a_{n-1}, q_n] \ .$$

où $C(\rho) = \{\rho' \in \Omega \mid \rho \preceq \rho'\}$.

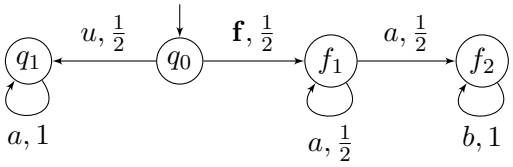

FIGURE 1 – Un exemple de pLTS.

Exemple. La figure 1 représente un pLTS. Pour $n \in \mathbb{N}$, $\rho = q_0uq_1(aq_1)^n$ et $\rho' = q_0\mathbf{f}f_1(af_1)^{n-2}af_2bf_2$ sont deux exécutions de longueur $n+1$. Par ailleurs la probabilité de ces deux exécutions, c'est à dire la mesure de probabilité de l'ensemble des séquences infinies ayant ρ (resp. ρ') pour préfixe est $\mathbb{P}(C(\rho)) = \frac{1}{2}$ et $\mathbb{P}(C(\rho')) = \frac{1}{2^n}$.

1.2 Observation partielle et ambiguïté

Afin de formaliser les problèmes liés à la détection de faute, nous séparons Σ en deux ensembles disjoints Σ_o et Σ_u, l'ensemble des actions *observables* et celui des actions *inobservables*. Nous nous intéressons en outre à une action particulière, la *faute*, $\mathbf{f} \in \Sigma_u$.
Soit σ un mot fini, sa longueur est notée $|\sigma|$. La projection de σ sur Σ_o est définie inductivement par $\mathcal{P}(\varepsilon) = \varepsilon$, pour $a \in \Sigma_o$, $\mathcal{P}(\sigma a) = \mathcal{P}(\sigma)a$; et pour $a \notin \Sigma_o$ $\mathcal{P}(\sigma a) = \mathcal{P}(\sigma)$. On note $|\sigma|_o$ pour $|\mathcal{P}(\sigma)|$. Quand σ est un mot infini, sa projection est la limite des projections de ses préfixes finis. Cette projection est applicable aux exécutions par le biais de leur séquence associée. Cette projection est aussi étendue au langage de façon classique.
Un pLTS est *convergent* vis à vis d'une partition $\Sigma = \Sigma_o \uplus \Sigma_u$ s'il ne peut y avoir une séquence infinie d'actions inobservables depuis un état accessible : $\mathcal{L}^\omega(\mathcal{A}) \cap \Sigma^*\Sigma_u^\omega = \emptyset$. Si $\mathcal{A}$ est convergent, pour tout $\sigma \in \mathcal{L}^\omega(\mathcal{A})$, $\mathcal{P}(\sigma) \in \Sigma_o^\omega$. Dans la suite, nous supposons les pLTS convergents. *Séquence* désignera un mot (fini ou infini) sur Σ et *séquence d'observation* un mot (fini ou infini) sur Σ_o.

La *longueur observable* d'une exécution ρ, $|\rho|_o \in \mathbb{N} \cup \{\infty\}$, est le nombre d'actions observables ayant lieu au cours de l'exécution. Une exécution *fermée* est une exécution finie terminant avec une action observable. Les exécutions fermées sont celles qui sont pertinentes dans le cadre de l'observation partielle car seules les actions observables apportent une information à un observateur externe. SR (exécution fermée étant signalling run en anglais) désignera par la suite l'ensemble des exécutions fermées et SR_n désignera l'ensemble des exécutions fermées de longueur observable n. Comme les pLTS considérés sont convergents, pour tout $n > 0$, SR_n est doté d'une distribution de probabilité définie en associant la valeur $\mathbb{P}(\rho) = \mathbb{P}(C(\rho))$ à tout $\rho \in \mathsf{SR}_n$. Pour une exécution ρ et $n \leq |\rho|_o$, $\rho_{\downarrow n}$ représente l'exécution fermée préfixe de ρ et de longueur observable n.

Soit $\mathcal{A}$ un pLTS. Une exécution ρ est *fautive* si σ_ρ contient $\mathbf{f}$, sinon elle est *correcte*. Pour simplifier les preuves et sans perdre de généralité, nous supposons que les états sont partitionnés

entre états corrects et états fautifs : $Q = Q_f \uplus Q_c$ où Q_f sont les états fautifs, et Q_c sont les états corrects. Un état fautif (resp. correct) ne peut être atteint que par des exécutions fautives (resp. correctes). Une séquence d'observation $\sigma \in \Sigma_o^\omega$ est *surement correcte* si $\mathcal{P}^{-1}(\sigma) \cap \mathcal{L}^\omega(\mathcal{A}) \subseteq (\Sigma \backslash \mathbf{f})^\omega$; elle est *surement fautive* si $\mathcal{P}^{-1}(\sigma) \cap \mathcal{L}^\omega(\mathcal{A}) \subseteq \Sigma^* \mathbf{f} \Sigma^\omega$; sinon, elle est *ambigüe*. Pour les séquences finies, on considère uniquement les exécutions fermées : une séquence d'observation finie $\sigma \in \Sigma_o^*$ est *surement fautive* (resp. *surement correcte*) si pour toute exécution fermée ρ telle que $\mathcal{P}(\sigma_\rho) = \sigma$, ρ est fautive (resp. correcte) ; sinon elle est ambigüe. Une exécution ρ est surement correcte/surement fautive/ambigüe si $\mathcal{P}(\rho)$ l'est.

Pour définir différentes formes de détection de faute et étudier leurs liens, nous nous intéressons aux ensembles d'exécutions infinies suivants.

Définition 2. Soit $\mathcal{A}$ un pLTS et $n > 0$. Alors :
- FAmb_∞ est l'ensemble des exécutions fautives infinies de $\mathcal{A}$;
- CAmb_∞ est l'ensemble des exécutions correctes infinies de $\mathcal{A}$;
- FAmb_n est l'ensemble des exécutions infinies ρ de $\mathcal{A}$ dont l'exécution préfixe fermée de longueur observable n, $\rho_{\downarrow n}$, est fautive et ambigüe ;
- CAmb_n est l'ensemble des exécutions infinies ρ de $\mathcal{A}$ dont l'exécution préfixe fermée de longueur observable n, $\rho_{\downarrow n}$, est correcte et ambigüe.

Exemple. Dans le pLTS de la figure 2 et toutes les figures à venir, les transitions étiquetées par les actions inobservables sont représentées avec des pointillés. Par ailleurs, sauf mention du contraire, les transitions issues d'un état sont uniformément distribuées. Pour $n \in \mathbb{N}$, $\mathsf{FAmb}_n = C(q_0\mathbf{f}(f_1a)^n f_1) \cup C(q_0\mathbf{f}(f_1a)^n f_2)$, donc $\mathbb{P}(\mathsf{FAmb}_n) = \frac{1}{2^n}$. On a aussi $\mathbb{P}(\mathsf{CAmb}_\infty) = \mathbb{P}(\{q_0u(q_1a)^\omega\}) = \frac{1}{2}$.

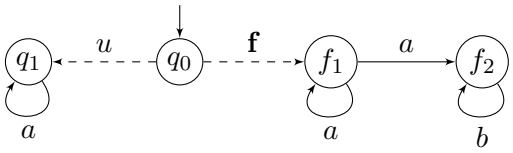

FIGURE 2 – Un pLTS IF-diagnostiquable qui n'est pas IA-diagnostiquable.

1.3 Quelles formes de diagnostic ?

Nous avons étudié quatre formes de diagnostic des systèmes probabilistes. Deux critères distinguent ces différentes spécifications : désire-t-on que l'absence d'ambiguïté ait lieu pour toutes les exécutions ou uniquement pour les fautives, et ladite ambiguïté doit-elle être étudiée pour des exécutions infinies ou pour des préfixes de plus en plus longs ?

Définition 3. Soit $\mathcal{A}$ un pLTS. Alors :
- $\mathcal{A}$ est IF-diagnostiquable si $\mathbb{P}(\mathsf{FAmb}_\infty) = 0$.
- $\mathcal{A}$ est IA-diagnostiquable si $\mathbb{P}(\mathsf{FAmb}_\infty \uplus \mathsf{CAmb}_\infty) = 0$.
- $\mathcal{A}$ est FF-diagnostiquable si $\limsup_{n\to\infty} \mathbb{P}(\mathsf{FAmb}_n) = 0$.
- $\mathcal{A}$ est FA-diagnostiquable si $\limsup_{n\to\infty} \mathbb{P}(\mathsf{FAmb}_n \uplus \mathsf{CAmb}_n) = 0$.

Le théorème suivant résume les liens existant entre les différentes spécifications du diagnostic.

Théorème 1. *Dans le tableau qui suit, les implications sont vraies pour des pLTS finis ou infinies, les non implications sont déjà vraies pour des pLTS finis. L'implication marquée d'une * requiert que le pLTS soit finiment branchant.*

Diagnostic	*Toutes les exécutions*		*Exécutions fautives*
Exécutions fermées	FA	$\Rightarrow$ $\not\Leftarrow$	FF
	$\Downarrow\not\Uparrow$		$\Downarrow\Uparrow^*$
Exécutions infinies	IA	$\Rightarrow$ $\not\Leftarrow$	IF

Exemple. Le pLTS de la figure 2 est IF-diagnostiquable sans être IA-diagnostiquable comme le montrent les calculs faits dans le paragraphe l'accompagnant.

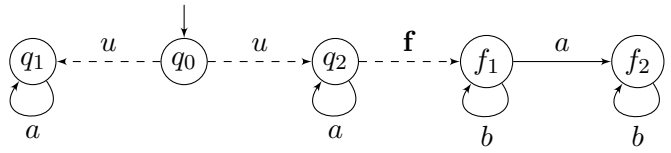

FIGURE 3 – Un pLTS IA-diagnostiquable sans être FA-diagnostiquable.

Exemple. Observons le pLTS de la figure 3 où $\Sigma_u = \{u, \mathbf{f}\}$. Une exécution infinie et fautive contient l'action b. Comme aucune exécution correcte ne peut produire de b, $\mathsf{FAmb}_\infty = \emptyset$. Il existe deux exécutions infinies correctes, toutes deux correspondant à la séquence d'observation a^ω. Elles ne peuvent donc être confondues avec une exécution fautive, donc $\mathsf{CAmb}_\infty = \emptyset$. Par conséquent ce pLTS est IA-diagnostiquable. L'exécution correcte et infinie $\rho = q_0u(q_1a)^\omega$ a probabilité $\frac{1}{2}$. Par ailleurs tous ses préfixes de longueur non nulle sont ambigus. En effet leurs séquences d'observation est a^n, pour un $n \in \mathbb{N}$ et $q_0u(q_2a)^{n-1}q_2\mathbf{f}f_1af_2$ est une exécution fautive ayant la même séquence d'observation. Par conséquent pour tout $n \geq 1$, $\mathbb{P}(\mathsf{CAmb}_n) \geq \frac{1}{2}$, donc ce pLTS n'est pas FA-diagnostiquable.

Ayant pour but le développement d'algorithmes, nous nous concentrons par la suite sur les pLTS finis. Par conséquent les notions de FF-diagnostiquabilité et IF-diagnostiquabilité sont confondues. De plus, la notion de IF-diagnostiquabilité coïncide alors avec celle de A-diagnostiquabilité introduite en [14].

Théorème 2. *Un pLTS fini $\mathcal{A}$ est* IF*-diagnostiquable si et seulement si il est A-diagnostiquable, c'est-à-dire : $\forall \varepsilon > 0$, $\exists N_\varepsilon \in \mathbb{N}$, pour toute exécution fermée fautive ρ et tout $n \geq N_\varepsilon$, $\mathbb{P}(\{\rho' \in \mathsf{FAmb}_{n+|\rho|_o} \mid \rho \preceq \rho'\}) < \varepsilon\mathbb{P}(\rho)$. Cette condition n'est que suffisante pour les pLTS finiment branchants.*

Notre approche du diagnostic soulève trois problèmes.
- Un problème de décision nommé le problème de la diagnostiquabilité : étant donné un pLTS et une spécification donnée, ce pLTS est-il diagnostiquable pour cette spécification ?
- Un problème sémantique : pour une spécification donnée de diagnostiquabilité, existe-t-il une spécification de diagnostiqueur telle que l'existence d'un diagnostiqueur soit équivalente à la diagnostiquabilité ?
- Un problème de synthèse : étant donné un pLTS diagnostiquable pour une spécification donnée, comment construire un diagnostiqueur et quelle est sa taille optimale ?

2 Étude de la complexité du diagnostic

Nous nous intéressons dans cette section aux trois problèmes évoqués précédemment. Les méthodes employées sont similaires pour chaque notion de diagnostiquabilité, par conséquent nous n'étudierons ici que le cas le plus compliqué : la IA-diagnostiquabilité.

2.1 Caractérisations

Afin de produire des algorithmes, nous commençons par fournir une caractérisation de la IA-diagnostiquabilité. Pour cela, étant donné un pLTS $\mathcal{A}$, on construit un automate déterministe qui accepte certaines séquences d'observation de $\mathcal{A}$. Puis on réalise le produit synchronisé de cet automate avec $\mathcal{A}$, ce qui nous donne un pLTS ayant le même comportement probabiliste que $\mathcal{A}$ mais auquel une information à propos de l'exécution actuelle (donnée par l'automate) a été ajoutée. Notre caractérisation de la diagnostiquabilité repose sur des propriétés de graphe de ce produit synchronisé.

Le IA-automate est l'automate déterministe de Büchi présenté dans [8]. Il maintient trois ensembles disjoints d'états ; ses états sont de la forme (U, V, W). U est l'ensemble des états corrects accessibles par une exécution fermée correspondant à la séquence d'observation courante. $V \cup W$ est l'ensemble des états fautifs accessibles. La séparation en deux ensembles reflète le fait que l'IA-automate essaie de résoudre l'ambiguïté entre les exécutions menant aux états de U et de W (quand ces deux ensembles sont non vides), tandis que V est une salle d'attente pour des exécutions potentiellement fautives qui seront examinées lorsque l'ambiguïté actuelle sera résolue. Formellement, $\mathsf{IA}(\mathcal{A})$ est défini par :

- L'ensemble des états est $Q = \{(U, V, W) | U \subseteq Q_c, V \subseteq Q_f, W \subseteq Q_f, V \cap W = \emptyset\}$.
- $s_0 = (\{q_0\}, \emptyset, \emptyset)$ est l'état initial d'$\mathsf{IA}(\mathcal{A})$;
- Pour (U, V, W) un état d'$\mathsf{IA}(\mathcal{A})$ et $a \in \Sigma_o$, il y a une transition $(U, V, W) \xrightarrow{a} (U', V', W')$ si :
  1. $E = \{\rho = q_{\alpha_0} a_1 \dots a_k q_{\alpha_k} \mid \rho \text{ est une exécution de } \mathcal{A}, q_{\alpha_0} \in U \cup V \cup W, \forall i < k \ a_i \in \Sigma_u, a_k = a\} \neq \emptyset$
  2. $U' = \{q \mid \exists \rho = q_{\alpha_0} a_1 \dots a_k q_{\alpha_k} \text{ exécution correcte de } E, q_{\alpha_0} \in U, q_{\alpha_k} = q\}$,
  3. Si $W = \emptyset$ alors $V' = \emptyset$ et
     $W' = \{q \mid \exists \rho = q_{\alpha_0} a_1 \dots a_k q_{\alpha_k} \text{ exécution de } E, q_{\alpha_0} \in V, q_{\alpha_k} = q\} \cup$
     $\{q \mid \exists \rho = q_{\alpha_0} a_1 \dots a_k q_{\alpha_k} \text{ exécution fautive de } E, q_{\alpha_0} \in U, q_{\alpha_k} = q\}$,
  4. Si $W \neq \emptyset$ alors $W' = \{q \mid \exists \rho = q_{\alpha_0} a_1 \dots a_k q_{\alpha_k} \text{ exécution de } E, q_{\alpha_0} \in W, q_{\alpha_k} = q\}$ et
     $V' = \Big(\{q \mid \exists \rho = q_{\alpha_0} a_1 \dots a_k q_{\alpha_k} \text{ exécution de } E, q_{\alpha_0} \in V, q_{\alpha_k} = q\} \cup$
     $\{q \mid \exists \rho = q_{\alpha_0} a_1 \dots a_k q_{\alpha_k} \text{ exécution fautive de } E, q_{\alpha_0} \in U, q_{\alpha_k} = q\}\Big) \setminus W'$.
- L'ensemble des états acceptants F est constitué de tous les triplets (U, V, W) tels que $U = \emptyset$ ou $W = \emptyset$.

Quand $U = \emptyset$, l'exécution actuelle est surement fautive. Quand $W = \emptyset$ l'exécution fermée actuelle peut être ambigüe (si V est non vide) mais les exécutions fautives possibles les plus « anciennes » ont pu être éliminées. Par conséquent, une séquence d'observation infinie de $\mathcal{A}$ visitant infiniment souvent F n'est pas ambigüe (toute ambiguïté sera résolue).

Exemple. La figure 4 représente le IA-automate du pLTS de la figure 3. La séquence d'observation a^ω alterne entre les états s_1 et s'_1 qui appartient à F. Elle est donc acceptée et non ambigüe alors que toutes les séquences d'observation a^n pour $n \in \mathbb{N}$ sont ambigües.

