Diagnostic, Pronostic et Opacité

Formation Systèmes à Evénements Discrets

1ère édition
Janvier 2024

Société d'Automatique,
de Génie Industriel & de Productique

Plan


UN TRAIN PEUT
EN CACHER
UN AUTRE

Plan

1. Introduction
    - Entraves à la sûreté de fonctionnement
    - Moyens pour la tolérance aux fautes
2. Diagnostic
3. Pronostic
4. Opacité
5. Conclusion

Entraves à la sûreté de fonctionnement

Taxonomie de la sûreté de fonctionnement

- Notions informelles
  - Défaut
  - Dysfonctionnement : Signifie que le comportement d'une entité ne correspond plus au comportement souhaité.
- Notions formelles (Laprie, 1995) :
  - Défaillance
  - **Faute**
  - Erreur
  - Panne

Défaillance (failure)

- **Définition 1 (Zwingelstein, 1995)** : L'altération ou la cessation de l'aptitude d'un ensemble à accomplir sa ou ses fonctions requise(s) avec les performances définies dans les spécifications techniques.
- Conséquences
  - Notion de défaillance complète : Une entité est défaillante si ses capacités fonctionnelles sont interrompues (arrêt du fonctionnement de l'entité)
  - Notion de défaillance partielle : Il n'y a pas perte totale de la fonction de l'entité mais sa performance passe en dessous d'un seuil défini
- **Exemple 1** : Considérons un aiguillage permettant d'orienter un train en position normal ou en position reverse (fonction aiguillage).
  - Si l'aiguillage reste bloqué dans la position normale, il y a une défaillance complète de sa fonction d'aiguillage
  - Si le positionnement de l'aiguillage doit se faire en moins de 15s et qu'il se positionne en 16s, il y a une défaillance partielle de la fonction de l'aiguillage

Panne

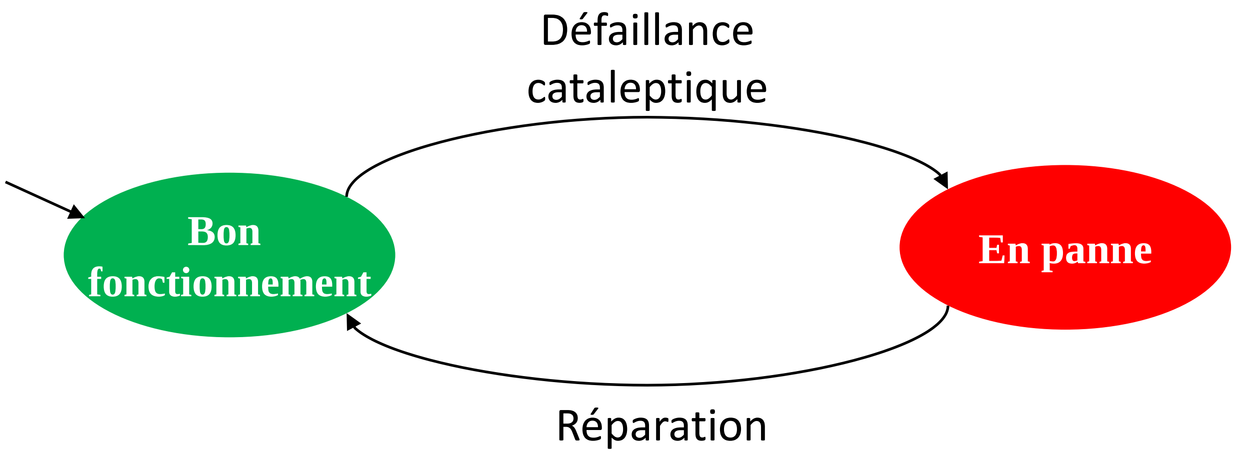
- **Définition 2** : La panne est l'inaptitude d'une entité à accomplir une mission. Une panne résulte toujours d'une défaillance

- Conséquences :
  - On confond souvent la notion de panne et la notion de défaillance
    - Une défaillance est un évènement
  - Il vaut mieux parler d’ « état de panne »
    - Une entité est dans un état de panne suite à une défaillance complète

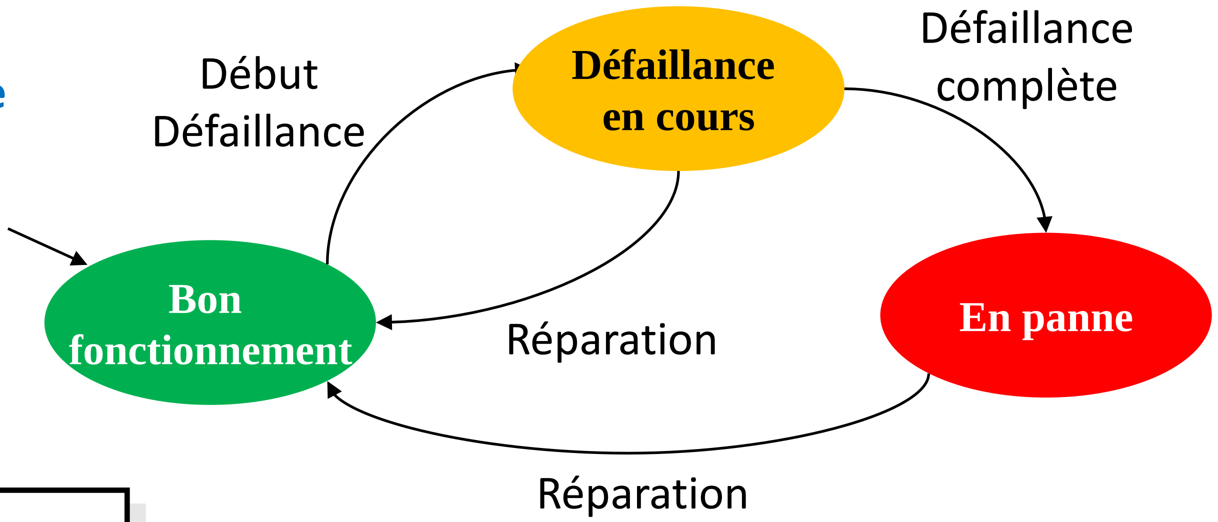
Défaillance vs Panne

- Classification des défaillances en fonction de la rapidité :
  - Défaillance progressive :
    - prévisible par un examen antérieur
  - Défaillance soudaine
- Classification des défaillances en fonction de l'amplitude :
  - Défaillance partielle
  - Défaillance complète

Défaillance vs Panne



Etats résultants d'une défaillance cataleptique



Etats résultants d'une défaillance par dégradation

Défaillance cataleptique : soudaine et complète
Défaillance par dégradation : progressive et partielle

Erreur - Faute

- **Définition 3** (Laprie 95) : Une erreur est susceptible de provoquer ou non une Défaillance, fonction de la redondance, de l'activité de l'entité, de la définition de la défaillance.
- **Définition 4** (Laprie 95) : La cause adjugée ou supposée d'une erreur est une faute.

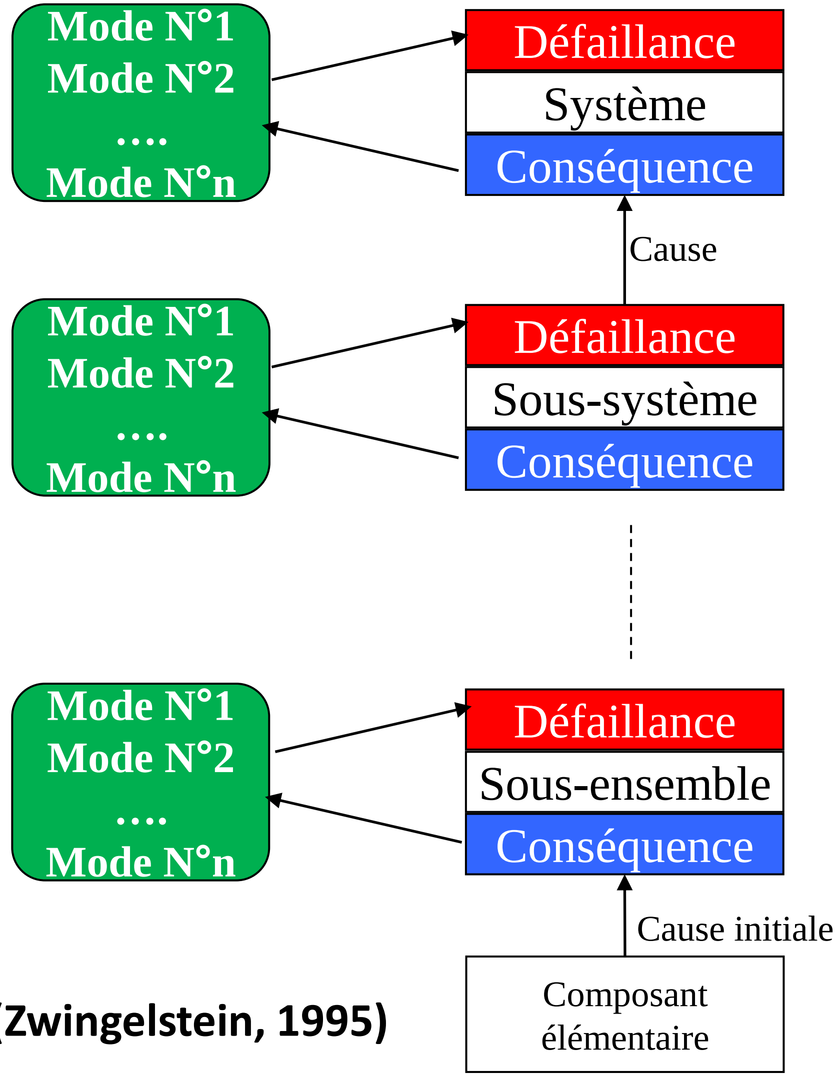
- Conséquences : Une faute est la cause d'une erreur

- Causalité des entraves à la sûreté de fonctionnement
  - **... -> défaillance -> faute -> erreur -> défaillance -> ...**

Défaillance - faute - erreur

- **La fonction d'un passage à niveau est d'empêcher les voitures de traverser les voies lorsqu'un train est en approche**

- **Exemple 2** :
  - 1) La non-fermeture des barrières d'un passage à niveau à l'approche d'un train est une faute
  - 2) La traversée des voitures en dépit de signal du passage à niveau est une erreur.
  - 3) La collision du train avec une voiture est une défaillance.



(Zwingelstein, 1995)

Notion de faute en SED

- Une faute est un évènement
  - Faute **permanente** : c'est que l'évènement est persistant jusqu'à ce qu'il y ait réparation de l'entité;
  - Faute **intermittente** : l'évènement de faute disparait sans qu'il y ait eu de réparation.(Boussif et al., 2020)
- En SED en considère deux catégories d'évènements :
  - Les évènements **observables** (seuls évènements considérés pour faire du contrôle)
  - Les évènements **non-observables** (évènements non détectés par un capteur)
- **Hypothèses générales pour le traitement des fautes en SED**
  - H1 : Les fautes sont des évènements non-observables (cas non trivial)
  - H2 : Le système continue à produire des évènements (le système n'est pas en panne)

Moyens pour la tolérance aux fautes

Diagnostic

- **Etymologie** : Les termes grecques : **dia « à travers** » et **gnosis « connaissance »**

Une discipline déterminant si le comportement d'un équipement est conforme à ses spécifications.

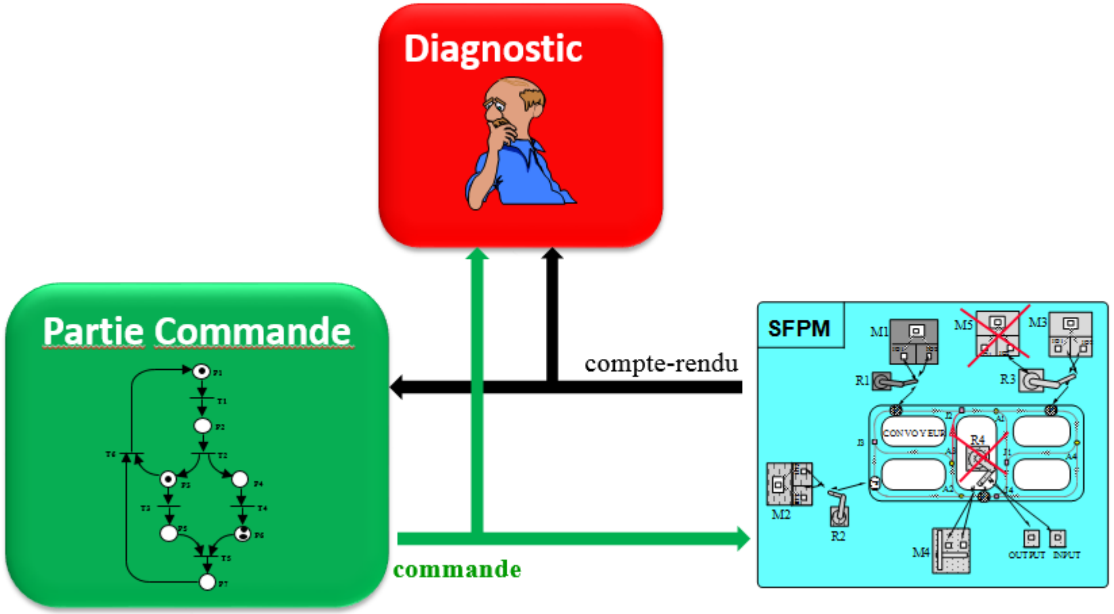
- **Définition 5 (industrielle parmi tant d'autres)** : Exploitation de toute la connaissance accessible existante sur le système afin de détecter, localiser et identifier le **comportement** (normal ou anormal).

FDI (Fault Detection and Isolation) : Terminologie des automaticiens du continu

- Détection : Détecter tout écart du comportement normal du système
- Isolation : Remonter à l'origine et localiser le composant défectueux
- Diagnosis : Déterminer l'instant d'apparition de la faute, sa durée et sa sévérité

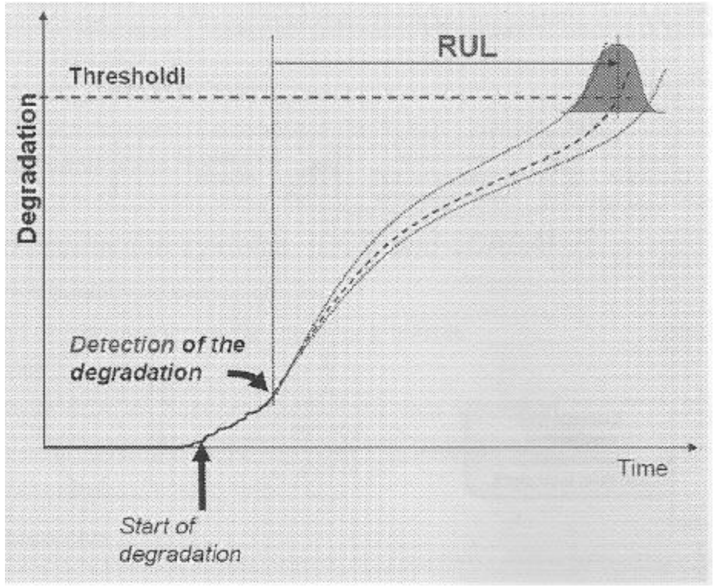
Diagnostic

- **Définition 6 (définition informelle)** : Déterminer en observant un système si une faute a altéré son fonctionnement
  - A partir de l'observation des évènements observables, arriver à dire si une faute (évènement non observable) est apparue dans le système.
- Remarque : Le Diagnostic des SED est-il de la détection ou de l'Isolation ?