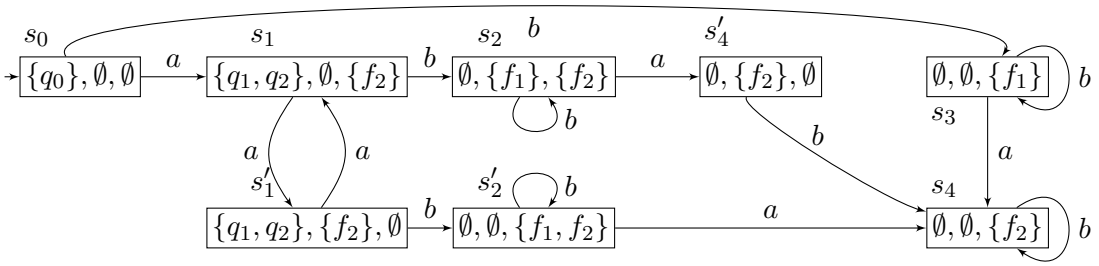

FIGURE 4 – Le IA-automate du pLTS de la figure 3.

La proposition suivante rappelle la propriété de cet automate.

Proposition 3 ([8]). *Soit $\mathcal{A}$ un pLTS fini. Alors l'automate déterministe de Büchi $\mathsf{IA}(\mathcal{A})$ accepte les séquences d'observation infinies non ambigües de $\mathcal{A}$.*

Afin d'obtenir une caractérisation, on construit $\mathcal{A}_{\mathsf{IA}} = \mathcal{A} \times \mathsf{IA}(\mathcal{A})$, le produit de $\mathcal{A}$ et $\mathsf{IA}(\mathcal{A})$ synchronisé par les actions observables.

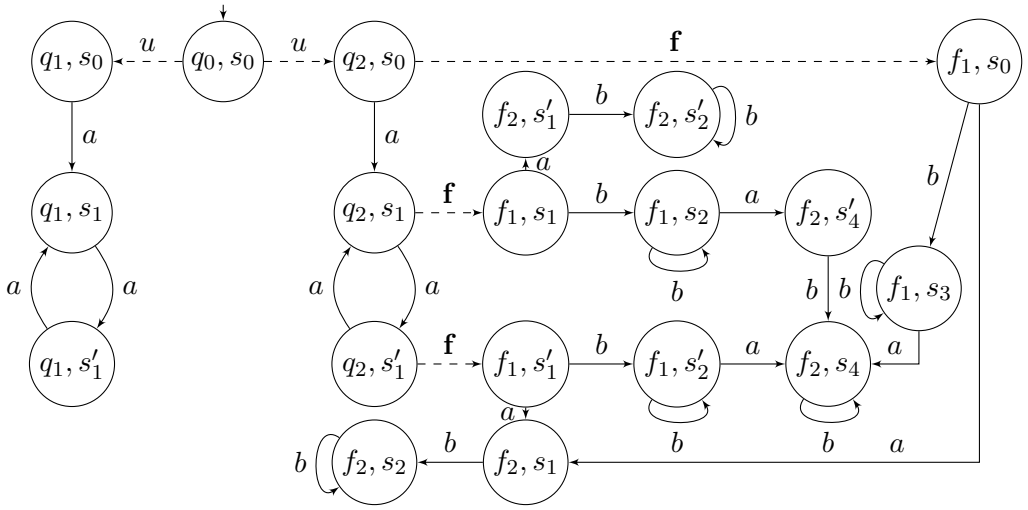

FIGURE 5 – Le produit synchronisé du pLTS de la figure 3 et de son IA-automate.

La proposition suivante établit l'intérêt de cette construction.

Proposition 4. *Soit $\mathcal{A}$ un pLTS fini. $\mathcal{A}$ est IA-diagnostiquable si et seulement si $\mathcal{A}_{\mathsf{IA}}$ ne contient pas de composantes fortement connexes terminales (CFCT) telle que :*
- *soit, tous ses états (q, U, V, W) vérifient $q \in Q_f$ et $U \neq \emptyset$;*
- *soit tous ses états (q, U, V, W) vérifient $q \in Q_c$ et $W \neq \emptyset$.*

Exemple. La figure 5 montre le produit synchronisé correspondant au pLTS de la figure 3. Parmi les CFCT du produit synchronisé, les états des CFCT *fautives* (i.e. accessibles par une faute) vérifient $U = \emptyset$ tandis que $\{(q_1, s_1), (q_1, s'_1)\}$, la seule CFCT *correcte* contient un état (q_1, s'_1) avec $W = \emptyset$. Ceci montre que ce pLTS est IA-diagnostiquable.

2.2 Complexité du problème de décision

Nous avons établi que le problème de la diagnostiquabilité est PSPACE-complet pour toutes les variantes du diagnostic étudiées.

Théorème 5. *Les problèmes de la IA-diagnostiquabilité, de la IF-diagnostiquabilité et de la FF-diagnostiquabilité sont PSPACE-complets.*

Détaillons le cas de la IA-diagnostiquabilité. L'algorithme de décision pour la IA-diagnostiquabilité vérifie si la caractérisation présentée en proposition 4 est satisfaite en cherchant un état violant l'une des contraintes et en vérifiant que cet état appartient à une CFCT. Ceci peut être réalisé en espace polynomial sans construire explicitement $\mathcal{A}_{\mathsf{IA}}$, et en s'appuyant sur le théorème de Savitch.

Afin d'établir une borne inférieure pour la complexité de la IA-diagnostiquabilité, nous introduisons une variante du problème de l'universalité d'un langage. Un langage $\mathcal{L}$ sur un alphabet

Σ est dit *éventuellement universel* s'il existe un mot $v \in \Sigma^*$ tel que $v^{-1}\mathcal{L} = \Sigma^*$. Récemment, plusieurs variantes du problème de l'universalité pour les automates finis non-déterministes (NFA) ont été prouvées PSPACE-complètes [12] mais, pour autant que nous le sachions, la question de l'éventuelle universalité n'a pas été considéré. Motivés par nos objectifs, nous nous intéressons à des NFA vivants pour lesquels tous les états sont terminaux. Comme pour les pLTS, un NFA est *vivant* si dans chaque état il y a au moins une transition franchissable. Le langage d'un NFA $\mathcal{A}$, $\mathcal{L}(\mathcal{A})$, est l'ensemble des mots finis qui sont acceptés par $\mathcal{A}$.

Proposition 6. *Soit $\mathcal{A}$ un NFA vivant où tous les états sont terminaux. Décider si $\mathcal{L}(\mathcal{A})$ est éventuellement universel est* PSPACE*-difficile.*

Esquissons maintenant comment réduire ce problème à la IA-diagnosabilité. Étant donné un NFA vivant $\mathcal{A}$ sur Σ où tous les états sont terminaux, on construit le pLTS de la figure 6 où $\Sigma \cup \{\sharp\}$ est l'ensemble des actions observables. Comme une exécution correcte « émet » presque surement un $\sharp$, une ambiguïté ne peut avoir lieu que sur une exécution fautive. Or après une faute on observe Σ^*. D'après la caractérisation de la proposition 4, on peut conclure que le pLTS n'est pas IA-diagnostiquable si et seulement si $\mathcal{A}$ est éventuellement universel.

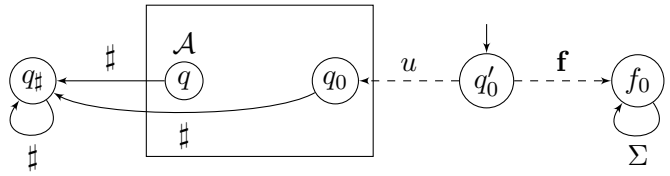

FIGURE 6 – Une réduction pour la PSPACE-difficulté de la IA-diagnostiquabilité.

Puisque l'IF-diagnostiquabilité coïncide avec la A-diagnostiquabilité, ceci contredit le résultat de [4] où une procédure de décision PTIME est proposée. Dans l'annexe B, nous décrivons cet algorithme, donnons un exemple de pLTS sur lequel il n'est pas correct et expliquons où se situe l'erreur dans la preuve.

Fait 7. *La procédure de décision de [4] pour la A-diagnostiquabilité est erronée.*

2.3 Construction des diagnostiqueurs

Dans cette section nous nous intéressons aux diagnostiqueurs : une fonction $D : \Sigma_o^* \to \{?, \top, \bot\}$ qui analyse la séquence d'observation pour indiquer si l'exécution est fautive ou correcte. De façon informelle, quand un diagnostiqueur retourne ?, il ne fournit aucune information, alors que lorsqu'il indique $\top$ une faute est certaine et $\bot$ indique qu'une information à propos de la correction a été fournie (information dépendant du type de diagnostiqueur). Ces valeurs sont partiellement ordonnées par $\prec$ où $? \prec \bot$ et $? \prec \top$.

Un diagnostiqueur à mémoire finie est défini par un tuple $(M, \Sigma, m_0, \mathsf{up}, D_{fm})$ où M est un ensemble fini d'états de mémoire, $m_0 \in M$ est l'état de mémoire initial, $\mathsf{up} : M \times \Sigma_o \to M$ est la fonction de mise à jour de la mémoire, et $D_{fm} : M \to \{?, \top, \bot\}$ est la fonction de diagnostic. up est étendue en une fonction $\mathsf{up} : M \times \Sigma_o^* \to M$ définie inductivement par $\mathsf{up}(m, \varepsilon) = m$ et $\mathsf{up}(m, wa) = \mathsf{up}(\mathsf{up}(m, w), a)$. Un tel diagnostiqueur à mémoire finie n'est pas directement un diagnostiqueur comme défini précédemment, néanmoins il induit un diagnostiqueur défini par $D(w) = D_{fm}(\mathsf{up}(m_0, w))$.

Les diagnostiqueurs qui nous intéressent ont deux propriétés importantes : la correction et la réactivité. La correction assure que l'information produite par le diagnostiqueur est exacte et la réactivité indique quelles informations le diagnostiqueur se doit de produire. La définition précise de ces deux propriétés dépend de la notion de diagnostiquabilité que l'on veut étudier. Il y a donc

des IA-diagnostiqueurs, des FA-diagnostiqueurs et des FF-diagnostiqueurs. Par ailleurs, nous ne considérerons que des diagnostiqueurs, qui une fois qu'ils ont annoncé une faute en émettant $\top$ ne changent plus leur verdict. Tout diagnostiqueur correct peut être transformé en un diagnostiqueur satisfaisant cette propriété d'engagement.

Définissons à présent les diagnostiqueurs correspondant à chaque spécification. $\mathcal{A}$ désignera un pLTS fini. Nous commençons par l'IF-diagnostiqueur réalisant le diagnostic sur des pLTS IF-diagnostiquables.

Définition 4. Un IF-*diagnostiqueur* pour $\mathcal{A}$ est une fonction $D : \Sigma_o^* \to \{\top, ?\}$ telle que :

correction Pour tout $w \in \Sigma_o^*$, si $D(w) = \top$, alors w est surement fautive.
réactivité Pour toute exécution fautive finie ρ, $\mathbb{P}(\{\rho' \in \Omega \mid \rho \preceq \rho' \wedge D(\mathcal{P}(\rho')) =?\}) = 0$ où pour $w \in \Sigma_o^\omega$, $D(w) = \lim_{n\to\infty} D(w_{\leq n})$.

Notons que dans la définition ci-dessus la limite est bien définie car nous avons supposé que le diagnostiqueur s'engage lorsqu'il émet $\top$.

Exemple. Une exécution fautive du pLTS de la figure 2 émet presque surement un b. On peut donc définir le IF-diagnostiqueur D par $D(w) = \top$ pour tout $w \in \Sigma_o^* b \Sigma_o^*$ et $D(w) =?$ sinon.

Les IF-diagnostiqueurs ne se souciant que des exécutions fautives, nous n'avions pas besoin de $\bot$. Les FA-diagnostiqueurs et les IA-diagnostiqueurs considèrent également le diagnostic des exécutions correctes.

Définition 5. Un FA-diagnostiqueur pour $\mathcal{A}$ est une fonction $D : \Sigma_o^* \to \{\top, \bot, ?\}$ telle que :

correction Pour tout $w \in \Sigma_o^*$
- si $D(w) = \top$, alors w est surement fautive ;
- si $D(w) = \bot$, alors w est surement correcte.

réactivité $\mathbb{P}(\{\rho \in \Omega \mid D_{\inf}(\mathcal{P}(\rho)) =?\}) = 0$ où pour $w \in \Sigma_o^\omega$, $D_{\inf}(w) = \liminf_{n\to\infty} D(w_{\leq n})$.

Exemple. Observons le pLTS de la figure 7, toute séquence $w \in \{a, b\}^*$ est ambiguë. Par conséquent, pour un tel w, $D(w) =?$. A l'inverse une séquence w contenant une occurrence de c est surement fautive. Aussi pour un tel w, $D(w) = \top$. Ce diagnostiqueur est réactif car presque surement une séquence infinie contient une occurrence de c. Observons cependant qu'il ne fournit aucune information sur la correction de la séquence.

Les IA-diagnostiqueurs diffèrent des FA-diagnostiqueurs dans leur condition de correction. Intuitivement, les IA-diagnostiqueurs peuvent résoudre une ancienne ambiguïté alors qu'une nouvelle ambiguïté s'est déjà formée, contrairement aux FA-diagnostiqueurs.

Définition 6. Un IA-*diagnostiqueur* pour $\mathcal{A}$ est une fonction $D : \Sigma_o^* \to \{\top, \bot, ?\}$ telle que

correction pour tout $w \in \Sigma_o^*$
- si $D(w) = \top$, alors w est surement fautif ;
- si $D(w) = \bot$, en notant $|D(w)|_\bot = |\{0 < n \leq |w| \mid D(w_{\leq n}) = \bot\}|$, alors pour toute exécution fermée ρ telle que $\mathcal{P}(\rho) = w$, l'exécution $\rho_{\downarrow |D(w)|_\bot}$ est correcte.

réactivité $\mathbb{P}(\{\rho \in \Omega \mid D_{\sup}(\mathcal{P}(\rho)) =?\}) = 0$ où pour $w \in \Sigma_o^\omega$, $D_{\sup}(w) = \limsup_{n\to\infty} D(w_{\leq n})$.

L'information fournie par les $\bot$ est que le préfixe de longueur observable $|D(w)|_\bot \leq |w|$ d'une exécution fermée de séquence d'observation w est correct. Le diagnostiqueur peut déduire cette information des dernières $|w| - |D(w)|_\bot$ observations.

Exemple. Dans le pLTS de la figure 7, après avoir observé une séquence waa, avec $w \in \{a, b\}^*$, le diagnostiqueur sait a posteriori que deux actions plus tôt, c'est-à-dire après w, l'éxecution était nécessairement correcte. En effet, on ne peut observer aa après une faute. On peut donc définir un IA-diagnostiqueur D par : pour $w \in \{a, b\}^*(ab \cup aa)$, $D(w) = \bot$, pour $w \in \{a, b, c\}^* c$,

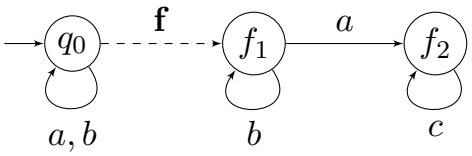

FIGURE 7 – Un pLTS qui admet un IA-diagnostiqueur.

$D(w) = \top$ et sinon $D(w) =?$. Sur cet exemple, le IA-diagnostiqueur apporte significativement plus d'informations que le FA-diagnostiqueur.

La proposition suivante répond au problème sémantique que nous avons soulevé et montre que les spécifications d'un IF/FA/IA-diagnostiqueur sont correctes vis à vis de la IF/FA/IA-diagnostiquabilité.

Proposition 8. *Un pLTS fini $\mathcal{A}$ est* IF/ FA/ IA*-diagnostiquable si et seulement si il admet un* IF/ FA/ IA*-diagnostiqueur.*

Lorsqu'un pLTS $\mathcal{A}$ est IA-diagnostiquable, on peut construire un IA-diagnostiqueur à partir du IA-automate : les états et transitions du diagnostiqueur à mémoire finie sont ceux de l'automate et la fonction de diagnostic D_{fm} renvoie $\top$ si $U = \emptyset$, $\bot$ si $W = \emptyset$ et ? sinon. On obtient donc une borne supérieure à la taille des IA-diagnostiqueurs et de la même façon une borne pour les IF-diagnostiqueurs et les FA-diagnostiqueurs.

Proposition 9. *Pour tout pLTS* IF/ FA/ IA*-diagnostiquable $\mathcal{A}$ avec n_c états corrects et n_f états fautifs, on peut construire un* IF/ FA/ IA*-diagnostiqueur avec au plus 2^{n_c}/ $2^{n_c+n_f}$/ $2^{n_c}3^{n_f}$ états.*

La famille de pLTS représentée par la figure 8 nécessite que son IA-diagnostiqueur ait une taille exponentielle. Intuitivement, chaque IA-diagnostiqueur de ce pLTS doit déclarer après avoir observé un c si l'exécution est fautive ou correcte. Pour cela il doit se souvenir si l'action ayant eu lieu n observations plus tôt était un a ou un b. À cause de la boucle sur q_0, il ne peut pas savoir quand un c va avoir lieu, et doit donc retenir les n dernières observations. Ceci nécessite au moins 2^n états de mémoire. Nous obtenons donc la borne inférieure suivante :

Proposition 10. *Il existe une famille de pLTS* FA*-diagnostiquables $\{\mathcal{A}_n\}_{n\in\mathbb{N}}$ telle que $\mathcal{A}_n$ a $2n+2$ états et n'accepte pas de* IF/ FA/ IA*-diagnostiqueur avec moins de 2^n états de mémoire.*

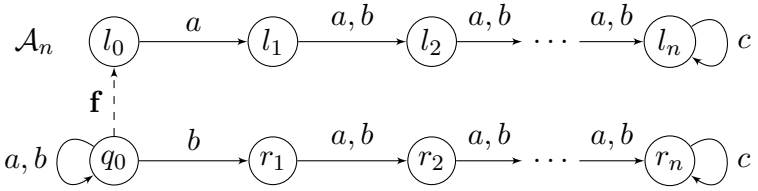

FIGURE 8 – Exemple d'un pLTS IA-diagnostiquable nécessitant que tout IA-diagnostiqueur ait une taille exponentielle.