Pronostic

- **Définition 6 en automatique continue** : Prédire la durée résiduelle de fonctionnement jusqu'à l'état de panne (**Remaining Useful Life – RUL**) d'un système, connaissant son état courant et ses conditions opérationnelles futures.



- **Définition 7 en SED** : A partir de l'observation d'évènements, prédire qu'une **faute** (évènement non observable) va entraver de manière certaine le comportement futur du système.

Opacité

- **Définition 8 (Saboori et Hadjicostis 2007)** : Capacité d'un système à garder secrètes certaines informations vis-à-vis des observateurs.

- Vise à limiter les inférences qu'un intrus passif peut faire concernant le comportement du système.

- Etroitement liée à la préservation de la **confidentialité** et à la **sécurité (security)** des systèmes.

Plan

1. Introduction
2. Diagnostic
    - Diagnostiqueur
    - Diagnosticabililité
    - Vérification de la diagnosticabilité
3. Pronostic
4. Opacité
5. Conclusion

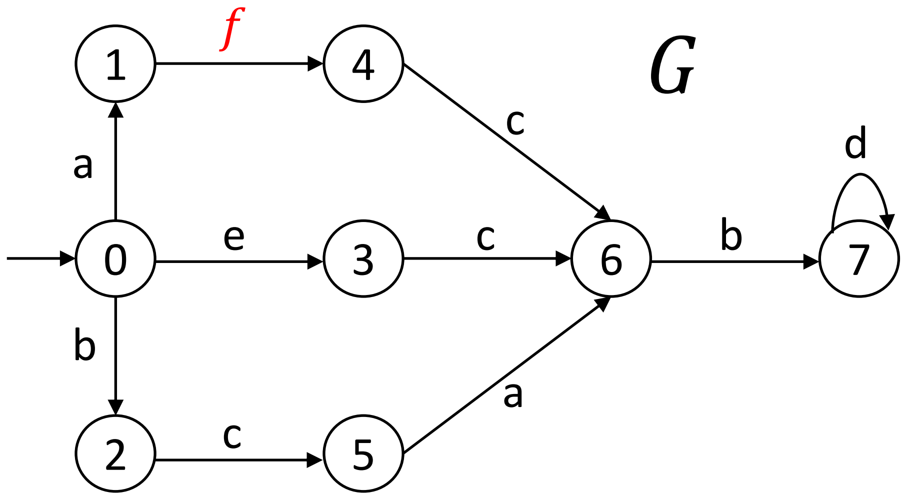
Diagnostic

Automates à états finis pour le diagnostic

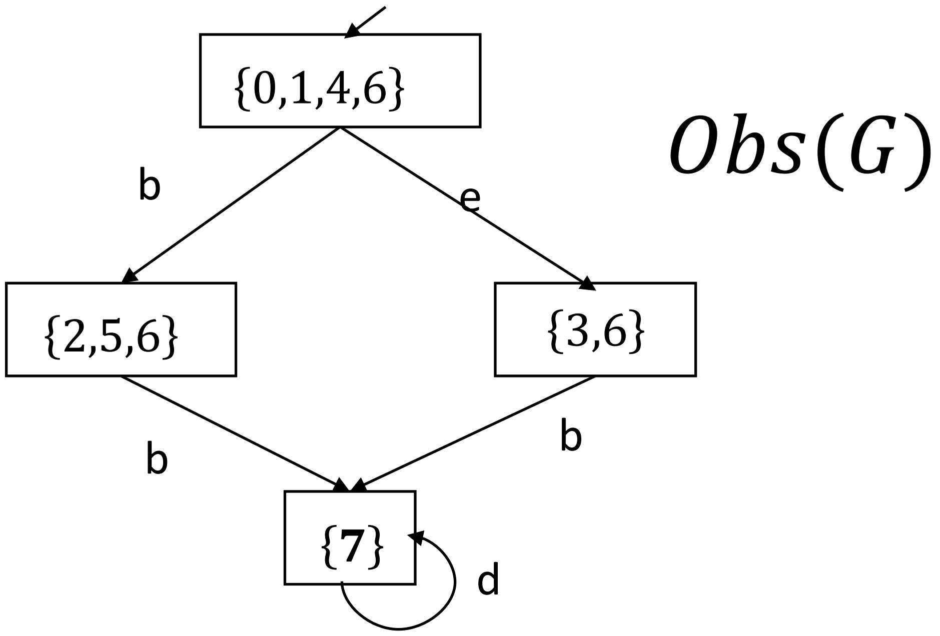
- G est un automate à état finis, $G = < X, \Sigma, \delta, x_0 >$ avec :
  - X : ensemble fini d'états;
  - Σ : ensemble fini d'événements ; $\boldsymbol{\Sigma = \Sigma_o \cup \Sigma_{uo}}$
    - Σ_o ensemble d'événements observables et Σ_{uo} : ensemble d'événements non observables
    - $\Sigma_o \cap \Sigma_{uo} = \emptyset$
  - $\delta : X \times \Sigma \rightarrow X$ ensemble des transitions ;
  - $x_0 \in X$ l'état initial;
- $L(G)$ est le langage généré par l'automate G,
- $L(G) = \{s \in \Sigma^* / \exists i \in X_0, \delta(i, s)\}$
- $P_o: \Sigma^* \rightarrow \Sigma_o{}^*$ est une projection

Observateur

- **Définition 9** : Un observateur $Obs(G)$ modélise l'ensemble des comportements observables d'un système G.
- Un état de l'observateur est une estimation à l'aides des évènements observables de l'état du modèle initial.
- **La taille de l'observateur est exponentielle par rapport à la taille du modèle.**



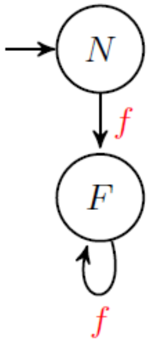
$\Sigma_o = \{b, d, e\}$ $\Sigma_{uo} = \{a, c, e, f\}$



Diagnostiqueur

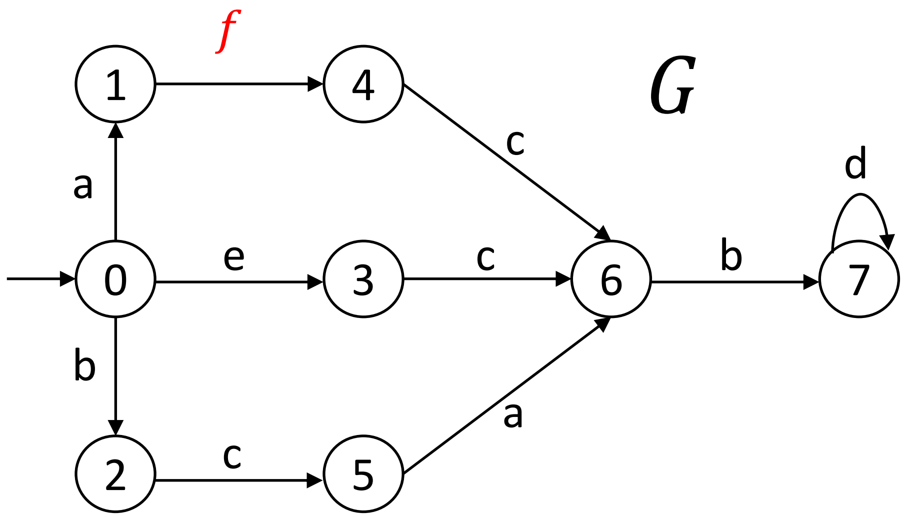
Diagnostiqueur (1) : Construction

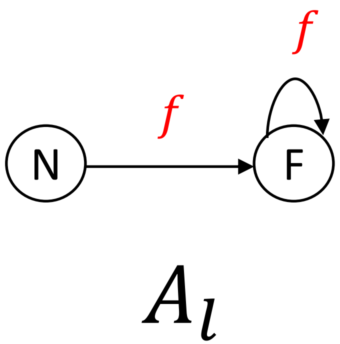
- **Définition :** Un diagnostiqueur est un observateur renseigné avec une fonction de décision qui décide si l'état courant et Normal (N), Fautif (F) ou incertain (N et F).
- **Méthode de construction off-line**
  - **Etape 1** : **Labellisation des états de l'automate G**
    - Etape 1-1 : Construire l'automate A_l
    - Etape 1-2 : Construire l'automate $G_l = G||A_l$
  - **Etape 2** : **Construction du diagnostiqueur** G_d =Obs(G_l)



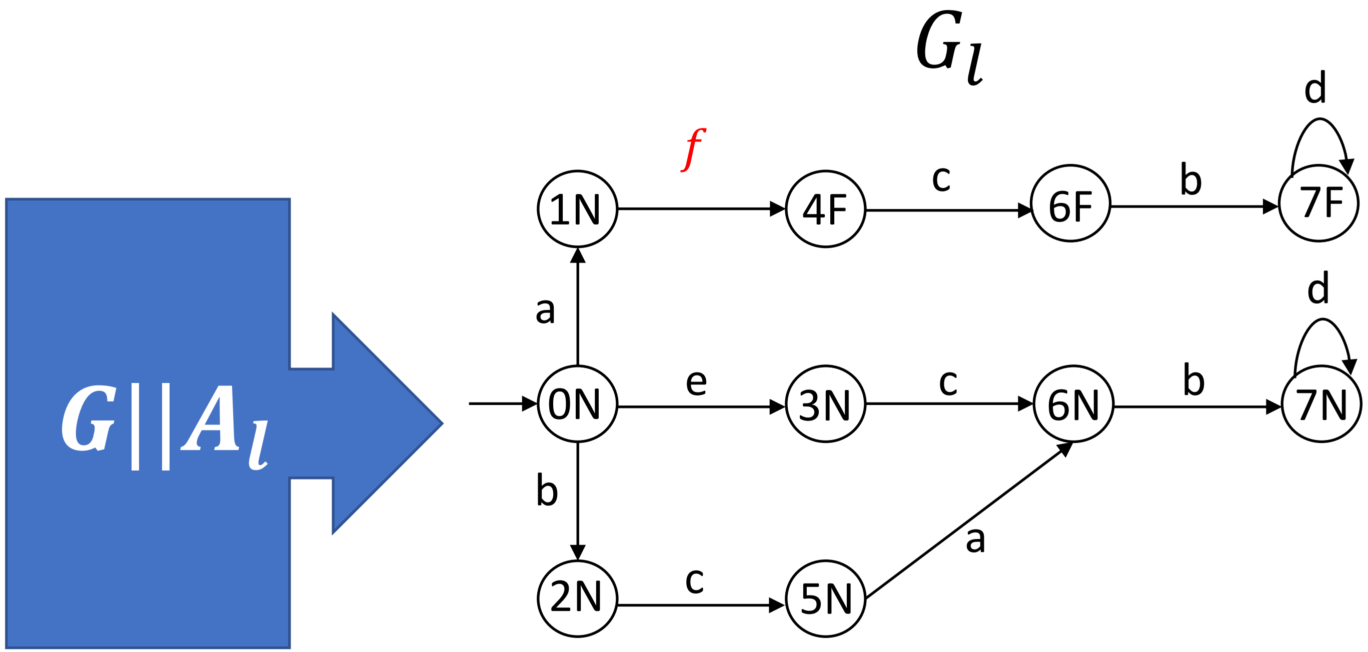
A_l

Diagnostiqueur (2) : labellisation

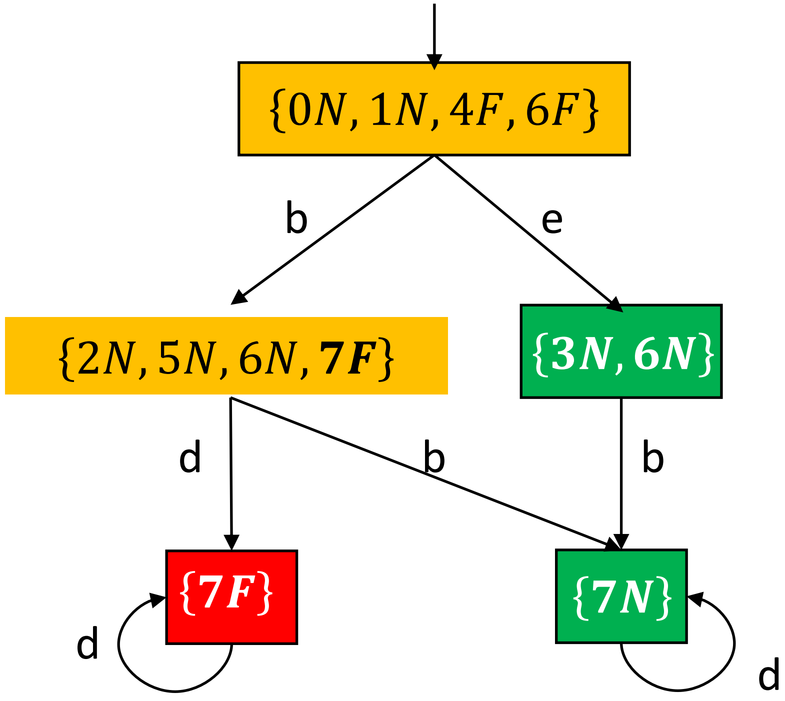




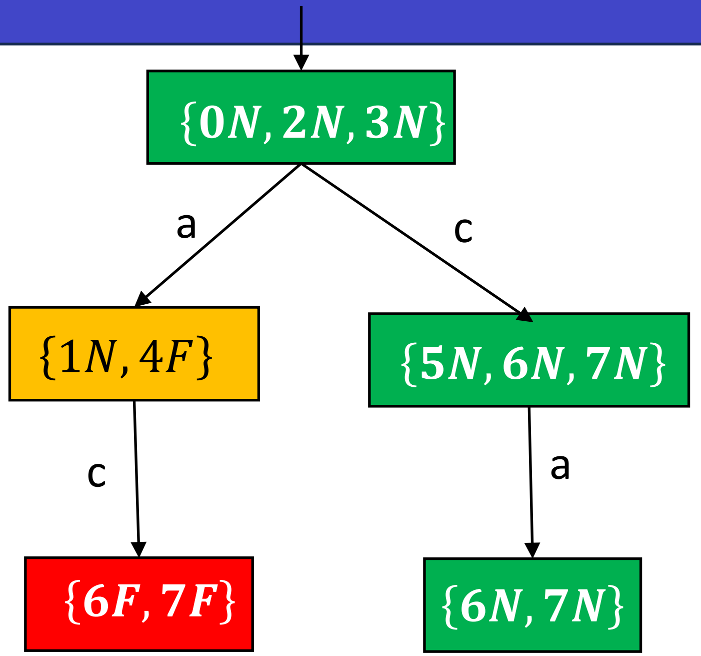
(automate de labellisation)