3 Prédiction et prédiagnostic

Dans cette section, nous étudions la prédiction et introduisons le concept de prédiagnostic, une combinaison de la prédiction et du diagnostic.

3.1 Prédiction

3.1.1 Quelles formes de prédiction ?

Un système à évènements discrets est prédictible (k-prédictible) si la première faute est prédite avant son occurrence (resp. au moins k observations avant) et ce quel que soit le comportement futur du système. Deux adaptations sont possibles pour les pLTS : soit on conserve la définition d'origine et on demande qu'une fois prédite la faute aura lieu ou on relâche cette condition en exigeant seulement que la faute ait lieu avec probabilité 1.

Afin de raisonner sur la prédiction, nous aurons besoin d'un préfixe particulier d'une exécution. Pour une exécution finie ρ, et $k \in \mathbb{N}$, on définit $pre_k(\rho)$, le k-passé de ρ, par :

$$pre_k(\rho) = \rho_{\downarrow |\rho|_o - \min(k, |\rho|_o)}.$$

Exemple. Dans le pLTS de la figure 9, $pre_0(q_0bq_1\mathbf{f}q_2) = q_0bq_1$ car $\mathbf{f}$ est inobservable et $pre_1(q_0bq_1\mathbf{f}q_2) = q_0$. Plus généralement, pour $k \geq 1$, $pre_k(q_0bq_1\mathbf{f}q_2) = q_0$.

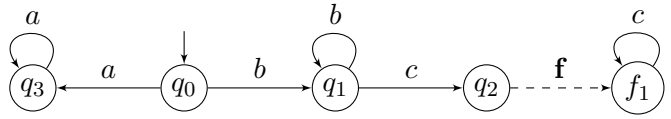

FIGURE 9 – Un pLTS 0-surement prédictible et 1-prédictible.

On introduit également des ensembles de séquences d'observation définies par leurs comportements futurs possibles. Informellement, une séquence observée σ interdit la prédiction d'une faute lorsque soit il existe une exécution correcte infinie pour laquelle σ est un préfixe de sa séquence d'observation (UPC) soit il existe un ensemble de mesure positive de telles exécutions (UPSC). Aussi pour être k-prédictible, k observations avant l'occurrence possible d'une faute, la séquence observée ne devrait pas appartenir à ces ensembles (voir la définition 8).

Définition 7. Soit σ une séquence d'observation finie d'un pLTS $\mathcal{A}$. Alors :
- σ est *ultimement potentiellement correcte* si $\{\rho' \in \Omega \mid \sigma \preceq \mathcal{P}(\rho')\} \cap \mathsf{C}_\infty \neq \emptyset$ où C_∞ est l'ensemble des séquences infinies correctes. L'ensemble des séquences d'observation ultimement potentiellement correctes est noté UPC.
- σ est *ultimement potentiellement significativement correcte* if $\mathbb{P}(\{\rho' \in \Omega \mid \sigma \preceq \mathcal{P}(\rho')\} \cap \mathsf{C}_\infty) > 0$. L'ensemble des séquences d'observation ultimement potentiellement significativement correctes est noté UPSC .

Définition 8. Soit $k \in \mathbb{N}$.
- Un pLTS $\mathcal{A}$ est *k-surement prédictible* si pour toute exécution $\rho\mathbf{f}q$ de $\mathcal{A}$, $\mathcal{P}(pre_k(\rho)) \notin \mathsf{UPC}$;
- Un pLTS $\mathcal{A}$ est *k-prédictible* si pour toute exécution $\rho\mathbf{f}q$ de $\mathcal{A}$, $\mathcal{P}(pre_k(\rho)) \notin \mathsf{UPSC}$.

On remarque que dans la définition précédente, on peut limiter la vérification de la condition aux exécutions correctes en considérant uniquement la première apparition de la faute dans l'exécution.

Exemple. Le pLTS de la figure 9 est 0-surement prédictible. Chaque exécution correcte ρ qui est suivie d'un $\mathbf{f}$ est telle que $\mathcal{P}(\rho) = b^n c$ où $n \geq 1$. Comme il n'y a qu'une exécution correspondant à cette séquence d'observation, la faute peut être prédite. Il n'est pas 1-surement prédictible car le 1-passé de $\rho = q_0bq_1cq_2\mathbf{f}f_1$ est $pre_1(\rho) = q_0bq_1$ et l'exécution infinie $\rho' = q_0(bq_1)^\omega$ est correcte. Par contre il est 1-prédictible car pour chaque exécution fermée dont la séquence d'observation est b^n pour $n \geq 1$ (donc finissant en q_1) une faute aura lieu avec probabilité 1. Enfin, il n'est pas 2-prédictible car le 2-passé de $\rho = q_0bq_1cq_2\mathbf{f}f_1$ est q_0 et l'exécution correcte infinie $\rho = q_0(aq_3)^\omega$ a probabilité $\frac{1}{2}$.

3.1.2 Complexité de la prédiction et construction des prédicteurs

Afin de déterminer une borne supérieure pour la complexité de la k-prédictibilité (sûre), on commence par fournir une caractérisation basée sur le graphe du pLTS. Dans ce qui suit, une *composante fortement connexe triviale* est constitué d'un unique état sans boucle.

Lemme 11. *Soit $\mathcal{A}$ un pLTS et $k \in \mathbb{N}$. $\mathcal{A}$ est k-surement prédictible (resp. k-prédictible) si et seulement si il n'existe pas de paire d'exécutions $q_0 \stackrel{\rho_0}{\Rightarrow} q_1$ et $q_0 \stackrel{\rho'_0}{\Rightarrow} q'_1$ telle que :*

- *$\mathcal{P}(\rho_0) = \mathcal{P}(\rho'_0)$;*
- *$q_1 \stackrel{\rho_1}{\Rightarrow} q' \xrightarrow{\mathbf{f}} q$ pour un $q' \in Q_c$ avec $|\rho_1|_o \leq k$;*
- *$q'_1 \stackrel{\rho_2}{\Rightarrow} q_2$, pour un $q_2 \in Q_c$ appartenant à une composante fortement connexe non triviale (resp. à une CFCT) de $\mathcal{A}$.*

Cette caractérisation se vérifie en NLOGSPACE sur le pLTS. Par ailleurs on montre que la k-prédictibilité (sûre) est NLOGSPACE-difficile par une réduction de l'accessibilité sur les graphes acycliques.

Théorème 12. *Étant donnés un pLTS $\mathcal{A}$ et $k \in \mathbb{N}$, décider si $\mathcal{A}$ est k-prédictible (resp. k-surement prédictible) est un problème* NLOGSPACE*-complet. De plus, la même complexité s'applique si k n'est pas fixé (et pas donné en entrée).*

De la même façon que nous avions défini des diagnostiqueurs, nous définissons les prédicteurs (sûrs).

Définition 9. Un *k-prédicteur* (resp. *k-prédicteur sûr*) est une fonction $D : \Sigma_o^* \to \{\top, ?\}$ telle que :

correction Pour toute exécution fermée ρ telle que $D(\mathcal{P}(\rho)) = \top$, $\mathbb{P}(\{\rho' \in \Omega \mid \rho \preceq \rho' \wedge \rho' \in \mathsf{C}_\infty\}) = 0$ (resp. pour toute exécution $\rho' \in \Omega$ telle que $\rho \preceq \rho'$, $\rho' \notin \mathsf{C}_\infty$).

réactivité Pour toute exécution fermée ρ, s'il existe ρ' avec $|\rho'|_o \leq k$ et $\rho\rho'$ fautive, alors $D(\mathcal{P}(\rho)) = \top$.

La proposition suivante établit que l'existence d'un k-prédicteur (resp. k-prédicteur sûr) est équivalente à la k-predictibilité (resp. k-sure predictibilité). De plus, une construction similaire aux automates des diagnostiqueurs fournit un prédicteur de taille exponentielle.

Proposition 13. *Un pLTS $\mathcal{A}$ est k-(surement) prédictible si et seulement si il admet un k-prédicteur (sûr). Dans le cas positif, $\mathcal{A}$ admet un k-prédicteur (sûr) avec au plus 2^{n_c} états où n_c est le nombre d'états corrects de $\mathcal{A}$.*

Exemple. Comme le pLTS de la figure 9 est 0-surement prédictible et 1-prédictible, on peut construire un 0-prédicteur sûr et un 1-prédicteur corrects. Cela est fait dans la figure 10 où les états doublement entourés correspondent à ceux où le prédicteur annonce $\top$. Intuitivement ces prédicteurs ont été construits par réduction du IF-automate qui consiste à maintenir l'ensemble des états corrects possibles.

Une borne inférieure sur la taille des prédicteurs peut être trouvée avec une famille de pLTS similaire à celle utilisée pour les diagnostiqueurs.

Proposition 14. *Il existe une famille $\{\mathcal{A}_n\}_{n\in\mathbb{N}}$ de pLTS 0-(surement) prédictibles où pour tout n, $\mathcal{A}_n$ a $2n+2$ états corrects, et $\mathcal{A}_n$ n'admet pas de 0-prédicteur (sûr) avec moins de 2^n états.*

3.2 Prédiagnostic

D'un côté, le diagnostic s'intéresse à la détection de fautes qui ont eu lieu : pour une séquence d'observation donnée, le diagnostiqueur cherche à détecter si une faute a eu lieu dans tous les passés

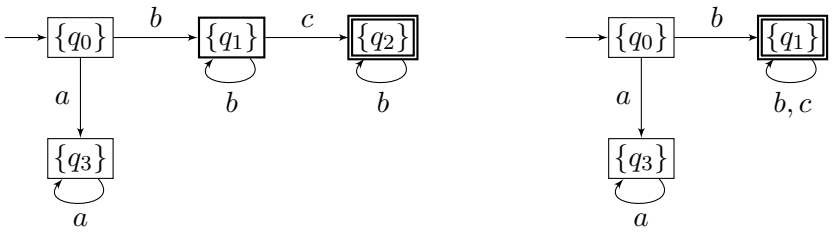

FIGURE 10 – À gauche un 0-prédicteur sûr du pLTS de la figure 9, à droite un 1-prédicteur.

possibles. De l'autre, la prédiction essaie d'anticiper les fautes : pour une séquence d'observation donnée, un prédicteur cherche à détecter si une faute aura lieu dans tous les futurs possibles. La notion que nous introduisons maintenant, le prédiagnostic, concerne la détection de fautes à la fois dans le passé et dans le futur.

Commençons par introduire deux ensembles d'exécutions infinies qui rendent le prédiagnostic impossible. FUPC_∞ est l'ensemble des exécutions fautives infinies pour lesquelles il existe, pour chacun de leurs préfixes finis, une exécution infinie correcte observationnellement équivalente. FUPSC_∞ (qui est inclus dans FUPC_∞) est l'ensemble des exécutions fautives infinies pour lesquelles il existe, pour chacun de leurs préfixes finis un ensemble de mesure positive d'exécutions correctes infinies observationnellement équivalentes.

Définition 10. Soit $\mathcal{A}$ un pLTS. Alors :
- FUPC_∞, l'ensemble des exécutions *fautives, ultimement possiblement correctes*, est défini par : $\mathsf{FUPC}_\infty = \{\rho \in \Omega \mid \rho \text{ fautive et } \forall i \in \mathbb{N},\ \mathcal{P}(\rho_{\downarrow i}) \in \mathsf{UPC}\}$
- FUPSC_∞, l'ensemble des exécutions *fautives, ultimement possiblement significativement correctes*, est définie par : $\mathsf{FUPSC}_\infty = \{\rho \in \Omega \mid \rho \text{ fautive et } \forall i \in \mathbb{N},\ \mathcal{P}(\rho_{\downarrow i}) \in \mathsf{UPSC}\}$

On introduit maintenant deux notions de prédiagnostiquabilité, *sûre* ou *presque sûre*, selon le choix du sous-ensemble, FUPC_∞ ou FUPSC_∞, à éviter presque surement.

Définition 11. Soit $\mathcal{A}$ un pLTS. Alors :
- $\mathcal{A}$ est *surement prédiagnostiquable* si $\mathbb{P}(\mathsf{FUPC}_\infty)=0$;
- $\mathcal{A}$ est *prédiagnostiquable* si $\mathbb{P}(\mathsf{FUPSC}_\infty)=0$.

Pour des pLTS à branchement infini, le prédiagnostic sûr se situe strictement entre le IF-diagnostic et le FF-diagnostic. Par conséquent, pour des pLTS finis, il coïncide avec ces notions impliquant que le problème de la prédiagnostiquabilité sûre est PSPACE-complet. Pour la prédiagnostiquabilité, nous avons établi une caractérisation similaire à celles du diagnostic qui conduit à la même complexité.

Théorème 15. *Le problème de la prédiagnostiquabilité (sûre) est* PSPACE*-complet.*

Le prédiagnostic permet de détecter les fautes plus rapidement que le diagnostic. Comme pour les diagnostiqueurs, nous supposons que les prédiagnostiqueurs (sûrs) s'engagent lorsqu'ils annoncent $\top$. Intuitivement, un prédiagnostiqueur sûr est un IF-diagnostiqueur avec la capacité de prédiction d'un prédicteur sûr, tandis qu'un prédiagnostiqueur est un IF-diagnostiqueur avec la capacité de prédiction d'un prédicteur.

Définition 12. Un *prédiagnostiqueur sûr* (resp. *prédiagnostiqueur*) est une fonction $D : \Sigma_o^* \to \{\top, ?\}$ telle que

correction Pour toute exécution fermée ρ, si $D(\mathcal{P}(\rho)) = \top$ alors $\{\rho' \in \Omega \mid \rho \preceq \rho'\} \subseteq \mathsf{Sf}_\infty$ (resp. $\mathbb{P}(\rho' \in \Omega \mid \rho \preceq \rho' \wedge \rho' \in \mathsf{C}_\infty) = 0$) où Sf_∞ est l'ensemble des exécutions infinies surement fautives.

réactivité Pour toute exécution fautive finie ρ, $\mathbb{P}(\{\rho' \in \Omega \mid \rho \preceq \rho' \wedge D(\mathcal{P}(\rho')) =?\}) = 0$ où pour $w \in \Sigma_o^\omega$, $D(w) = \lim_{n\to\infty} D(w_{\leq n})$.

Bien que le prédiagnostic sûr soit confondu avec le IF-diagnostic pour des pLTS finimement branchants, il y a des différences entre un prédiagnostiqueur sûr et un IF-diagnostiqueur. Tout IF-diagnostiqueur est un prédiagnostiqueur sûr mais un prédiagnostiqueur sûr peut rendre un verdict $\top$ plus tôt. Ce phénomène peut même avoir lieu si le pLTS n'est pas prédictible.

Exemple. Dans le pLTS non prédictible de la figure 11, un IF-diagnostiqueur peut rendre $\top$ après avoir observé deux a car une faute a alors nécessairement eu lieu. Cependant un prédiagnostiqueur sûr peut émettre $\top$ après un unique a car soit une faute a eu lieu, soit elle aura lieu.

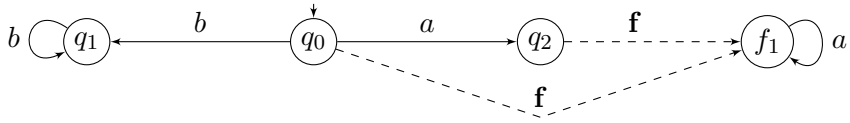

FIGURE 11 – Un pLTS non prédictible pour lequel un prédiagnostiqueur sûr est plus rapide qu'un IF-diagnostiqueur.

Comme souhaité, l'existence d'un prédiagnostiqueur (sûr) est équivalente à la prédiagnostiquabilité (sûre). Par ailleurs, avec les mêmes méthodes que pour les diagnostiqueurs, on obtient des bornes à la taille du prédiagnostiqueur (sûr).

Proposition 16. *Un pLTS $\mathcal{A}$ est (surement) prédiagnostiquable si et seulement si il admet un prédiagnostiqueur (sûr). Dans le cas positif, $\mathcal{A}$ possède un prédiagnostiqueur (sûr) avec au plus 2^{n_c} états où n_c est le nombre d'états corrects de $\mathcal{A}$. De plus, il existe une famille $(\mathcal{A}_n)$ de pLTS (surement) prédiagnostiquables telle que $\mathcal{A}_n$ a $n+1$ états corrects et $\mathcal{A}_n$ ne possède pas de prédiagnostiqueur (sûr) avec moins de 2^n états.*

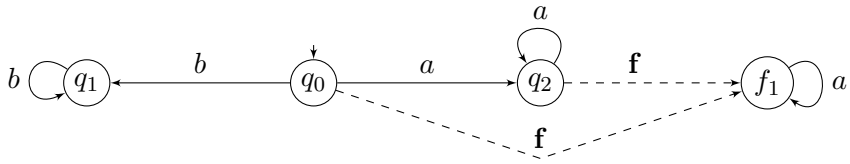

FIGURE 12 – Un pLTS non prédictible, non diagnostiquable et pourtant prédiagnostiquable.