Diagnostiqueur (3) : construction de Gd



Diagnostiqueur
$G_{d,1}$
pour $\Sigma_o = \{b, d, e\}$



Diagnostiqueur
$G_{d,2}$
pour $\Sigma_o = \{a, c\}$

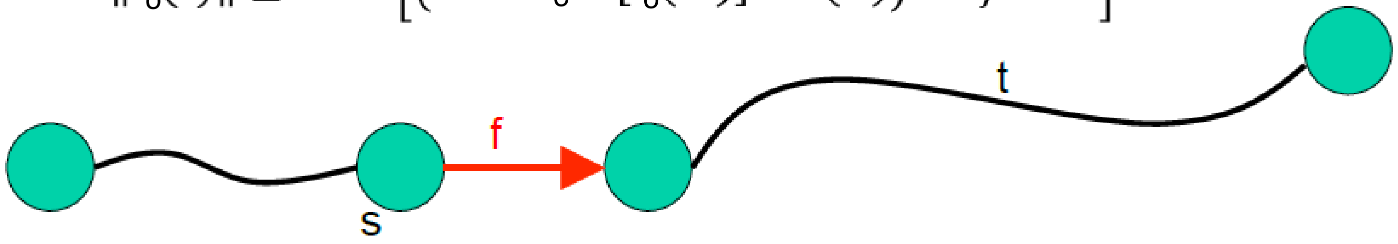
$$G_d = \mathrm{Obs}(G_l) = (X_d, \Sigma_o, \delta_d, x_{d0})$$

Diagnosticabilité

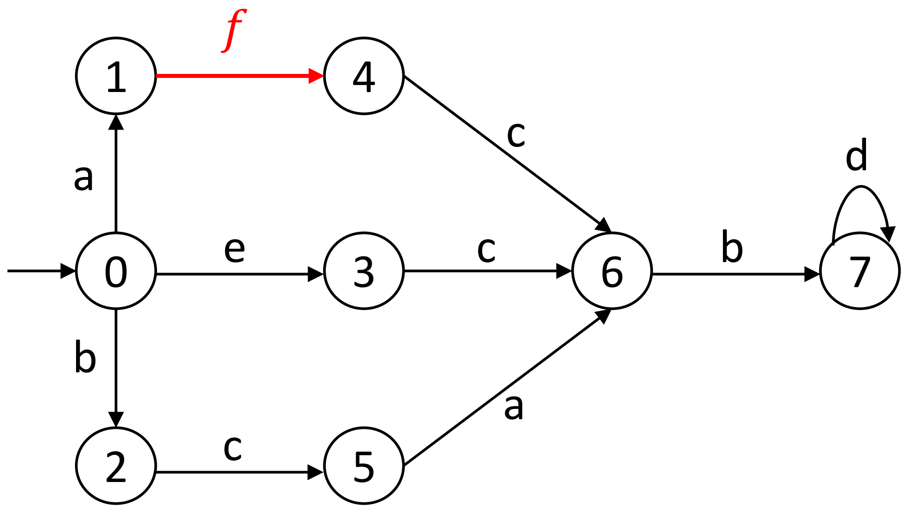
Diagnosticabilité (1)

- **Définition 10** : Une faute est diagnosticable si elle peut être détectée avec certitude après un nombre fini d’événements observables après son occurrence
- **Définition 11** : Une faute f est diagnosticable si pour chaque trace s se terminant par f, il existe une suite suffisamment longue t telle que toute autre trace indiscernable de st (produisant le même enregistrement d'événements observables) contient f (est également défaillante).

$$\|P_o(t)\| \geq n \Rightarrow \left[\left(\forall \omega \in P_o^{-1}[P_o(st)] \cap L(G)\right) \Rightarrow f \in \omega\right]$$



Diagnosticabilité (2)



G

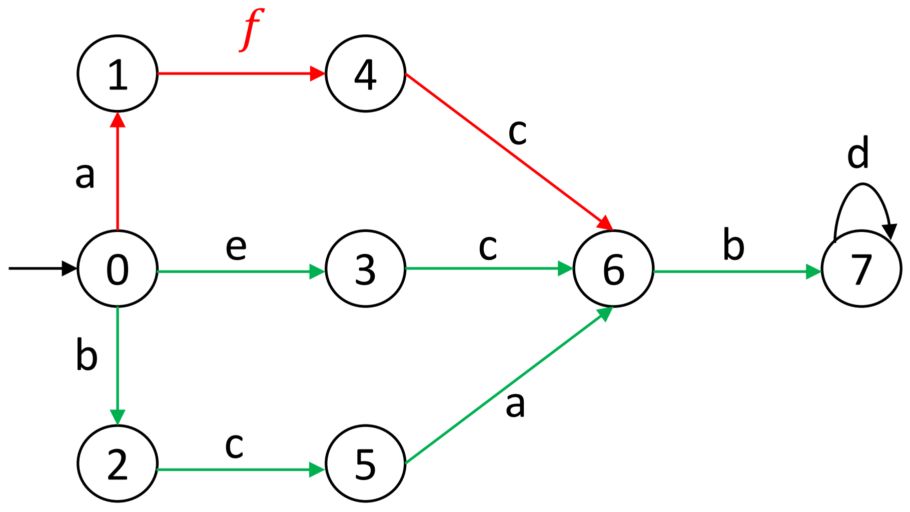
$$L(G) = \overline{(ec + bca + afc)bd^*}$$

$$s_f = afcbd^*$$

$$s_n = ecbd^*$$

$$s_n' = bcabd^*$$

Diagnosticabilité (3)