Exemple. Le pLTS de la figure 12 n'est ni prédictible ni diagnostiquable quelle que soit la spécification. Par contre il est prédiagnostiquable. La figure 13 représente un prédiagnostiqueur possible. La faute est annoncée quand l'ensemble des états corrects que l'on a pu atteindre avec la séquence observée est réduit à $\{q_2\}$. On peut donc encore être dans un état correct, mais comme de q_2 une faute interviendra presque surement, le prédiagnostiqueur l'annonce.

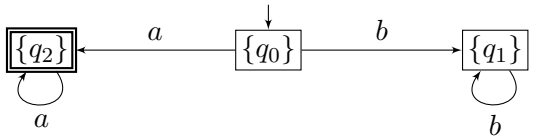

FIGURE 13 – Un prédiagnostiqueur pour le pLTS de la figure 12.

3.3 Diagnostic versus prédiction versus prédiagnostic

L'ensemble des relations entre les différentes formes de diagnostic, prédiction et prédiagnostic ainsi que la complexité des problèmes de décision sont résumés en figure 16. Illustrons maintenant certaines des non implications.

Le FA-diagnostic n'implique pas la prédiction. Le pLTS de la figure 11 n'est pas 0-prédictible, cependant il est FA-diagnostiquable.

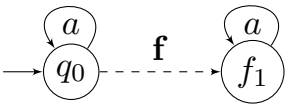

FIGURE 14 – Un pLTS k-predictible qui n'est pas IF-diagnostiquable.

La prédiction n'implique pas le IF-diagnostic. Le pLTS de la figure 14 est k-prédictible pour tout $k \in \mathbb{N}$ car la probabilité des exécutions correctes infinies est zéro. Cependant il n'est pas IF-diagnostiquable car la séquence d'observation a^ω est ambigüe, et l'exécution $q_0\mathbf{f}f_1(af_1)^\omega$ a probabilité $\frac{1}{2}$.

La prédiagnostiquabilité sûre n'implique pas la FF-diagnostiquabilité. Observons le pLTS infinimement branchant de la figure 15. Il ne contient aucune exécution infinie correcte, ainsi $\mathsf{UPC} = \emptyset$, donc ce pLTS est surement prédiagnostiquable. Par contre, pour tout $n \geq 1, \mathbb{P}(\mathsf{FAmb}_n) = \frac{1}{2}$, donc ce pLTS n'est pas FF-diagnostiquable.

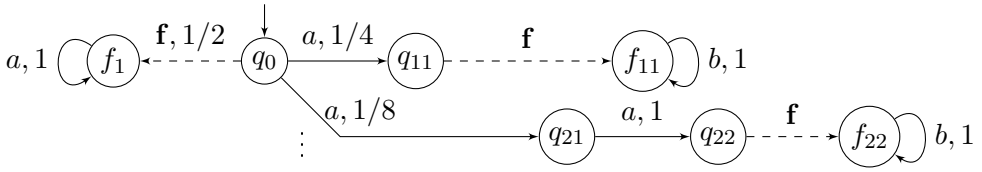

FIGURE 15 – Un pLTS infiniment branchant qui est sûrement prédiagnostiquable mais pas FF-diagnostiquable.

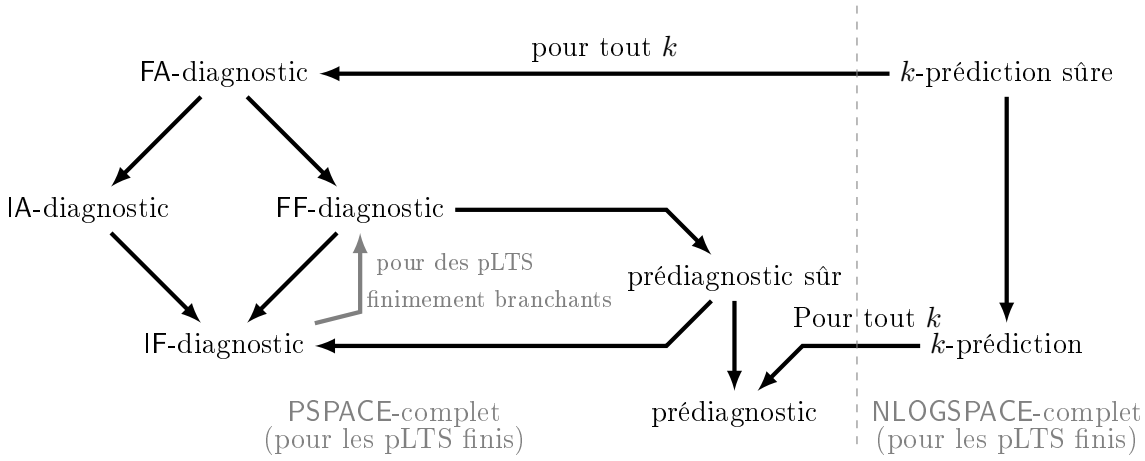

FIGURE 16 – Résumé des liens entre les différentes specifications, et leur complexité.

4 Travaux en cours

4.1 Diagnostic approché

Jusqu'à présent la condition de correction exigeait que l'information donnée par le diagnostiqueur soit exacte. Pour le diagnostic approché (AA-diagnostic), nous autorisons le diagnostiqueur à se tromper à condition que la probabilité d'erreur puisse être choisie arbitrairement petite.

Définition 13. Un pLTS $\mathcal{A}$ est AA-diagnostiquable si et seulement si pour tout $\varepsilon > 0$ et pour toute exécution fautive ρ on a :

$$\lim_{n\to\infty} \frac{\mathbb{P}(\{\rho' \in \mathsf{SR}_{n+|\rho|_o} \mid \rho \preceq \rho' \wedge \mathbb{P}_{amb}(\mathcal{P}(\rho')) > \varepsilon\})}{\mathbb{P}(\{\rho' \in \mathsf{SR}_{n+|\rho|_o} \mid \rho \preceq \rho'\})} = 0$$

où $\mathbb{P}_{amb}(\sigma) = \frac{\mathbb{P}(\{\rho' \in \mathsf{SR}_{|\sigma|} \mid \mathcal{P}(\rho') = \sigma \wedge \rho' \in \mathsf{C}_{|\sigma|}\})}{\mathbb{P}(\{\rho' \in \mathsf{SR}_{|\sigma|} \mid \mathcal{P}(\rho') = \sigma\})}$ (avec C_n, l'ensemble des exécutions infinies dont le préfixe de longueur observable n est correct) représente la probabilité que la séquence soit correcte.

La AA-diagnostiquabilité induit une définition du diagnostiqueur : la réactivité est assurée car la limite tend vers 0 et la correction vient du fait que la probabilité d'ambiguité, $\mathbb{P}_{amb}$, peut être choisie arbitrairement faible. Remarquons qu'ainsi présenté, un diagnostiqueur n'a pas forcément une mémoire finie. Cette notion de diagnostiquabilité a été étudiée dans [4] où une procédure de décision PTIME est proposée. En annexe, nous décrivons cet algorithme et donnons un exemple de pLTS sur lequel il n'est pas correct.

Fait 17. *La procédure de décision de [4] pour la AA-diagnostiquabilité est erronée.*

Nous avons étudié une notion proche de la AA-diagnostiquabilité : la AA-ε-diagnostiquabilité. Ici on fixe $\varepsilon > 0$ et on recherche un diagnostiqueur se trompant avec probabilité inférieure à ε.

Définition 14. Un pLTS $\mathcal{A}$ est AA-ε-diagnostiquable si et seulement si pour toute exécution fautive ρ on a :

$$\lim_{n\to\infty} \frac{\mathbb{P}(\{\rho' \in \mathsf{SR}_{n+|\rho|_o} \mid \rho \preceq \rho' \wedge \mathbb{P}_{amb}(\mathcal{P}(\rho')) > \varepsilon\})}{\mathbb{P}(\{\rho' \in \mathsf{SR}_{n+|\rho|_o} \mid \rho \preceq \rho'\})} = 0$$

Afin d'identifier la complexité de ce problème, nous avons réduit le problème du vide pour les automates probabilistes : étant donnés un automate probabiliste $\mathcal{A}$ et une valeur de seuil $0 < \lambda < 1$, existe-t-il un mot w tel que $\mathcal{A}$ accepte w avec probabilité supérieure à λ. Ce problème est indécidable [7]. même en supposant l'automate *quasi-acyclique*. Un automate ou un pLTS est quasi-acyclique si toutes ses composantes fortement connexes du graphe sous-jacent ne sont composées que d'un seul état.

Proposition 18. *Le problème du vide pour les automates probabilistes quasi-acycliques est indécidable.*

Nous réduisons le problème du vide au problème de la AA-ε-diagnostiquabilité et cette réduction ne modifiant pas le quasi-acyclisme du système étudié, on obtient :

Théorème 19. *Pour tout $\varepsilon > 0$, le problème de la AA-ε-diagnostiquabilité pour les pLTS quasi-acycliques est indécidable.*

L'indécidabilité de la AA-ε-diagnostiquabilité ne semble pas s'adapter à la AA-diagnostiquabilité. En effet nous conjecturons que l'algorithme de [4] est correct pour les pLTS quasi-acycliques.

4.2 Optimisation et diagnostic

Dans les sytèmes que nous avons étudiés jusqu'à présent, la structure d'observation était statique et décrite par Σ_o. On désire désormais minimiser les capacités d'observation des détecteurs, que ce soit spatialement ou temporellement. Tout en préseservant la diagnostiquabilité du système, (1) on peut rechercher un Σ_o de taille minimale ou (2) observer le système à certains instants. C'est cette deuxième option que nous avons commencé à étudier en travaillant sur des jeux se déroulant sur une chaîne de Markov discrète. Le but du jeu est de détecter que l'on a atteint un état spécifique q_f et ce en limitant l'espérance du nombre d'observations.

Définition 15. Une *chaîne de Markov discrète* (DTMC) est un tuple $\mathcal{A} = \langle Q, \mathbf{P} \rangle$ où :
- Q est un ensemble d'états ;
- $\mathbf{P}$ est une fonction de $Q \times Q$ vers $\mathbb{Q}_{\geq 0}$ satisfaisant pour tout $q \in Q$: $\sum_{q' \in Q} \mathbf{P}[q, q'] = 1$.

Par analogie avec les diagnostiqueurs à mémoire finie qui induisent un diagnostiqueur, nous définissons ici des stratégies abstraites qui induisent une stratégie. Cette stratégie peut ensuite être utilisée dans un jeu sur une chaîne de Markov discrète.

Définition 16. Une *stratégie abstraite* est une fonction $d : (\mathbb{N} \cdot Q)^* \to \mathbb{N}$. Étant donnée d, on introduit deux fonctions auxiliaires $\mathsf{wait}_d : Q^* \to \mathbb{N}$, qui indique combien d'instants il faut attendre avant la prochaine observation et $\mathsf{mem}_d : Q^* \to \mathbb{N} \cdot (Q \cdot \mathbb{N})^*$ qui contient l'historique des observations. Ces deux fonctions sont définies inductivement par :
- $\mathsf{mem}_d(\varepsilon) = \mathsf{wait}_d(\varepsilon) = d(\varepsilon)$;
- Si $\mathsf{wait}_d(\rho) > 0$ alors $\mathsf{mem}_d(\rho q) = \mathsf{mem}_d(\rho)$, $\mathsf{wait}_d(\rho q) = \mathsf{wait}_d(\rho) - 1$;
- Si $\mathsf{wait}_d(\rho) = 0$ alors $\mathsf{wait}_d(\rho q) = d(\mathsf{mem}_d(\rho)q)$, $\mathsf{mem}_d(\rho q) = \mathsf{mem}_d(\rho)q\mathsf{wait}_d(\rho q)$.

La *stratégie* σ_d associée à d est une fonction $\sigma_d : Q^* \to \{\bot, \top\}$ définie par :

$$\text{Si } \mathsf{wait}_d(\rho) > 0 \text{ alors } \sigma_d(\rho) = \bot \text{ sinon } \sigma_d(\rho) = \top.$$

Étant données une exécution $\rho \in Q^\omega$ et une stratégie abstraite d, le coût de ρ par d est défini par : $cost_d(\rho) = \min(|\mathcal{P}_Q(\mathsf{mem}_d(\rho')) \mid \rho' q_f \preceq \rho \wedge \sigma_d(\rho' q_f) = \top)$. Le problème d'optimisation consiste à trouver pour tout $q \in Q$ une stratégie abstraite d_q telle que : $\mathbb{E}_q(cost_{d_q}) = \inf_d(\mathbb{E}_q(cost_d)) \stackrel{\text{def}}{=} val(q)$. Cependant une telle stratégie n'existe pas forcément. Dans la chaîne de Markov de la figure 17, si on commence en q, plus $d(q)$ est grand, plus le coût de la stratégie associée à d est proche de 0, mais aucune stratégie n'a pour coût 0.

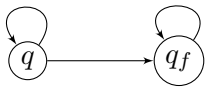

FIGURE 17 – Un exemple de chaîne de Markov.

Par conséquent nous nous intéressons au problème d'optimisation approché qui consiste à trouver pour tout $\varepsilon > 0$ et tout $q \in Q$ une stratégie abstraite $d_{q,\varepsilon}$ telle que : $\mathbb{E}_q(cost_{d_{q,\varepsilon}}) - \varepsilon \leq \inf_d(\mathbb{E}_q(cost_d)) \stackrel{\text{def}}{=} val(q)$.

Nous supposons qu'il existe une distribution stationnaire π_∞ indépendante de l'état initial telle que $\pi_\infty(q_f) > 0$. Nous avons établi le théorème suivant (dont la preuve n'est pas constructive).

Théorème 20. *Étant donné une DTMC $\mathcal{A}$, $\varepsilon > 0$ et un état initial $q \in Q$, il existe une stratégie abstraite stationnaire $d_{q,\varepsilon}$ telle que $\mathbb{E}_q(cost_{d_{q,\varepsilon}}) - \varepsilon \leq val(q)$.*

4.3 Détection d'état

La détection d'état a été introduite pour les systèmes probabilistes dans [11]. Elle consiste à déterminer grâce aux observations dans quel état le système se trouve.

Définition 17. Un pLTS $\mathcal{A}$ est A-détectable si et seulement si $\forall \varepsilon > 0$, $\exists N_\varepsilon \in \mathbb{N}$, pour tout $n \geq N_\varepsilon$, $\mathbb{P}(\{\rho \in \mathsf{SR}_n \mid \sharp_Q(\rho) > 1\}) < \varepsilon$ où $\sharp_Q(\rho) = |\{q \in Q \mid \rho' q \in \mathsf{SR}_{|\rho|_o} \text{ et } \mathcal{P}(\rho) = \mathcal{P}(\rho' q)\}|$ est le nombre d'états dans lequel le système peut être après avoir observé $\mathcal{P}(\rho)$.

Dans [11] une caractérisation de la A-détection est proposée. À l'aide de cette caractérisation et en réutilisant la réduction de l'éventuelle universalité on obtient :

Théorème 21. *Le problème de la A-détection est* PSPACE*-complet.*

Conclusion

Dans ce travail, nous avons établi les fondations du diagnostic et de la prédiction pour les systèmes probabilistes partiellement observables. En particulier, nous avons étudié les problèmes sémantiques et proposé plusieurs définitions significatives pour la diagnostiquabilité et la prédictabilité dans un contexte probabiliste. Nous avons également introduit le prédiagnostic qui combine les avantages du diagnostic et de la prédiction. Après avoir fourni les relations existantes entre ces notions, nous avons utilisé des caractérisations, obtenues à partir du produit du système étudié avec un observateur approprié, afin d'obtenir les complexités exactes de ces concepts. Ces complexités sont de NLOGSPACE pour la prédictibilité et de PSPACE pour la diagnostiquabilité et la prédiagnostiquabilité, (cf. figure 16). Nous avons également prouvé des bornes supérieures et inférieures exponentielles pour la synthèse des diagnostiqueurs, des prédicteurs et des prédiagnostiqueurs. Nous avons étudié comment nos résultats pouvaient s'étendre à d'autres questions proches comme la détection d'état. L'algorithme de Chen et Kumar étant erroné, il n'existe pas à l'heure actuelle d'algorithme de décision pour la AA-diagnostiquabilité. Nous avons établi que des problèmes proches étaient indécidables mais le statut de l'AA-diagnostiquabilité reste un problème ouvert. Enfin, nous avons également commencé des travaux relatifs à l'optimisation du diagnostic en donnant par exemple un coût à l'observation d'un évènement du système.

Notre contribution ouvre quelques perspectives de recherche intéressantes. Le problème du diagnostic est issu de la communauté de l'automatique et du contrôle. Il serait intéressant de développer un prototype qui pourrait être un point de départ à une collaboration avec des chercheurs de cette communauté. Par ailleurs, nos algorithmes de décision et de synthèse s'appliquent uniquement aux systèmes finis. Que peut-on obtenir avec des systèmes infinis ? Quelles notions de diagnostiquabilité sont intéressantes dans ce cadre ? Un axe de recherche à entreprendre consisterait à étudier les systèmes infinis munis d'un attracteur fini [2].

Références

[1] N. Bertrand, S. Haddad, and E. Lefaucheux. Foundation of diagnosis and predictability in probabilistic systems. Research Report LSV-14-09, June 2014. Available at `http://www.lsv.ens-cachan.fr/Publis/RAPPORTS_LSV/PDF/rr-lsv-2014-09.pdf`.

[2] N. Bertrand and P. Schnoebelen. Solving stochastic Büchi games on infinite arenas with a finite attractor. In *Proceedings 11th International Workshop on Quantitative Aspects of Programming Languages and Systems, QAPL 2013*, volume 117 of *EPTCS*, pages 116–131, 2013.

[3] B.G. Buchanan and E.H. Shortliffe. *Rule Based Expert Systems : The MYCIN Experiments of the Stanford Heuristic Programming Project*. Reading, MA : Addison-Wesley, 1984.

[4] J. Chen and R. Kumar. Polynomial test for stochastic diagnosability of discrete-event systems. *IEEE Transactions on Automation Science and Engineering*, 10(4) :969–979, 2013.

[5] L. Doyen, T. A. Henzinger, and J-F. Raskin. Equivalence of labeled Markov chains. *International Journal of Foundations of Computer Science*, 19(3) :549–563, 2008.

[6] S. Genc and S. Lafortune. Predictability of event occurrences in partially-observed discrete-event systems. *Automatica*, 45(2) :301–311, 2009.

[7] H. Gimbert and Y. Oualhadj. Probabilistic automata on finite words : Decidable and undecidable problems. In *ICALP (2)*, volume 6199 of *Lecture Notes in Computer Science*, pages 527–538. Springer, 2010.

[8] S. Haar, S. Haddad, T. Melliti, and S. Schwoon. Optimal constructions for active diagnosis. In *Proceedings of FSTTCS'13*, volume 24 of *LIPIcs*, pages 527–539. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2013.

[9] S. Jiang, Z. Huang, V. Chandra, and R. Kumar. A polynomial algorithm for testing diagnosability of discrete-event systems. *IEEE Trans. Aut. Cont.*, 46(8) :1318–1321, 2001.

[10] N.D. Jones. Space-bounded reducibility among combinatorial problems. *Journal of Computer and System Sciences*, 11(1) :68–85, 1975.

[11] C. Kerouglou and C.N. Hadjicostis. Detectability in stochastic discrete event systems. In *Preprints 12th International Workshop on Discrete Event Systems, WODES 2014*, pages 27–38, 2014.

[12] N. Rampersad, J. Shallit, and Z. Xu. The computational complexity of universality problems for prefixes, suffixes, factors, and subwords of regular languages. *Fundam. Inf.*, 116(1-4) :223–236, January 2012.

[13] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis. Diagnosability of discrete-event systems. *IEEE Trans. Aut. Cont.*, 40(9) :1555–1575, 1995.

[14] D. Thorsley and D. Teneketzis. Diagnosability of stochastic discrete-event systems. *IEEE Transactions on Automatic Control*, 50(4) :476–492, 2005.

A Preuves

A.1 Preuves de la section 2

Preuve de la proposition 4.
Supposons d'abord que $\mathcal{A}_{\mathsf{IA}}$ a une CFCT avec (au moins) un état (q, U, V, W) tel que $q \in Q_f$ et $U \neq \emptyset$. Soit ρ une exécution fermée menant de l'état initial s_0 de $\mathcal{A}_{\mathsf{IA}}$ à s. Maintenant, pour tout état $s' = (q', U') \in C$, obligatoirement $q' \in Q_f$ et $U' \neq \emptyset$, car C est fortement connexe. Donc pour toute exécution fermée ρ' étendant ρ, en écrivant $s' = (q', U')$ pour l'état atteint par ρ', Il existe une exécution fermée correcte ρ'' telle que $\mathcal{P}(\rho'') = \mathcal{P}(\rho')$ et $q_0 \xrightarrow{\rho''} q''$ avec $q'' \in U'$. Par conséquent, la séquence d'observation $\mathcal{P}(\rho'')$ est ambigüe, et pour tout $n \geq |\rho|_o$, $\mathbb{P}(\mathsf{FAmb}_n) \geq \mathbb{P}(\rho)$, ainsi $\mathcal{A}$ n'est pas IA-diagnostiquable.

Supposons qu'une CFCT C de $\mathcal{A}_{\mathsf{IA}}$ ait tous ses états (q, U, V, W) avec $q \in Q_c$ et $W \neq \emptyset$. Alors aucun de ses états n'est acceptant pour l'automate déterministe de Büchi $\mathsf{IA}(\mathcal{A})$. Soit ρ une exécution fermée finie terminant dans C. Par la proposition 3, toute exécution infinie ρ' étendant ρ est ambigüe. Comme $q \in Q_c$, $\mathbb{P}(\mathsf{CAmb}_\infty) \geq \mathbb{P}(\rho) > 0$. Par conséquent $\mathcal{A}$ n'est pas IA-diagnostiquable.

Supposons maintenant que $\mathcal{A}_{\mathsf{IA}}$ n'a pas de CFCT telle que, soit tous ses états (q, U, V, W) vérifient $q \in Q_f$ et $U \neq \emptyset$, ou tous ses états (q, U, V, W) vérifient $q \in Q_c$ et $W \neq \emptyset$. On observe d'abord que si certaines CFCT de $\mathcal{A}_{\mathsf{IA}}$ contiennent des états (q, U, V, W) avec $q \in Q_f$ et $U \neq \emptyset$, alors tous ses états satisfont les mêmes contraintes. De plus, si certains états (q, U, V, W) d'une CFCT ont $q \in Q_c$, alors tous les états de cette CFCT ont leur q-composante dans Q_c. Par conséquent, la condition peut être reformulée comme suit. Toutes les CFCT C de $\mathcal{A}_{\mathsf{IA}}$ satisfont :

- soit tous les états (q, U, V, W) de C satisfont $q \in Q_f$ et $U = \emptyset$;
- soit tous les états (q, U, V, W) de C satisfont $q \in Q_c$ et certains états (q, U, V, W) de C satisfont $W = \emptyset$.

Quel que soit le cas, toutes contiennent (au moins) un état acceptant pour la condition de Büchi de $\mathsf{IA}(\mathcal{A})$. Comme presque surement toute exécution entre dans une CFCT et visite chacun de ses états infiniment souvent, d'après la proposition 3, les exécutions de $\mathcal{A}_{\mathsf{IA}}$ ne sont pas ambigües avec probabilité 1. Ceci prouve que $\mathcal{A}$ est IA-diagnostiquable. □

Preuve du théorème 5.

Nous développons ici la preuve dans le cas de la IA-diagnostiquabilité.

Proposition 22. *Le problème de la* IA*-diagnostiquabilité est dans* PSPACE.

Démonstration. On utilise la caractérisation de la IA-diagnostiquabilité donnée dans la proposition 4 sans construire explicitement $\mathcal{A}_{\mathsf{IA}}$. Étant donnés deux états s, s' de $\mathcal{A}_{\mathsf{IA}}$, on peut vérifier en espace polynomial si l'un est accessible depuis l'autre. En utilisant cette procédure, on peut vérifier si un état s n'est pas dans une CFCT en devinant un autre état s' tel que s' est accessible depuis s mais s n'est pas accessible depuis s' (on utilise ici le théorème de Savitch).

Par conséquent la procédure décidant si $\mathcal{A}$ n'est pas IA-diagnostiquable consiste à deviner un état $s = (q, U, V, W)$, vérifier qu'il est accessible depuis s_0 et qu'il appartient à une CFCT où tous les états (q', U', V', W') satisfont $U' \neq \emptyset$ et $W' \neq \emptyset$ (on vérifie que s est dans une CFCT et qu'il ne peut atteindre aucun état contredisant cette propriété). □

Pour montrer la PSPACE-difficulté de la IA-diagnostiquabilité, on utilise l'éventuelle universalité. Établissons donc la proposition 6.

Preuve de la proposition 6.
Soit $\mathcal{A} = (Q, \Sigma, T, q_0, F)$ un NFA. À partir de $\mathcal{A}$ nous définissons le NFA $\mathcal{A}' = (Q', \Sigma', T', q_0, Q')$

où $\Sigma' = \Sigma \uplus \{\sharp\}$, $Q' = Q \uplus \{s\}$, et

$$T' = T \cup \{(q, \sharp, q_0) \mid q \in F\} \cup \{(s, a, s) \mid a \in \Sigma\} \cup \{(q, a, s) \mid a \in \Sigma,\ q \not\to_{\mathcal{A}}\} \ .$$

Supposons d'abord que $\mathcal{L}(\mathcal{A}) = \Sigma^*$. Tout mot w sur l'alphabet Σ' peut être décomposé en $w = w_0\sharp w_1\sharp \dots \sharp w_n$ with $w_i \in \Sigma^*$. Pour chaque facteur w_i, comme $\mathcal{A}$ est universel, il existe une exécution ρ_i sur w finissant dans un état terminal q_i de $\mathcal{A}$. Par conséquent w est accepté par $\mathcal{A}'$ par l'exécution $\rho_0\sharp\rho_1\sharp \dots \sharp\rho_n$. Ainsi $\mathcal{A}'$ est universel, et donc éventuellement universel : $\varepsilon^{-1}\mathcal{L}(\mathcal{A}') = \Sigma'^*$.

Inversement, supposons que $\mathcal{A}'$ est éventuellement universel et soit $v \in \Sigma'^*$ un mot tel que $v^{-1}\mathcal{L}(\mathcal{A}') = \Sigma'^*$. Étant donné $w \in \Sigma^*$, on considère le mot $w' = v\sharp w\sharp$. Puisque $\mathcal{A}'$ est éventuellement universel avec v pour témoin, (1) $w' \in \mathcal{L}(\mathcal{A}')$ et (2) une exécution acceptante peut être décomposée en $\rho\sharp\rho'\sharp q_0$ où l'exécution ρ', qui correspond au mot w, a q_0 pour état initial, fini dans un état terminal de $\mathcal{A}$ et n'utilise que des transitions de $\mathcal{A}$. Donc ρ' est une exécution de $\mathcal{A}$ acceptant w. Par conséquent $w \in \mathcal{L}(\mathcal{A})$, et $\mathcal{A}$ est universel. □

Proposition 23. *Le problème de la* IA*-diagnostiquabilité est* PSPACE*-difficile.*

Démonstration. La preuve s'obtient par réduction du problème de l'éventuelle universalité et utilise la caractérisation de la proposition 4. Soit $\mathcal{A}$ un NFA vivant sur l'alphabet Σ, où tous les états sont terminaux. À partir de $\mathcal{A}$, on construit un pLTS vivant $\mathcal{A}'$ comme montré en figure 6. Dans $\mathcal{A}'$, l'ensemble des actions est $\Sigma \uplus \{u, \mathbf{f}, \sharp\}$, et u et $\mathbf{f}$ sont inobservables. Alors $\mathcal{A}$ est éventuellement universel si et seulement si $\mathcal{A}'$ n'est pas IA-diagnostiquable.

Observons plus attentivement le pLTS produit $\mathcal{A}'_{\mathsf{IA}}$. Pour $w \in \Sigma^+$, si $\mathbf{f}w$ mène à un état (f_0, U, V, W) de $\mathcal{A}'_{\mathsf{IA}}$, par construction de $\mathcal{A}'$, U correspond au sous-ensemble d'états accessibles dans $\mathcal{A}$ après avoir lu w et $W = \{f_0\}$. Autrement formulé, (1) soit $w \in \mathcal{L}(\mathcal{A})$ et l'ensemble non vide U consiste en la déterminisation au vol de $\mathcal{A}$, (2) soit $w \notin \mathcal{L}(\mathcal{A})$ et $U = \emptyset$.

Supposons d'abord que $\mathcal{A}'$ n'est pas IA-diagnostiquable. Par la proposition 4, $\mathcal{A}'_{\mathsf{IA}}$ contient une CFCT accessible $\mathcal{C}$ où tous les états $s = (q, U, V, W) \in \mathcal{C}$ satisfont $U \neq \emptyset$ et $W \neq \emptyset$. $q = f_0$ car si un $\sharp$ a été lu, $W = \emptyset$ et f_0 est le seul état d'une CFCT de $\mathcal{A}'$ accessible sans lire de $\sharp$. Comme $\mathcal{C}$ est une CFCT, et comme f_0 est un état puits de $\mathcal{A}'$ où toutes les actions de Σ sont franchissables, pour tout $v \in \Sigma^*$ tel que $\mathbf{f}v$ aille de l'état initial à $s \in \mathcal{C}$ dans $\mathcal{A}'_{\mathsf{IA}}$, on doit avoir $v^{-1}\mathcal{L}(\mathcal{A}) = \Sigma^*$. Ainsi, $\mathcal{A}$ est éventuellement universel.

Inversement, supposons qu'il existe un mot $v \in \Sigma^*$ tel que $v^{-1}\mathcal{L}(\mathcal{A}) = \Sigma^*$. Bien sûr, tout mot étendant v est aussi un témoin de l'éventuelle universalité de $\mathcal{A}$. Soit $v' \in \Sigma^*$ tel que, dans $\mathcal{A}'_{\mathsf{IA}}$, l'exécution lisant $\mathbf{f}vv'$ finit dans une CFCT $\mathcal{C}$. Comme $(vv')^{-1}\mathcal{L}(\mathcal{A}) = \Sigma^*$, tous les états de $\mathcal{C}$ sont de la forme (f_0, U, V, W) avec $U \neq \emptyset$ et $W \neq \emptyset$. Ainsi, par la proposition 4, $\mathcal{A}'$ n'est pas IA-diagnostiquable. □

(fin de la preuve du théorème 5)

□

Preuve de la proposition 8 .

Supposons d'abord qu'il existe un IA-diagnostiqueur D pour $\mathcal{A}$, et soit ρ une exécution infinie. Grâce à la condition de réaction, presque surement $D_{\sup}(\mathcal{P}(\rho)) \in \{\top, \bot\}$. Si $D_{\sup}(\mathcal{P}(\rho)) = \top$ alors il existe un n tel que $D(\mathcal{P}(\rho_{\downarrow n})) = \top$. Grâce à la condition de correction, $\rho_{\downarrow n}$ est surement fautif et donc ρ est surement fautif. Si $D_{\sup}(\mathcal{P}(\rho)) = \bot$, alors ρ est surement correcte. En effet, remarquons que le diagnostiqueur émet infiniment souvent $\bot$, donc par correction, pour tout n, $\mathcal{P}(\rho_{\downarrow n})$ est surement correcte et donc en particulier $\rho_{\downarrow n}$ est correcte. Supposons qu'il existe une

exécution fautive infinie ρ' avec $\mathcal{P}(\rho') = \mathcal{P}(\rho)$. Il existe n tel que pour tout $m \geq n$, $\rho'_{\downarrow n}$ est fautive. Donc, par correction, il ne peut y avoir plus de n $\perp$ verdicts pour $\mathcal{P}(\rho)$ ce qui contredit le fait que $D_{\sup}(\mathcal{P}(\rho)) = \perp$. Par conséquent avec probabilité 1, une exécution infinie n'est pas ambigüe.

Inversement, supposons que $\mathcal{A}$ est IA-diagnosable, et notons IA($\mathcal{A}$) son IA-automate. Pour tout mot $w \in \Sigma_o^*$, on représente par (U_w, V_w, W_w) l'état de IA($\mathcal{A}$) atteint après avoir observé w. Pour toute exécution fermée finie ρ de $\mathcal{A}$, on note $(U_\rho, V_\rho, W_\rho) = (U_{\mathcal{P}(\rho)}, V_{\mathcal{P}(\rho)}, W_{\mathcal{P}(\rho)})$. La fonction D est ensuite définie par : $D(w) = \top$ iff $U_w = \emptyset$, $D(w) = \perp$ ssi $W_w = \emptyset$ et $U_w \neq \emptyset$, et dans tous les autres cas $D(w) =?$. Prouvons maintenant que D est un IA-diagnostiqueur de $\mathcal{A}$.

correction Pour un mot w, si $U_w = \emptyset$, par construction de IA($\mathcal{A}$), w est surement fautif. Supposons maintenant que $W_w = \emptyset$ et $U_w \neq \emptyset$. Soit w' le plus grand préfixe de w tel que $W_{w'} = \emptyset$. Soit ρ une exécution fermée avec $\mathcal{P}(\rho) = w$. Supposons que $\rho_{\downarrow|w'|}$ est fautive. Donc les états visités par $\rho_{\downarrow n}$ pour $|w'| < n \leq |w|$ sont toujours dans $W_{\rho_{\downarrow n}}$. Comme $W_w = \emptyset$, ceci n'est pas possible et donc $\rho_{\downarrow|w'|}$ est correcte. Par conséquennt chaque fois qu'un état a $W = \emptyset$, la taille de son plus grand préfixe, pour lequel toute les exécutions fermées correspondantes à ce préfixe sont correctes, a augmenté. Ceci établit la correction.

réactivité Soit ρ une exécution infinie telle que $D_{sup}(\mathcal{P}(\rho)) =?$. De par la caractérisation de la proposition 4, soit (1) la composante fortement connexe de $\mathcal{A}_{\mathsf{IA}}$ que ρ visite infiniment souvent n'est pas une CFCT soit (2) ρ ne visite pas infiniment souvent tous les états de cette CFCT. La probabilité de telles exécutions est nulle, ce qui établit la réactivité.