G

$$L(G) = \overline{(ec + bca + afc)bd^*}$$

$$s_f = afcbd^*$$

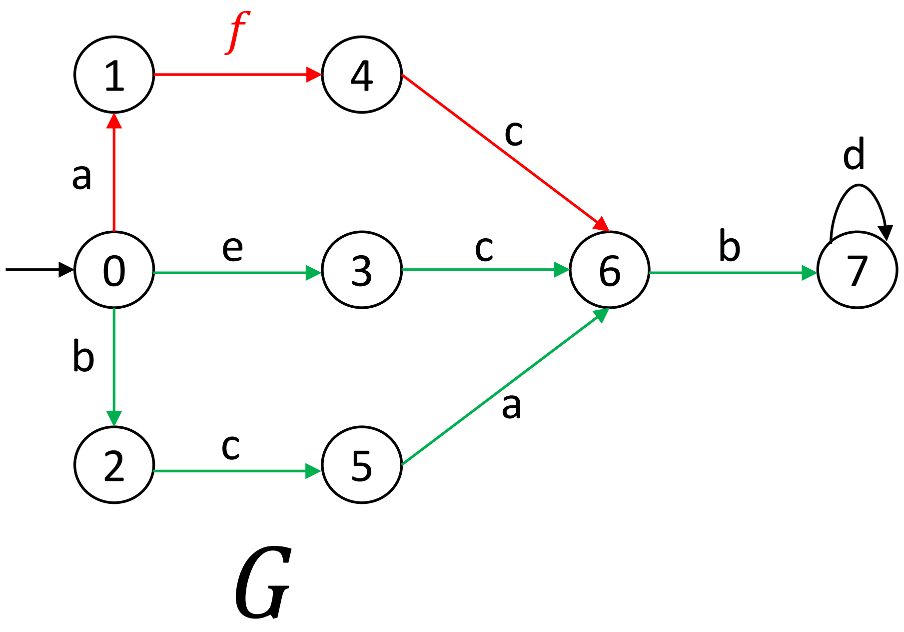
$$s_n = ecbd^*$$

$$s_n' = bcabd^*$$

Diagnosticabilité (4)

Test de la diagnosticabilité : G n'est pas diagnosticable s'il existe deux séquences équivalentes arbitrairement longues, l'une fautive et l'autre non.

$$s_f = afcbd^k$$
$$s_n = ecbd^k$$
$$s'_n = bcabd^k$$

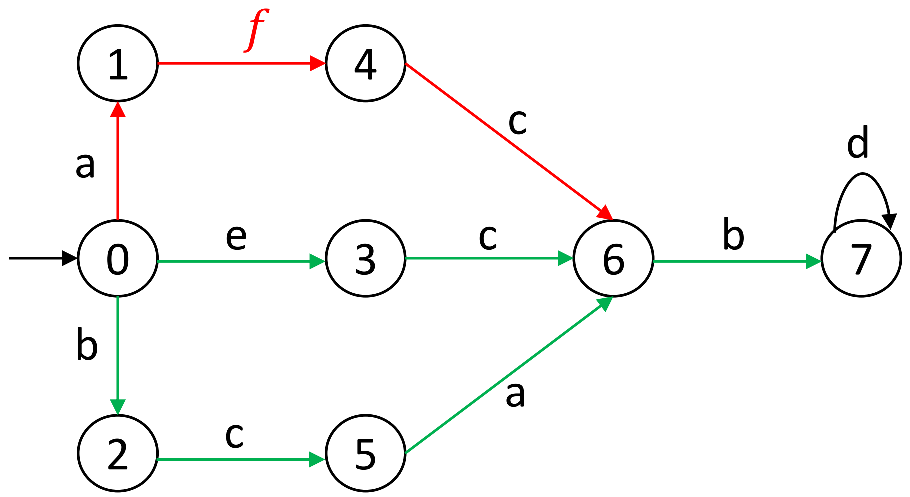


$\boldsymbol{\Sigma_o} = \{\boldsymbol{c}, \boldsymbol{d}, \boldsymbol{e}\}$ *et* $\boldsymbol{\Sigma_{uo}} = \{\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{f}\}$ alors $P_o(S_f) = P_o(S'_n) = cd^k$

L(G) n'est donc pas diagnosticable

Diagnosticabilité (5)

$$s_f = afcbd^k$$
$$s_n = ecbd^k$$
$$s'_n = bcabd^k$$



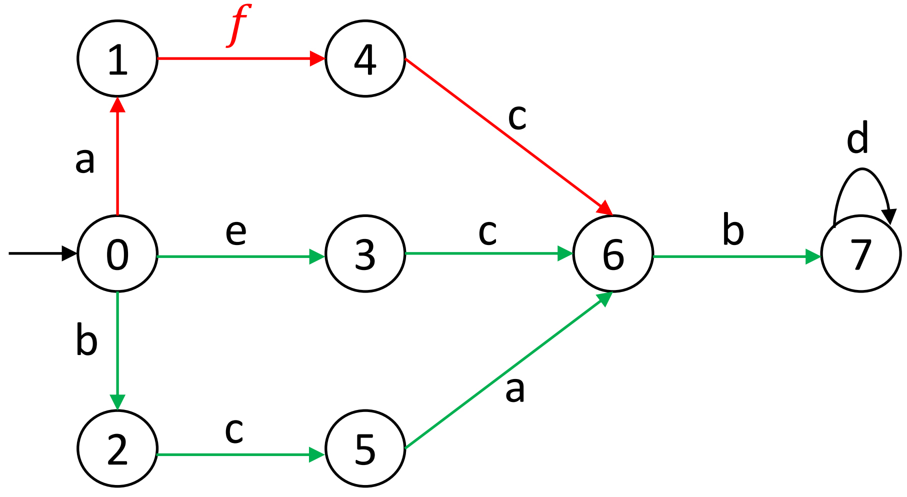
G

$\boldsymbol{\Sigma_o} = \{\boldsymbol{b}, \boldsymbol{d}\}$ ***et*** $\boldsymbol{\Sigma_{uo}} = \{\boldsymbol{a}, \boldsymbol{c}, \boldsymbol{e}, \boldsymbol{f}\}$ alors $P_o(S_f) = P_o(S'_n) = bd^k$

L(G) n'est donc pas diagnosticable

Diagnosticabilité (6)

$$s_f = afcbd^k$$
$$s_n = ecbd^k$$
$$s_n' = bcabd^k$$

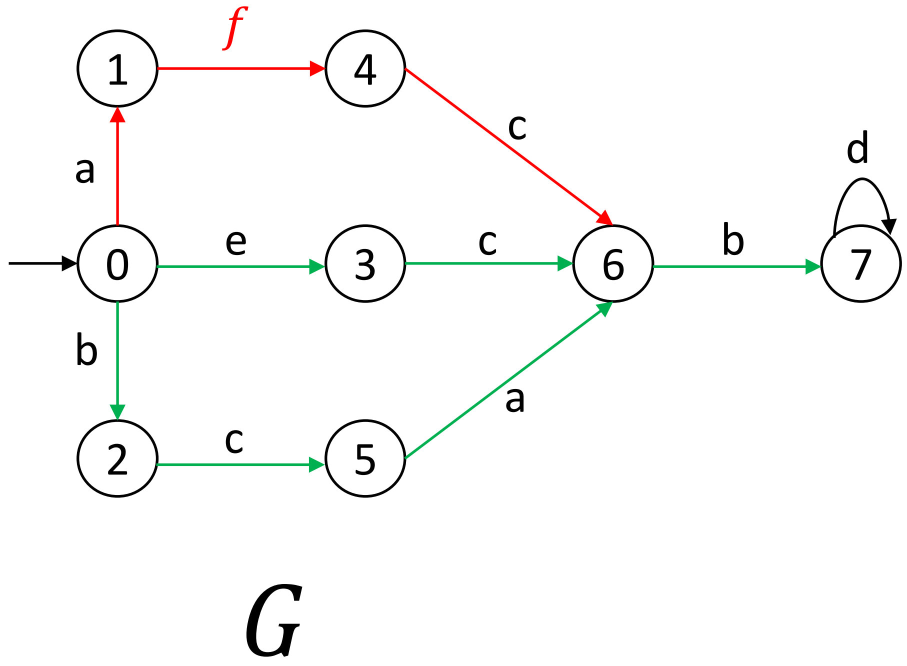


G

$\boldsymbol{\Sigma_o} = \{\boldsymbol{b}, \boldsymbol{d}, \boldsymbol{e}\}$ ***et*** $\boldsymbol{\Sigma_{uo}} = \{\boldsymbol{a}, \boldsymbol{c}, \boldsymbol{f}\}$ alors $P_o(S_f) = bd^k$
$et\ P_o(S_n) = ebd^k et\ P_o(S_n') = b^2 d^k$