□

Preuve de la proposition 10.

Intéressons nous à l'exemple de la figure 8 où $\Sigma_o = \{a, b, c\}$ et l'état initial est q_0. Considérons une exécution fautive finie incluant une action c. Sa séquence d'observation appartient à $\mathcal{L} = \{a, b\}^* b \{a, b\}^{n-1} c^+$. Puisque toute exécution correcte finie a une séquence d'observation appartenant à $\mathcal{L}' = \{a, b\}^* \cup \{a, b\}^* a \{a, b\}^{n-1} c^+$ et $\mathcal{L} \cap \mathcal{L}' = \emptyset$, $\mathsf{FAmb}_n \uplus \mathsf{CAmb}_n \subseteq \{\rho \mid \mathcal{P}(\rho) \in \{a, b\}^n\}$. Comme $\lim_{n\to\infty} \mathbb{P}(\{\rho \mid \mathcal{P}(\rho) \in \{a, b\}^n\}) = 0$, le pLTS est FA-diagnostiquable et donc IA-diagnostiquable.

Intuitivement, quand un c est observé, un IA-diagnostiqueur doit se souvenir de l'action observable ayant eu lieu n étapes plus tôt afin de savoir si l'exécution est fautive ou non. Donc, il doit retenir les n dernières actions observables au cas où un c ait lieu.

Plus formellement, supposons qu'il existe un diagnostiqueur $D = (M, \Sigma, m_0, \mathsf{up}, D_{fm})$ avec moins de 2^n états de mémoires. Il existe donc deux mots différents $w_1 \in \{a, b\}^n$ et $w_2 \in \{a, b\}^n$ menant au même état de mémoire : $\mathsf{up}(m_0, w_1) = \mathsf{up}(m_0, w_2)$. Les mots w_1 et w_2 diffèrent au moins d'une lettre, disons $w_1[i] = a$ et $w_2[i] = b$. Considérons pour $k \geq 1$, l'exécution fermée correcte $\rho_{1,k}$ correspondant à la séquence d'observation $w_1 a^{i-1} c^k$ dont la séquence d'états visités est $q_0^i r_1 \dots r_n^{k+1}$ et l'exécution fermée fautive $\rho_{2,k}$ correspondant à la séquence d'observation $w_2 a^{i-1} c^k$ dont la séquence d'états visités est $q_0^i l_0 l_1 \dots l_n^{k+1}$. Elles mènent aussi au même état de mémoire. Par correction, $D(w_1 a^{i-1} c^k) =?$. Donc pour tout suffixe ρ de $\rho_{2,1}$, $D(\rho) =?$ ce qui contredit la réactivité de D. □

A.2 Preuves de la section 3

Preuve du lemme 11.

Supposons qu'il existe une telle paire (ρ_0, ρ'_0). Soit $\rho = q_0 \overset{\rho_0}{\Longrightarrow} q_1 \overset{\rho_1}{\Longrightarrow} q'$ avec $|\rho_1|_o \leq k$ et $\rho^* = q_0 \overset{\rho'_0}{\Longrightarrow} q'_1 \overset{\rho_2}{\Longrightarrow} q_2$. On observe que $pre_k(\rho) \preceq \rho_0$, donc en choisissant pour ρ_p l'exécution fermée $\rho^*_{\downarrow|pre_k(\rho)|_o}$, on a $\mathcal{P}(\rho_p) = \mathcal{P}(pre_k(\rho))$. Par ailleurs, comme q_2 est correct et appartient à une composante fortement connexe non triviale (resp. CFCT), il existe un suffixe infini ρ' de ρ_p qui ne visite que

les états de cette composante strictement connexe après avoir atteint q_2, et donc ρ' est correcte (resp. $\mathbb{P}(\{\rho' \in \Omega \mid \rho_p \preceq \rho' \wedge \rho' \in \mathsf{C}_\infty\}) \geq \mathbb{P}(C(\rho^*)) > 0$). Cela montre que $\mathcal{A}$ n'est pas surement k-prédictible.

Supposons maintenant que $\mathcal{A}$ n'est pas surement k-prédictible (resp. k-prédictible). Soit $\rho \mathbf{f} q$ une exécution telle que ρ est correcte et il existe ρ_p avec $\rho_p \in \mathcal{P}^{-1}(\mathcal{P}(pre_k(\rho)))$ et $\rho' \in \Omega$ telle que $\rho_p \preceq \rho' \wedge \rho' \in \mathsf{C}_\infty$ (resp. $\mathbb{P}(\{\rho' \in \Omega \mid \rho_p \preceq \rho' \wedge \rho' \in \mathsf{C}_\infty\}) > 0$). Soit q_1 l'état atteint par $pre_k(\rho)$. En particulier, $q_1 \in Q_c$, et $q_1 \overset{\rho_1}{\Longrightarrow} q' \xrightarrow{\mathbf{f}} q$ où $\rho = pre_k(\rho) \cdot \rho_1$ et donc $q' \in Q_c$ et $|\rho_1|_o \leq k$. On note q'_1 l'état atteint par ρ_p.
Pour la sure k-prédictibilité on a : Comme ρ' est infinie, elle finit dans une composante strictement connexe non triviale d'états corrects. On choisit donc pour q_2 le premier état de cette composante strictement connexe atteinte par ρ' depuis q'_1.
Pour la k-prédictibilité on a : On décompose la probabilité $\mathbb{P}(\{\rho' \in \Omega \mid \rho_p \preceq \rho' \wedge \rho' \in \mathsf{C}_\infty\})$ selon la CFCT que les exécutions ρ' atteignent presque surement :

$$\mathbb{P}(\{\rho' \in \Omega \mid \rho_p \preceq \rho' \wedge \rho' \in \mathsf{C}_\infty\}) = \sum_{\mathcal{C} \text{ CFCT accessible depuis } q'_1} \mathbb{P}(\{\rho' \in \Omega \mid \rho_p \preceq \rho' \wedge \rho' \in \mathsf{C}_\infty \wedge \rho' \text{ fini en } \mathcal{C}\}) \ .$$

De $\mathbb{P}(\{\rho' \in \Omega \mid \rho_p \preceq \rho' \wedge \rho' \in \mathsf{C}_\infty\}) > 0$, on déduit qu'il existe une CFCT $\mathcal{C}$ telle que $\mathbb{P}(\{\rho' \in \Omega \mid \rho_p \preceq \rho' \wedge \rho' \in \mathsf{C}_\infty \wedge \rho' \text{ ends in } \mathcal{C}\}) > 0$. Nécessairement, $\mathcal{C} \subseteq Q_c$ et on choisit pour q_2 un des états de $\mathcal{C}$ pour conclure.

□

Preuve du théorème 12.

On établit simultanément des bornes supérieures et inférieures de complexité pour les problèmes de k-sure prédictibilité et k-prédictibilité.

Proposition 24. *Soit $k \in \mathbb{N}$. Decider, étant donné un pLTS $\mathcal{A}$, si $\mathcal{A}$ est surement k-prédictible (resp. k-predictible) est* NLOGSPACE*-difficile.*

Démonstration. On réduit le problème de l'accessibilité dans les graphes orientés acyclique qui est NLOGSPACE-complet [10]. Soit $G = (V, E)$ un graphe orienté acyclique et $s, t \in V$. Comme

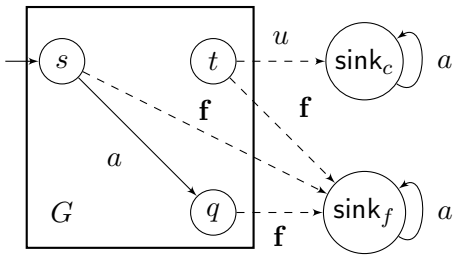

FIGURE 18 – Réduction du problème d'accessibilité.

montré dans la figure 18, on transforme G en un pLTS $\mathcal{A} = \langle Q, q_0, \Sigma, T, \mathbf{P} \rangle$ où :
- $Q = V \uplus \{\mathsf{sink}_f, \mathsf{sink}_c\}$;
- $q_0 = s$
- $\Sigma_o = \{a\}$, $\Sigma_u = \{u, f\}$;
- $T = \{(q, a, q') | (q, q') \in E\} \cup \{(q, f, \mathsf{sink}_f) \mid q \in V\} \cup \{(t, u, \mathsf{sink}_c)\} \cup \{(\mathsf{sink}_f, a, \mathsf{sink}_f)\} \cup \{(\mathsf{sink}_c, a, \mathsf{sink}_c)\}$;
- $\mathbf{P}$ est la matrice spécifiant la probabilité uniforme sur les transitions sortant d'un état.

Cette construction peut être réalisée en LOGSPACE.

Supposons d'abord que t n'est pas accessible depuis s dans G. Alors, de par l'acyclisme du graphe, tout chemin infini finit dans sink_f, ce qui implique $\mathsf{C}_\infty = \emptyset$. Par conséquent $\mathcal{A}$ est k-surement prédictible (et donc k-prédictible) pour tout $k \in \mathbb{N}$.

Supposons maintenant que t est accessible depuis s dans G et soit ρ une exécution fermée correcte atteingant t. Alors $\rho \mathbf{f}\mathsf{sink}_f$ est une exécution fautive et l'exécution infinie correcte $\rho u \mathsf{sink}_c a(\mathsf{sink}_c a)^\omega$ a une probabilité strictement positive. D'où $\mathcal{A}$ n'est pas 0-prédictible. Ainsi, il n'est ni k-prédictible ni surement k-prédictible pour tout $k \in \mathbb{N}$. □

Grâce à la caractérisation du lemme 11, nous sommes maintenant capables de réaliser des procédures efficaces pour les problèmes de décision de la sure prédictibilité et de la prédictibilité.

Proposition 25. *Décider, étant donnés un pLTS $\mathcal{A}$ et $k \in \mathbb{N}$, si $\mathcal{A}$ est k-prédictible (resp. surement k-prédictible) peut être réalisé en* NLOGSPACE.

Démonstration. Nous décrivons un algorithme non-déterministe opérant en espace logarithmique pour décider si $\mathcal{A}$ n'est pas surement k-prédictible, en utilisant la caractérisation du lemme 11. Ceci prouvera que le problème complémentaire, la sure prédictabilité, est aussi dans NLOGSPACE par le théorème d'Immerman-Szelepscényi's (théorème qui sera également utilisé implicitement dans la suite de la preuve).

Premièrement on devine $q_1, q_1', q_2 \in Q_c$, et on vérifie que $q_1' \Rightarrow q_2$, que q_2 appartient à une composante strictement connexe non triviale de $\mathcal{A}$ (resp. une CFCT de $\mathcal{A}$), et que q_1 et q_1' peuvent être atteints par des exécutions ayant même séquence d'observation. Tout ces tests peuvent être fait en NLOGSPACE. Plus précisément, pour le dernier test, on devine au vol une paire de chemin observationnellement équivalent, en se limitant en longueur grâce à un compteur borné par n^2 où n est le nombre d'états de $\mathcal{A}$. Dans le cas positif, on devine une exécution commençant en q_1 qui produit une faute et ayant au plus k actions observables. □

(**fin de la preuve du théorème 12**)

□

A.3 Preuves de la section 4

Preuve de la proposition 18, réalisée avec Hugo Gimbert lors de mon stage de L3.
Commençons par définir formellement la notion d'automate probabiliste.

Définition 18. Un automate probabiliste sur des mots finis $\mathcal{A}$ est défini par un quintuplet $\{Q, \Sigma, M, q_0, F\}$, avec
- Q un ensemble fini d'états ;
- Σ est l'alphabet de travail ;
- $M = (M_a)_{a\in\Sigma}$ où M_a est une matrice de transition : $M_a \in [0,1]^{Q\times Q}$ et pour tout $q \in Q$, $\sum_{r\in Q} M_{q,r} = 1$;
- q_0 l'état initial ;
- F est l'ensemble des états terminaux.

On procède en deux étapes : on réduit d'abord le problème de correspondance de Post au problème de l'égalité pour les automates quasi acycliques, puis on réduit ce problème de l'égalité au problème du vide des automates quasi acycliques.

Première étape :

Une instance du problème de correspondance de Post (PCP en abrégé) est la donnée d'un entier m ≥ 0 et de deux suites $u_0, \ldots, u_m$ et $v_0, \ldots, v_m$ de mots sur un alphabet A. On dit qu'une instance du problème de Post a une solution s'il existe une suite finie $i_1, \ldots, i_n$ d'entiers dans l'intervalle [1,m] telle que $u_{i_1} u_{i_2} \ldots u_{i_n} = v_{i_1} v_{i_2} \ldots v_{i_n}$. Ce problème est connu comme étant indécidable.

Montrons que le problème de l'égalité est indécidable en réduisant le problème de correspondance de Post.
Soit $m \in \mathbb{N}$ et soient f, g deux fonctions de $\mathbb{N}^{\leq m}$ vers $\{1;2\}^*$ représentant une instance de PCP ($\forall i \in \mathbb{N}^{\leq m}, f(i) = u_i$ et $g(i) = v_i$).
Soit h : $\{1;2\}^* \to [\frac{1}{3}; 1]$ définie par $\mathrm{h}(a_1 \ldots a_n) = \frac{a_1}{3^{2n-1}} + \ldots + \frac{a_n}{3^n}$
La fonction h est injective (décomposition en base 3).

On construit l'automate probabiliste quasi acyclique $\mathcal{H}$ de la figure 19 à trois états où $q = 0$ est initial, q_1 est acceptant et q_2 est un état puits.
Pour tout a $\in \Sigma$, $n = |f(a)|$, on choisit la matrice de transition $M_a =$

$$\begin{pmatrix} \frac{1}{3^n} & h(f(a)) & 1 - \frac{1}{3^n} - h(f(a)) \\ 0 & \frac{1}{9^n} & 1 - \frac{1}{9^n} \\ 0 & 0 & 1 \end{pmatrix}$$

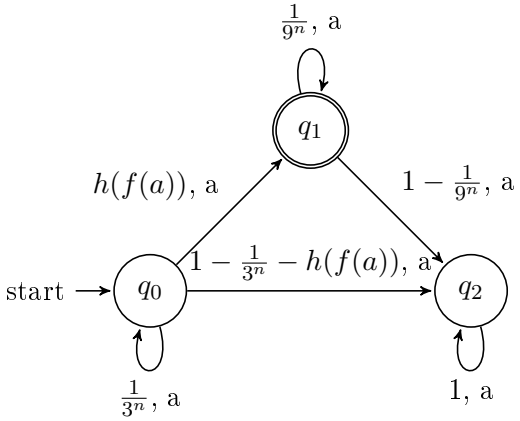

FIGURE 19 – Un automate probabiliste acceptant un mot w avec probabilité $h(f(w))$.

Par récurrence, on obtient $M(a_1 \ldots a_n) =$

$$\begin{pmatrix} \frac{1}{3^m} & h(f(a_1 \ldots a_n)) \\ 0 & \frac{1}{9^m} \end{pmatrix}$$

où $m = |f(a_1 \ldots a_n)|$.

L'automate $\mathcal{H}$ accepte un mot w avec une probabilité $h(f(w))$. On construit l'automate $\mathcal{H}'$ en remplaçant f par g et en échangeant les états acceptants et rejetants. L'automate $\mathcal{H}'$ accepte un mot w avec une probabilité $1 - h(g(w))$.
On construit ensuite l'automate $\mathcal{H}''$ en ajoutant un état supplémentaire initial amenant à l'état initial de $\mathcal{H}$ et $\mathcal{H}'$ avec une probabilité $\frac{1}{2}$ pour chacun en lisant $\sharp$. L'automate $\mathcal{H}''$ accepte le mot $\sharp w$ avec la probabilité $\frac{(1-h(f(w))+h(g(w)))}{2}$.
Donc $\mathcal{H}''$ accepte $\sharp w$ avec la probabilité $\frac{1}{2}$ si et seulement si $h(f(w)) = h(g(w))$ ce qui est équivalent à $f(w) = g(w)$ car h est injective.

Le problème de l'égalité est donc indécidable sur les automates quasi acycliques.

Deuxième étape :
On réduit maintenant le problème de l'égalité au problème du vide. Soit $\mathcal{H}$ un automate et $0 < \lambda < 1$ une valeur de seuil. Nous traitons ici le cas $\lambda = \frac{1}{4}$, mais une construction similaire permet d'obtenir les autres cas.

Soit $\mathcal{H}_1 = \{Q_1, \Sigma, M_1, q_1, F_1\}$ et $\mathcal{H}_2 = \{Q_2, \Sigma, M_2, q_2, F_2\}$ deux automates probabilistes. On construit l'automate $\mathcal{H}_3 = \{Q_3, \Sigma, M_3, q_3, F_3\}$ tel que :

- $Q_3 = Q_1 \times Q_2$.
- $q_3 = (q_1, q_2)$.
- $F_3 = F_1 \times F_2$
- M_3 la matrice de transition tel que si on peut passer de q_1 à q_1' dans $\mathcal{H}_1$ avec une probabilité p en lisant a et de q_2 à q_2' dans $\mathcal{H}_2$ avec une probabilité p' en lisant a, alors on passe de (q_1,q_2) à (q_1', q_2') en lisant a avec une probabilité pp'.

On a alors $\forall w \in \Sigma^*, \mathbb{P}_{\mathcal{H}_3}(w) = \mathbb{P}_{\mathcal{H}_1}(w)\mathbb{P}_{\mathcal{H}_2}(w)$.
En effet, soit $w = a_1...a_n \in \Sigma^*, q \in Q_1, q' \in Q_2$.

$$\begin{aligned}
\mathbb{P}_{\mathcal{H}_3}(q_3, w, (q, q')) &= 1_{(q_1,q_2)}.M^{a_1}_{\mathcal{H}_1\times\mathcal{H}_2}...M^{a_n}_{\mathcal{H}_1\times\mathcal{H}_2}.(1_{(q,q')})^t \\
&= (1_{(q_1)}.M^{a_1}_{\mathcal{H}_1}...M^{a_n}_{\mathcal{H}_1}.(1_q)^t).(1_{(q_2)}.M^{a_1}_{\mathcal{H}_2}...M^{a_n}_{\mathcal{H}_2}.(1_{q'})^t) \\
&= \mathbb{P}_{\mathcal{H}_1}(q_1, w, q).\mathbb{P}_{\mathcal{H}_2}(q_2, w, q')
\end{aligned}$$

On appelle $\mathcal{H}_3$ le produit cartésien de $\mathcal{H}_1$ et de $\mathcal{H}_2$.
Donc en construisant le produit cartésien $\mathcal{G}$ de l'automate $\mathcal{H}$ et de son conjugué, conjugué qui a les mêmes transitions mais échange les états acceptants et non acceptants, on a pour tout mot $w \in \Sigma^*$, $\mathbb{P}_{\mathcal{G}}(w) = (\mathbb{P}_{\mathcal{H}}(w))(1 - \mathbb{P}_{\mathcal{H}}(w))$. Or la fonction $f(x) = x(1-x)$ atteint son maximum $\frac{1}{4}$ en $\frac{1}{2}$. Donc : $\mathbb{P}_{\mathcal{H}}(w) = \frac{1}{2} \Leftrightarrow \mathbb{P}_{\mathcal{G}}(w) \geq \frac{1}{4}$.
Ainsi le problème du vide est indécidable.

□

Preuve de la proposition 19.

On réduit le problème du mot pour les automates probabilistes quasi acyclique qui est indécidable selon la proposition 18.
Soit $\mathcal{A} = \langle Q, \Sigma, q_0, T, P, F\rangle$ un automate probabiliste quasi acyclique et $0 < \varepsilon < 1$ une valeur de seuil. Sans perte de généralité, on suppose que $\mathcal{A}$ est complet. On veut savoir s'il existe un mot dont la probabilité dans $\mathcal{A}$ est supérieure à ε ($\exists w \in \Sigma^*, \mathbb{P}^{\mathcal{A}}(w) \geq \varepsilon$?). Nous construisons le pLTS quasi acyclique $\mathcal{A}' = \langle Q', q_0, \Sigma', T', P'\rangle$ représenté en figure 20 où :

- $Q' = Q \cup \{q_c^\sharp, q_f^\sharp, f^\sharp\}$
- $\Sigma' = \Sigma \cup \{\sharp, f\}$, $\Sigma'_{uo} = \{f\}$
- $T' = T \cup \{(q, \sharp, q_c^\sharp \mid q \in F\} \cup \{(q, \sharp, q_f^\sharp \mid q \in Q \setminus F\} \cup \{q_c^\sharp, \sharp, q_c^\sharp\} \cup \{q_f^\sharp, f, f^\sharp\} \cup \{f^\sharp, \sharp, f^\sharp\}$
- $P'[q, a, q'] = \frac{P[q,a,q']}{1+|\Sigma|}$ si $a \in \Sigma$, $P'[q, f, q'] = 1$ et si $q \in Q$, $P'[q, \sharp, q'] = \frac{1}{1+|\Sigma|}$, sinon, $P'[q, \sharp, q'] = 1$.

Alors $\mathcal{A}'$ n'est pas AA-ε-diagnostiquable si et seulement si $\mathcal{A}$ accepte un mot avec probabilité supérieure à ε. En effet, supposons qu'il existe un mot $w \in \Sigma^*$ tel que $\mathbb{P}^{\mathcal{A}}(w) \geq \varepsilon$. Soit ρ une exécution fautive telle que $\mathcal{P}(\rho) = w\sharp\sharp$. Comme ρ est fautive ρ termine avec $f^\sharp$. Après avoir lu w, ρ était donc dans un état de $Q \setminus F$. Par construction de $\mathcal{A}'$, soit R_f (resp. R_c) l'ensemble des exécutions fautives (resp. correctes) ayant la même séquence d'observation que ρ, $\mathbb{P}(R_f) = \frac{1-P^{\mathcal{A}}(w)}{(1+|\Sigma|)^{|w|}}$ (resp. $\mathbb{P}(R_c) = \frac{P^{\mathcal{A}}(w)}{(1+|\Sigma|)^{|w|}}$). Donc $\mathbb{P}_{amb}(\rho) = \mathbb{P}^{\mathcal{A}}(w) \geq \varepsilon$. Comme les exécutions correctes et fautives vont continuer en lisant des $\sharp$ avec probabilité 1, toute extension d'une exécution de R_f est fautive, ambigüe et a probabilité supérieure à ε. Donc $\mathcal{A}'$ n'est pas AA-ε-diagnostiquable.

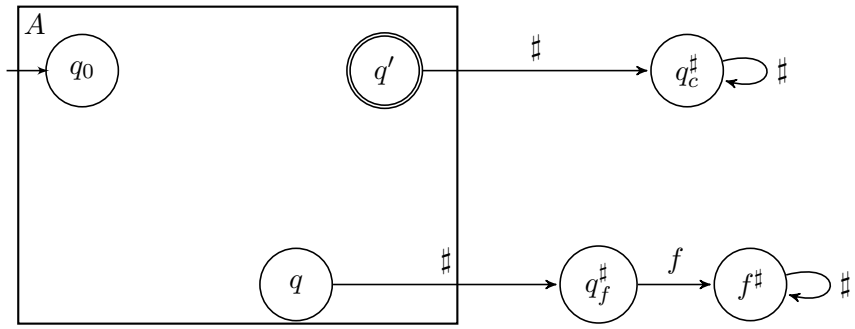

FIGURE 20 – Réduction du problème du vide.

Inversement supposons que $\mathcal{A}'$ n'est pas AA-ε-diagnostiquable. Il existe donc une exécution fautive ρ dont la probabilité d'ambiguité est supérieure à ε'. $\mathcal{P}(\rho) = w\sharp\sharp^+$ avec $w \in \Sigma^*$ car ρ démarre q_0 et termine en $f^\sharp$. Puisqu'après avoir lu un $\sharp$, les exécutions deviennent déterministes et de par notre choix de probabilité, $\mathbb{P}(\{\rho q \in \mathsf{SR}_{|w|} \mid q \in F, \mathcal{P}(\rho q) = w\}) \geq \frac{\varepsilon}{(1+|\Sigma|)^{|w|}}$. Ceci implique que $\mathbb{P}^{\mathcal{A}}(w) \geq \varepsilon$.

□

B Une procédure PTIME erronée pour la A-diagnostiquabilité.

L'algorithme de [4] est une adaptation de celui de [9] qui décide en PTIME la diagnostiquabilité des systèmes à événements discrets déterministes. Dans [4], la comportement correct du système est spécifié par un automate déterministe et une faute consiste à produire un mot n'appartenant pas au langage de cet automate. Une fois effectué le produit synchronisé entre le système et la spécification, ce cadre d'étude se ramène au notre. Nous décrivons donc comment la procédure donnée en [4] vérifie la A-diagnostiquabilité du pLTS $\mathcal{A}$ dans notre cadre d'étude. Tout d'abord, on construit le sous-pLTS (ce qui signifie que la somme des probabilités des transitions sortantes des état peut être inférieure à 1) testeur tel que :

- Un état est une paire (q_1, q_2) où $q_1 \in Q$ et $q_2 \in Q_c$. L'état initial est (q_0, q_0) où q_0 est l'état initial de $\mathcal{A}$.
- Il y a une transition étiquetée par a de (q_1, q_2) vers (q_1', q_2') avec probabilité $p_1 p_2 > 0$ si :
  1. la somme des probabilités des exécutions fermées de q_1 vers q_1' et associées à l'action observable a est égal à p_1 ;
  2. en supposant que l'ensemble des exécutions correctes commençant en q_2 et associées à l'action observable a est non vide, alors la somme des probabilités des exécutions fermées correctes de q_2 à q_2' et associées à a conditionné sur l'ensemble des exécutions fermées commençant en q_2 et associées à a, est égal à p_2.

Alors $\mathcal{A}$ est A-diagnostiquable s'il n'existe aucune composante récurrente (i.e. une CFCT telle que pour tout état la somme des probabilités des transitions sortantes vaut 1) dans le sous-pLTS testeur dont la première composante des états appartienne à Q_f.

Nous avons représenté en figure 21 le sous-pLTS testeur associé au pLTS A-diagnostiquable de la figure 2. De q_0 en observant a, on atteint q_1, f_1 ou f_2. Par conséquent, de (q_0, q_0) nous avons trois trnasitions sortantes dont la première composante est q_1, f_1 or f_2 et la seconde est q_1, le seul état non fautif accessible. De q_1, on peut rester en q_1 en observant un a, ce qui implique une boucle sur (q_1, q_1). De f_1 en lisant a soit on reste sur place, soit on arrive en f_2 ce qui implique une boucle sur (f_1, q_1) et une transition vers (f_2, q_1). De f_2, seul b est observable et b ne peut être observé depuis q_1. Donc (f_2, q_1) n'a pas de transition sortante. Dans ce sous-pLTS, la seule composante récurrente est $\{(q_1, q_1)\}$ dont la première composante appartient à Q_c. Ainsi, l'algorithme répond

correctement que le pLTS est A-diagnostiquable.

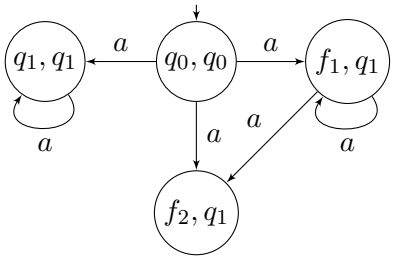

FIGURE 21 – Le sous-pLTS testeur du pLTS de la figure 2.

Observons le pLTS de la figure 22. Il y a une unique exécution fautive $q_0\mathbf{f}(f_1a)^\omega$ et elle a une probabilité positive. Cette exécution est rendue ambigüe par l'exécution correcte $q_0u(q_1a)^\omega$, donc ce pLTS n'est pas A-diagnostiquable.

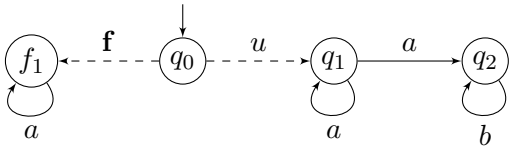

FIGURE 22 – Un pLTS qui n'est pas A-diagnostiquable.

Le sous-pLTS testeur du pLTS de la figure 22 est représenté dans la figure 23. La seule composante récurrente est $\{(q_2, q_2)\}$. Donc l'algorithme répond incorrectement que le pLTS est A-diagnostiquable.

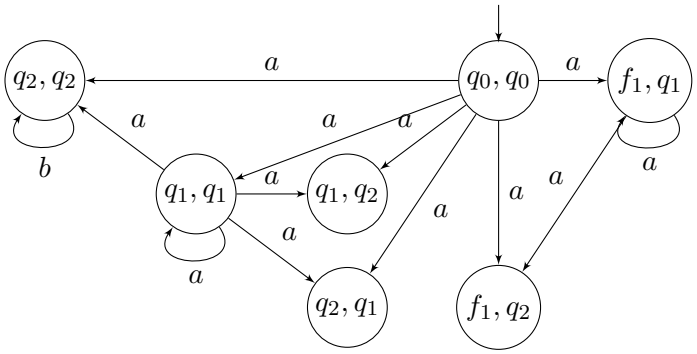

FIGURE 23 – Le sous-pLTS testeur du pLTS de la figure 22.

Puisque la preuve de la correction de l'algorithme est fournie dans [4], montrons où se situe l'erreur. Celle-ci est située dans la preuve de la condition suffisante. L'équation (2), page 973, affirme qu'étant donnée une exécution fautive s la probabilité d'une exécution prolongeant s dans le pLTS, t telle que $\mathcal{P}(st)$ est ambigüe est inférieure ou égale à la somme des probabilité des exécutions $(\mathcal{P}(t), \mathcal{P}(t'))$ prolongeant une exécution $(\mathcal{P}(s), \mathcal{P}(s'))$ dans le sous-pLTS testeur. Maintenant considérons l'exécution $s = q_0\mathbf{f}f_1af_1$ et la prolongation $t = f_1af_1af_1$. Sa probabilité est égale à 1. Les exécutions correspondantes dans le sous-pLTS sont $(f_1, q_1)a(f_1, q_1)a(f_1, q_2)$ et $(f_1, q_1)a(f_1, q_1)a(f_1, q_1)$ dont les probabilités somment à $\frac{1}{2} = \frac{1}{4} + \frac{1}{4}$. Donc l'équation (2) est fausse.

C Une procédure PTIME erronée pour la AA-diagnostiquabilité.

Dans [4] une procédure PTIME est également fournie pour traiter une autre forme de diagnostiquabilité : la AA-diagnostiquabilité. Malheureusement, cet algorithme est également faux.

Voici comment la AA-diagnostiquabilité est définie dans [4] :

Définition 19. Un pLTS $\mathcal{A}$ est AA-diagnostiquable si et seulement si pour tout $\varepsilon > 0, \tau > 0$ il existe $n_{\varepsilon,\tau} \in \mathbb{N}$ tel que pour toute exécution fautive ρ et tout $n \geq n_{\varepsilon,\tau}$ on a :

$$\frac{\mathbb{P}(\{\rho' \in \mathsf{SR}_{n+|\rho|_o} \mid \rho \preceq \rho' \wedge \mathbb{P}_{amb}(\mathcal{P}(\rho')) > \varepsilon\})}{\mathbb{P}(\{\rho' \in \mathsf{SR}_{n+|\rho|_o} \mid \rho \preceq \rho'\})} < \tau$$

$$\text{où } \mathbb{P}_{amb}(\sigma) = \frac{\mathbb{P}(\{\rho' \in \mathsf{SR}_{|\sigma|} \mid \mathcal{P}(\rho') = \sigma \wedge \rho' \in \mathsf{C}_n\})}{\mathbb{P}(\{\rho' \in \mathsf{SR}_{|\sigma|} \mid \mathcal{P}(\rho') = \sigma\})}$$

Nous introduisons maintenant une notion d'équivalence entre les pLTS utilisée dans l'algorithme de [4].

Définition 20. Soit $\mathcal{A}_i = \langle Q_i, \pi_{0,i}, \Sigma_o, T_i, \mathbf{P}_i \rangle$ pour $i \in \{1, 2\}$ deux pLTS (où $\pi_{0,i}$ décrit la distribution initiale). Alors $\mathcal{A}_1$ et $\mathcal{A}_2$ sont p-équivalents si pour tout mot $w \in \Sigma_o^*$,

$$\mathbb{P}_1(\rho \mid \sigma_\rho = w) = \mathbb{P}_2(\rho \mid \sigma_\rho = w)$$

où $\mathbb{P}_i$ est la probabilité induite par le pLTS $\mathcal{A}_i$.

La p-équivalence de deux pLTS peut être vérifiée en temps polynomial en adaptant une technique classique pour les automates probabilistes [5].

Décrivons comment l'algorithme proposé dans [4] vérifie la AA-diagnostiquabilité. Tout d'abord, il construit le sous-pLTS testeur utilisé dans l'algorithme vérifiant la A-diagnostiquabilité. Il enrichit ensuite les étiquettes de chaque transition en ajoutant une seconde probabilité définie par : en supposant qu'il existe une transition étiquetée par a de (q_1, q_2) à (q_1', q_2'), la seconde probabilité est $p_1 p_2$ où :

1. la somme des probabilitées des exécutions fermées de q_2 à q_2' étiquetées par a est égale à p_2 ;
2. la somme des probabilités des exécutions fermées de q_1 à q_1' et étiquetées par a, conditionné sur l'ensemble des exécutions fermées commençant en q_1 et associé à l'action observable a, est égale à p_1.

Ensuite, on observe les CFCT pour lesquelles la somme des probabilités des transitions sortantes de chaque état vaut 1 quelle que soit la composante de la paire de probabilité choisie (de telles CFCT sont dites *bi-fermées*) et pour lesquelles la première composante de l'état est fautif. Pour chacune de ces CFCT $\mathcal{C}$ on construit le pLTS $\mathcal{A}_1$ (resp. $\mathcal{A}_2$) ayant les états et les transitions de $\mathcal{C}$ et où la probabilité des transitions est obtenue en considérant la première (resp. la seconde) probabilité de l'étiquette. Finalement la procédure vérifie si $\mathcal{A}_1$ et $\mathcal{A}_2$, en prenant la distribution stationnaire pour distribution initiale, sont p-équivalents. Si au moins une CFCT donne cette p-équivalence, alors le pLTS n'est pas AA-diagnostiquable.

La premier problème vient de la définition donnée. Regardons le pLTS de la figure 24. Il s'agit d'un exemple tiré de [4] où il est annoncé étant AA-diagnostiquable. La figure 25 représente le graphe que l'algorithme associe à ce pLTS. La seule CFCT bi-fermée est (f_1, q_1) et elle n'est pas p-équivalente. Donc le test réalisé par l'algorithme affirme que le pLTS est AA-diagnostiquable.