L(G) est donc diagnosticable

Diagnosticabilité (7)



$$s_f = afcbd^k$$
$$s_n = ecbd^k$$
$$s'_n = bcabd^k$$

$\Sigma_o = \{a, c\}\ et\ \Sigma_{uo} = \{b, d, e, f\}$ alors $P_o(S_f) = ac$
$et\ P_o(S_n) = c\ et\ P_o(S'_n) = ca$

L(G) est donc diagnosticable

Diagnosticabilité (8)

- Soient $\sigma \in \Sigma$ un événement , $\Psi(\sigma)$ est l'ensemble des séquences de $L(G)$ qui finissent par σ,

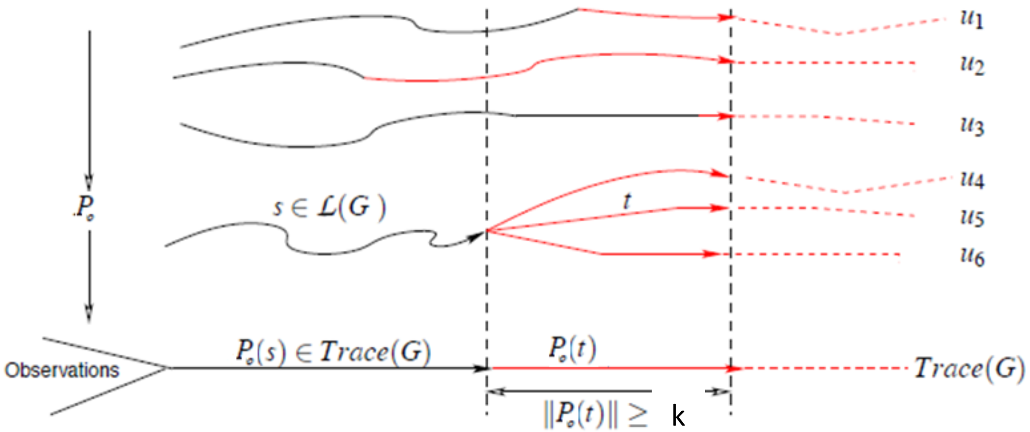
$$\Psi(\sigma) = \{s\sigma \in L(G): s \in \Sigma^*, \sigma \in \Sigma\}$$

- $L(G)/_s = \{t \in \Sigma^*: st \in L(G)\}$
- **Définition 12**: $L(G)$ est diagnosticable par rapport à une projection P_o et $\Sigma_f = \{f\}$ si la propriété suivante est vraie :

  Propriété : $(\exists k \in \mathbb{N})(\forall s \in \Psi(\Sigma_f))(\forall t \in L(G)/_s)(\|P_o(t)\| \geq k \Rightarrow D)$,

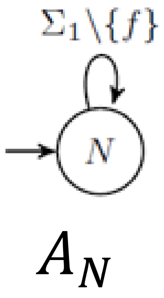
  Où la condition de diagnosticabilité D est

  D: $(\forall \omega \in P_o^{-1}(P_o(st)) \cap L(G))(\Sigma_f \in \omega)$

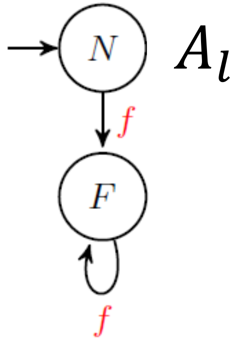


Vérification de la diagnosticabilité

Vérificateur de Moreira (Moreira et al., 2011)



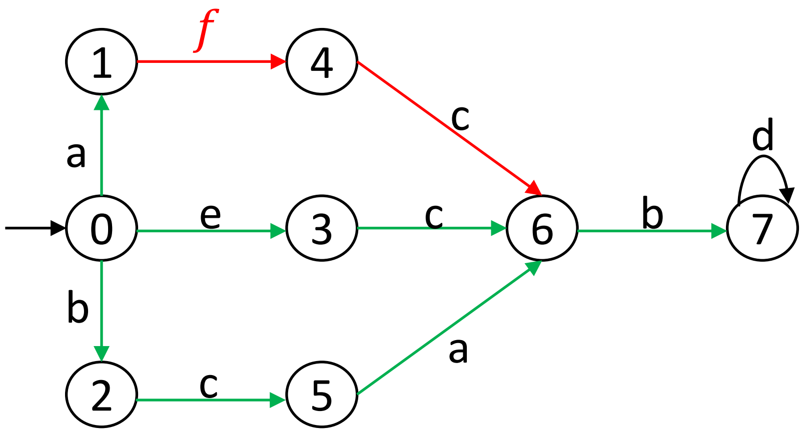
- **Etape 1 : Construction de l'automate normal**
  - Etape 1-1 : Construire l'automate A_N
  - Etape 1-2 : Construire G_N=G X A_N
  - Etape 1-3 : renommer les évènements non observables de G_N
- **Etape 2 : Construction de l'automate du comportement fautif**
  - Etape 2-1 : Construire l'automate A_l
  - Etape 2-2 : Construire G_l= G || A_l
  - Etape 2-3 : Marquer les états étiquetés par F
  - Etape 2-3 : Construire G_F=CoAc(G_l)
- **Etape 3 : Construction du vérificateur G_V=G_N||G_F**

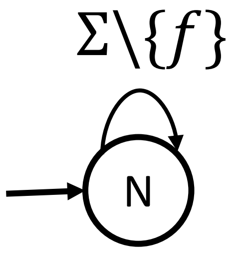


Vérification de la diagnosticabilité (2)