Malheureusement, cette affirmation est fausse.

Proposition 26. *Le pLTS de la figure 24 n'est pas AA-diagnostiquable selon la définition 19.*

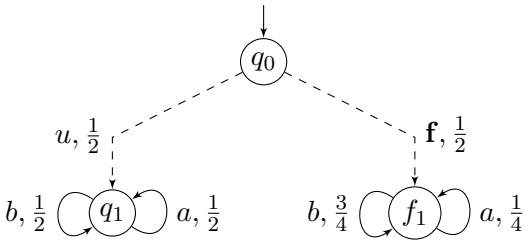

FIGURE 24 – Un pLTS qui n'est pas AA-diagnostiquable selon la définition 19 mais l'est selon la définition 21.

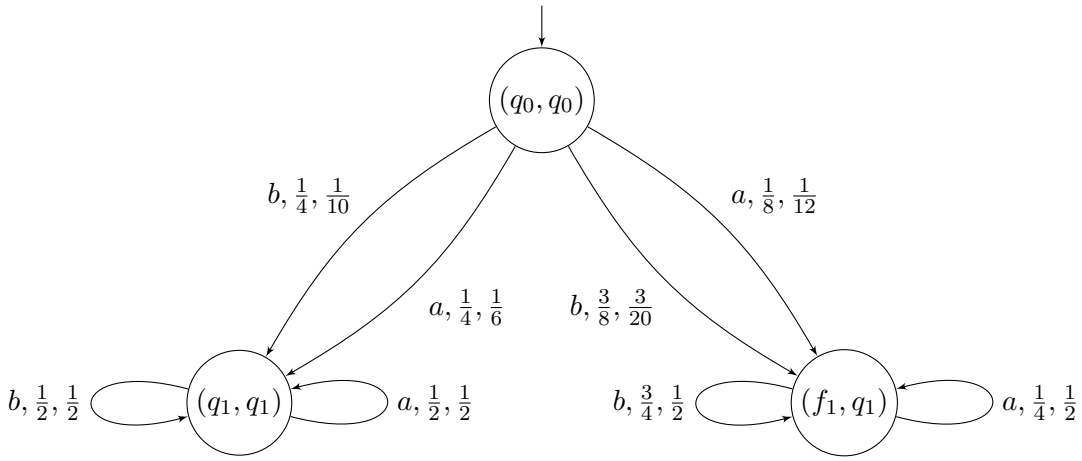

FIGURE 25 – Construction de Chen et Kumar pour le pLTS de la figure 24

Démonstration. Supposons que $\mathcal{A}$, le pLTS de la figure 24, est AA-diagnostiquable. Alors soit $\varepsilon > 0, 1 > \tau > 0$, il existe $n_0 \in \mathbb{N}$ tel que pour toute exécution fautive ρ et tout $n \geq n_0$ on a :

$$\frac{\mathbb{P}(\{\rho' \in \mathsf{SR}_{n+|\rho|_o} \mid \rho \preceq \rho' \wedge \mathbb{P}_{amb}(\mathcal{P}(\rho')) > \varepsilon\})}{\mathbb{P}(\{\rho' \in \mathsf{SR}_{n+|\rho|_o} \mid \rho \preceq \rho'\})} < \tau.$$

Soit ρ l'exécution fautive $q_0\mathbf{f}(f_1 a)^{n_1} f_1$ avec $n_1 > \log_2(\frac{\varepsilon}{1-\varepsilon}) + n_0(\log_2(3) - 1)$.
Alors pour $\rho' \in \mathsf{SR}_{n_0+|\rho|_o}$ telle que $\rho \preceq \rho'$ on a :

$$\mathbb{P}_{amb}(\mathcal{P}(\rho')) \geq \frac{\frac{1}{2^{n_0+n_1}}}{\frac{1}{2^{n_0+n_1}} + \frac{1}{4^{n_1}}(\frac{3}{4})^{n_0}} = \frac{1}{1 + \frac{3^{n_0}}{2^{n_0+n_1}}} > \varepsilon.$$

Donc, $\frac{\mathbb{P}(\{\rho' \in \mathsf{SR}_{n+|\rho|_o} | \rho \preceq \rho' \wedge \mathbb{P}_{amb}(\mathcal{P}(\rho')) > \varepsilon\})}{\mathbb{P}(\{\rho' \in \mathsf{SR}_{n+|\rho|_o} | \rho \preceq \rho'\})} = 1 < \tau$ ce qui contredit $\tau < 1$.
Par conséquent $\mathcal{A}$ n'est pas AA-diagnostiquable. □

Comme vu dans la preuve de la proposition précédente, la condition que l'entier $n_{\varepsilon,\tau}$ ne dépend pas de l'exécution fautive considérée pourrait expliquer le problème de l'algorithme. Aussi introduisons une définition alternative de la AA-diagnostiquabilité.

Définition 21. Un pLTS $\mathcal{A}$ est AA-diagnostiquable si et seulement si pour tout $\varepsilon > 0$ et toute exécution fautive ρ on a :

$$\lim_{n\to\infty} \frac{\mathbb{P}(\{\rho' \in \mathsf{SR}_{n+|\rho|_o} \mid \rho \preceq \rho' \wedge \mathbb{P}_{amb}(\mathcal{P}(\rho')) > \varepsilon\})}{\mathbb{P}(\{\rho' \in \mathsf{SR}_{n+|\rho|_o} \mid \rho \preceq \rho'\})} = 0$$

$$\text{où } \mathbb{P}_{amb}(\sigma) = \frac{\mathbb{P}(\{\rho' \in \mathsf{SR}_{|\sigma|} \mid \mathcal{P}(\rho') = \sigma \wedge \rho' \in \mathsf{C}_n\})}{\mathbb{P}(\{\rho' \in \mathsf{SR}_{|\sigma|} \mid \mathcal{P}(\rho') = \sigma\})}$$

Proposition 27. *Le pLTS de la figure 24 est AA-diagnostiquable selon la définition 21.*

Démonstration. Soit $\varepsilon > 0, \tau > 0$.

Prenons $\lambda > 0$ avec $\lambda < \frac{3}{4} - \ln(2)$.
De par notre choix de λ, il existe $n_0 \in \mathbb{N}$ tel que $\forall n \geq n_0, \frac{1}{1+\frac{3^{n(\frac{3}{4}-\lambda)}}{2^n}} < \varepsilon$.

Considérons une séquence d'observation σ avec $|\sigma| = n \geq n_0$ telle que $|\frac{|\sigma|_a}{n} - \frac{1}{4}| < \lambda$. Il y a une unique exécution fermée correcte et une unique exécution fermée fautive correspondant à σ. Donc :

$$\mathbb{P}_{amb}(\sigma) = \frac{\frac{1}{2^n}}{\frac{1}{2^n} + \frac{3^{n-|\sigma|_a}}{4^n}} \leq \frac{1}{1+\frac{3^{n(\frac{3}{4}-\lambda)}}{2^n}} < \varepsilon$$

soit $\rho = q_0 \mathbf{f} f_1 x_1 f_1 \dots x_k f_1$ une exécution fautive quelconque où $x_i \in \{a, b\}$.
Selon la loi (faible) des grands nombres, comme la probabilité d'apparition d'un a dans f_1 est égale à $\frac{1}{4}$:

$$\exists n_1 \in \mathbb{N}, \forall n \geq n_1, \frac{\mathbb{P}(\{\rho' \in \mathsf{SR}_n \mid \rho \preceq \rho' \wedge \left|\frac{|\rho'|_a}{n} - \frac{1}{4}\right| \geq \lambda\})}{\mathbb{P}(\{\rho' \in \mathsf{SR}_n \mid \rho \preceq \rho'\})} < \tau$$

(remarquons que n_1 depend de ρ).

Soit $m = \max(n_1, n_0)$. En combinant les résultats précédents pour tout $n \geq m$,

$$\frac{\mathbb{P}(\{\rho' \in \mathsf{SR}_n \mid \rho \preceq \rho' \wedge \mathbb{P}_{amb}(\mathcal{P}(\rho')) \geq \varepsilon\})}{\mathbb{P}(\{\rho' \in \mathsf{SR}_n \mid \rho \preceq \rho'\})} < \tau$$

Donc le pLTS n'est pas AA-diagnostiquable. □

Malheureusement, même avec la définition 21, l'algorithme de [4] est erroné. Observons le pLTS de la figure 26. La construction associée, représentée dans la figure 27, contient une unique CFCT bi-fermée à considérer : $\{(f_1, q_1), (f_2, q_2)\}$. Quelle que soit la composante considérée, la distribution stationnaire de cette CFCT est équidistribuée. On peut donc échanger (f_1, q_1) et (f_2, q_2) dans une des composantes et on observe alors que les deux pLTS sont identiques et donc p-équivalents. Donc l'algorithme de [4] renvoie que le pLTS n'est pas AA-diagnostiquable. La proposition suivante établit que ce n'est pas le cas pour la définition 21.

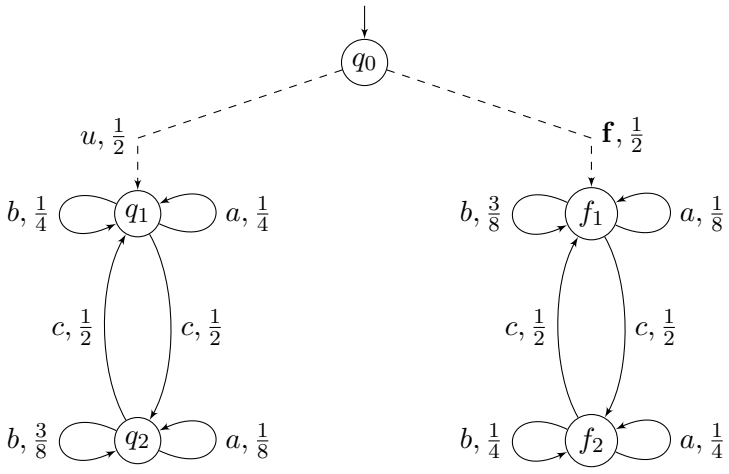

FIGURE 26 – Un autre pLTS AA-diagnostiquable pour la définition 21.

Proposition 28. *Le pLTS de la figure 24 est AA-diagnostiquable selon la définition 21.*

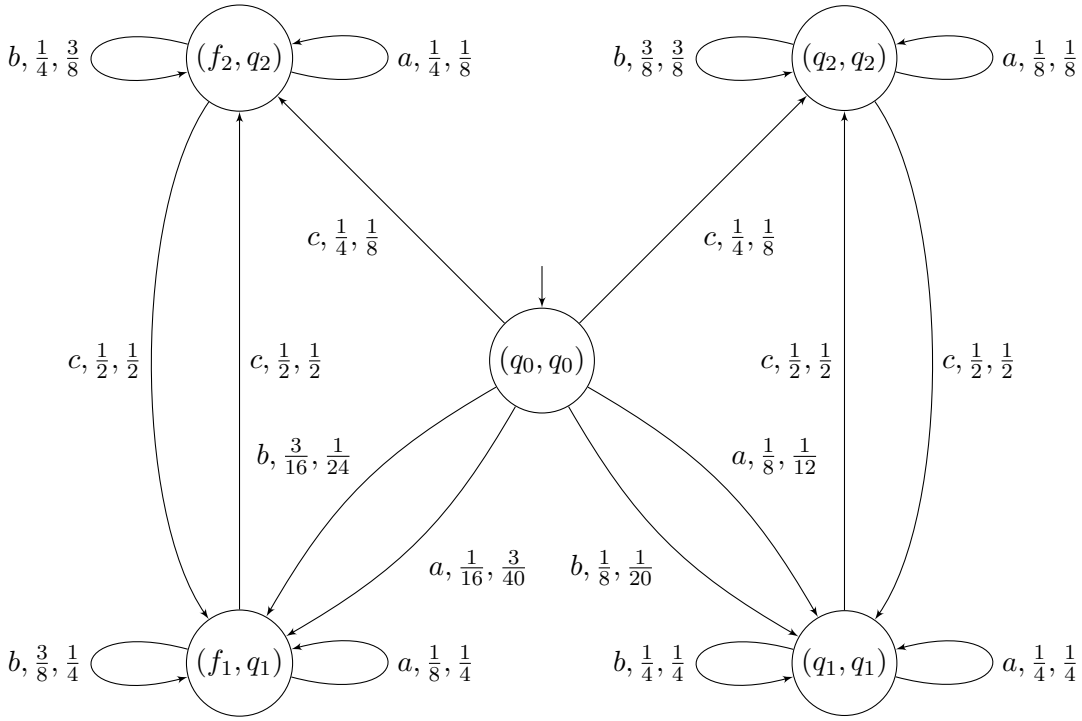

FIGURE 27 – Construction de Chen et Kumar pour le pLTS de la figure 26.

Démonstration. Intuitivement, le pLTS de la figure 26 est AA-diagnostiquable car dans une exécution fautive, selon la parité du nombre de c observés, la probabilité d'apparition d'un b est soit plus grande que celle d'apparition d'un a soit égale.

Soit $\varepsilon > 0, \tau > 0$. Prenons $\lambda > 0$ avec $\lambda < \frac{3}{4} - \ln(2)$.
De par notre choix de λ, il existe $n_0 \in \mathbb{N}$ tel que $\forall n \geq n_0, \frac{1}{1+\frac{3^{n(\frac{3}{4}-\lambda)}}{2^n}} < \varepsilon$.

On définit inductivement pour $x \in \{a, b\}$, la fonction $even_x$ de Σ_o^* à $\mathbb{N}$ par :
- $even_x(\varepsilon) = 0$;
- Si $|\sigma|_c$ est pair alors $even_x(\sigma x) = even_x(\sigma) + 1$ et $even_x(\sigma y) = even_x(\sigma)$ pour $y \neq x$;
- Si $|\sigma|_c$ est impair alors $even_x(\sigma y) = even_x(\sigma)$ pour tout y.

Considérons la séquence d'observation σ avec $even_a(\sigma) + even_b(\sigma) = n \geq n_0$ telle que $|\frac{even_a(\sigma)}{n} - \frac{1}{4}| < \lambda$. Il y a une unique exécution fermée correcte et une unique exécution fermée fautive associée à σ. Donc :

$$\mathbb{P}_{amb}(\sigma) = \frac{\frac{1}{2^n}}{\frac{1}{2^n} + \frac{3^{n-even_a(\sigma)}}{4^n}} \leq \frac{1}{1 + \frac{3^{n(\frac{3}{4}-\lambda)}}{2^n}} < \varepsilon.$$

Soit ρ une exécution fautive quelconque. Considérons une variable aléatoire S_n de distribution binomiale $B(n, \frac{1}{4})$. Selon la loi faible des grands nombres :

$$\exists n_1 \in \mathbb{N}, \forall n \geq n_1, \mathbb{P}(\left| \frac{S_n + even_a(\rho)}{n + even_a(\rho) + even_b(\rho)} - \frac{1}{4} \right| \geq \lambda) < \frac{\tau}{2}.$$

Dans le pLTS, $\mathbb{P}(\{\rho' \in \Omega \mid even_a(\mathcal{P}(\rho')) + even_b(\mathcal{P}(\rho')) = \infty\}) = 1$. Ainsi, il existe n_2 tel que pour $n \geq n_2$:

$$\frac{\mathbb{P}(\{\rho' \in \mathsf{SR}_n \mid \rho \preceq \rho' \wedge even_a(\mathcal{P}(\rho')) + even_b(\mathcal{P}(\rho')) < \max(n_0, n_1)\})}{\mathbb{P}(\{\rho' \in \mathsf{SR}_n \mid \rho \preceq \rho'\})} < \frac{\tau}{2}.$$

En combinant les résultats précédents, pour tout $n \geq n_2$,

$$\frac{\mathbb{P}(\{\rho' \in \mathsf{SR}_n \mid \rho \preceq \rho' \wedge \mathbb{P}_{amb}(\mathcal{P}(\rho')) \geq \varepsilon\})}{\mathbb{P}(\{\rho' \in \mathsf{SR}_n \mid \rho \preceq \rho'\})} \leq$$

$$\frac{\mathbb{P}(\{\rho' \in \mathsf{SR}_n \mid \rho \preceq \rho' \wedge even_a(\mathcal{P}(\rho')) + even_b(\mathcal{P}(\rho')) < \max(n_0, n_1)\})}{\mathbb{P}(\{\rho' \in \mathsf{SR}_n \mid \rho \preceq \rho'\})} +$$

$$\frac{\sum_{x \geq \max(n_0,n_1)} \mathbb{P}(\{\rho' \in \mathsf{SR}_n \mid \rho \preceq \rho' \wedge even_a(\mathcal{P}(\rho')) + even_b(\mathcal{P}(\rho')) = x \wedge \mathbb{P}_{amb}(\mathcal{P}(\rho')) \geq \varepsilon\})}{\mathbb{P}(\{\rho' \in \mathsf{SR}_n \mid \rho \preceq \rho'\})}$$

$$< \frac{\tau}{2} + \frac{\sum_{x \geq \max(n_0,n_1)} \frac{\tau}{2} \mathbb{P}(\{\rho' \in \mathsf{SR}_n \mid \rho \preceq \rho' \wedge even_a(\mathcal{P}(\rho')) + even_b(\mathcal{P}(\rho')) = x\})}{\mathbb{P}(\{\rho' \in \mathsf{SR}_n \mid \rho \preceq \rho'\})}.$$

$$< \tau$$

Donc le pLTS est AA-diagnostiquable.

□