Etape 1 : Construction de l'automate normal

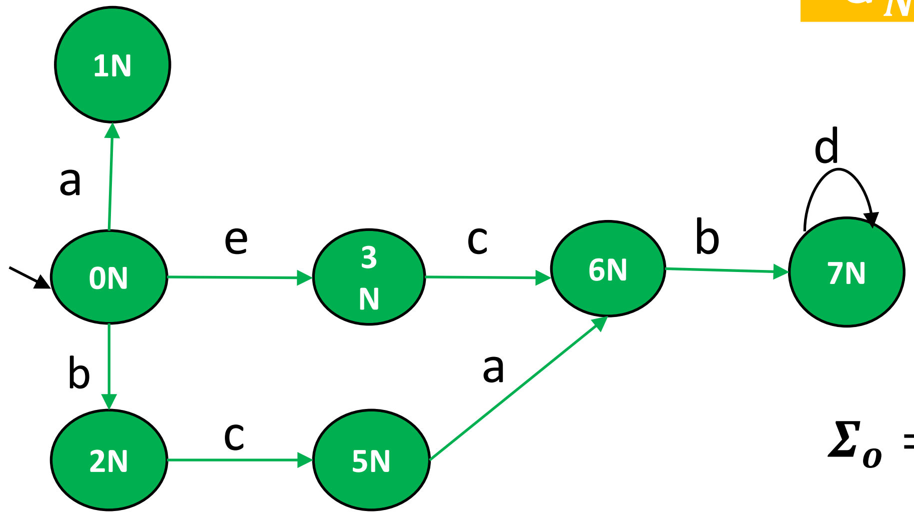
G





A_N

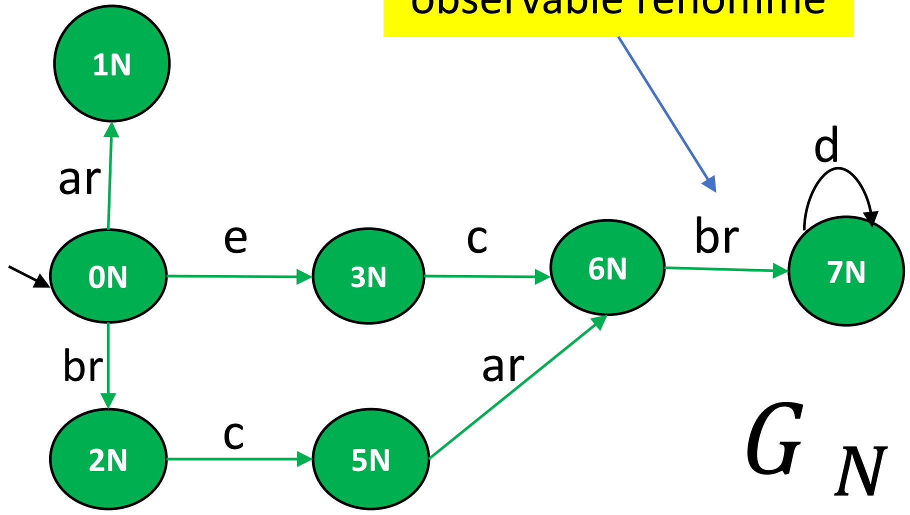
$G_N = G \times A_N$



Renommage des événements non observables

$\boldsymbol{\Sigma_o} = \{\boldsymbol{c}, \boldsymbol{d}, \boldsymbol{e}\}$ *et* $\boldsymbol{\Sigma_{uo}} = \{\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{f}\}$

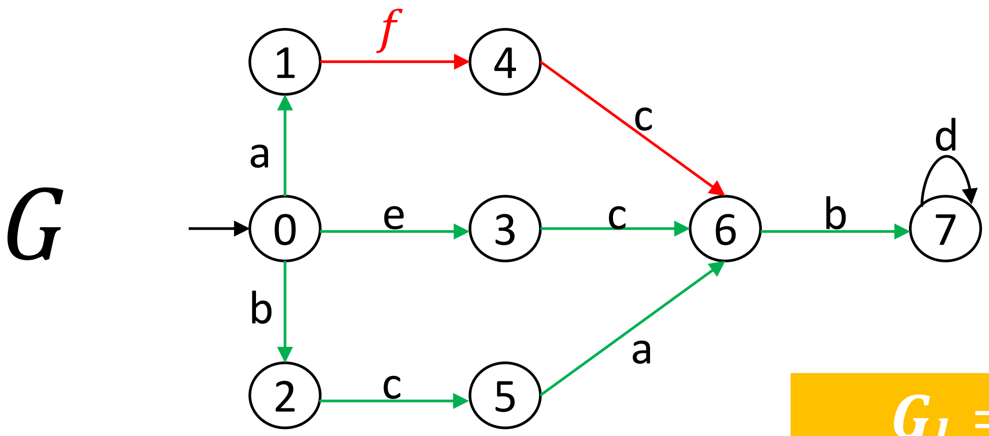
Evènement non observable renommé

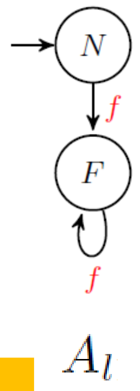


G_N

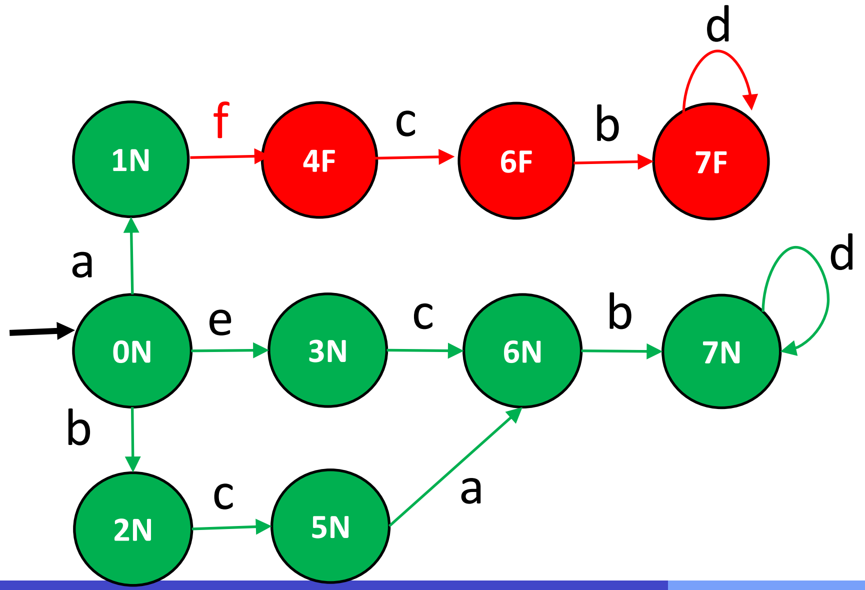
Vérification de la diagnosticabilité (3)

Etape 2-1 : Construction de l'automate du comportement fautif

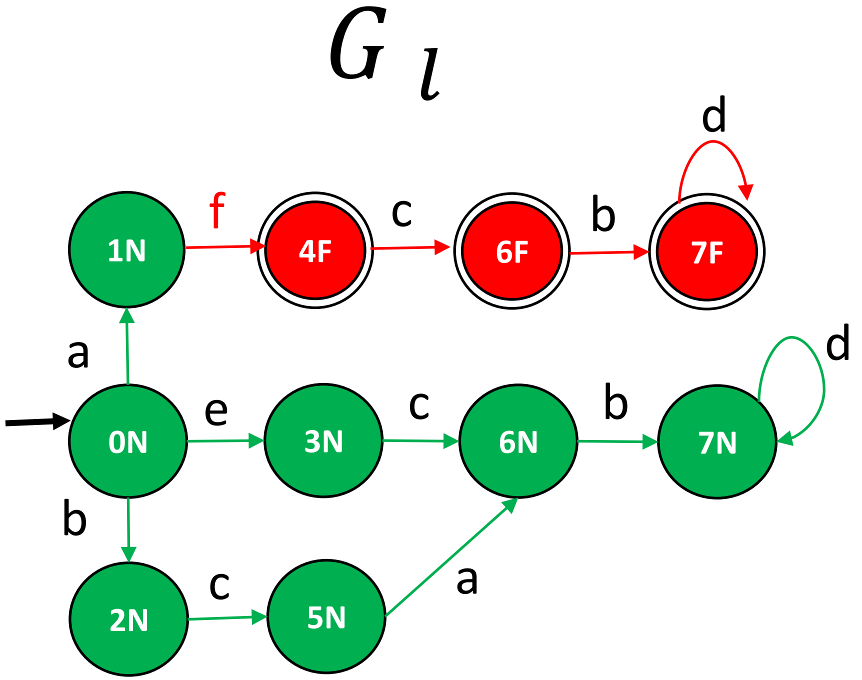




$G_l = G||A_l$

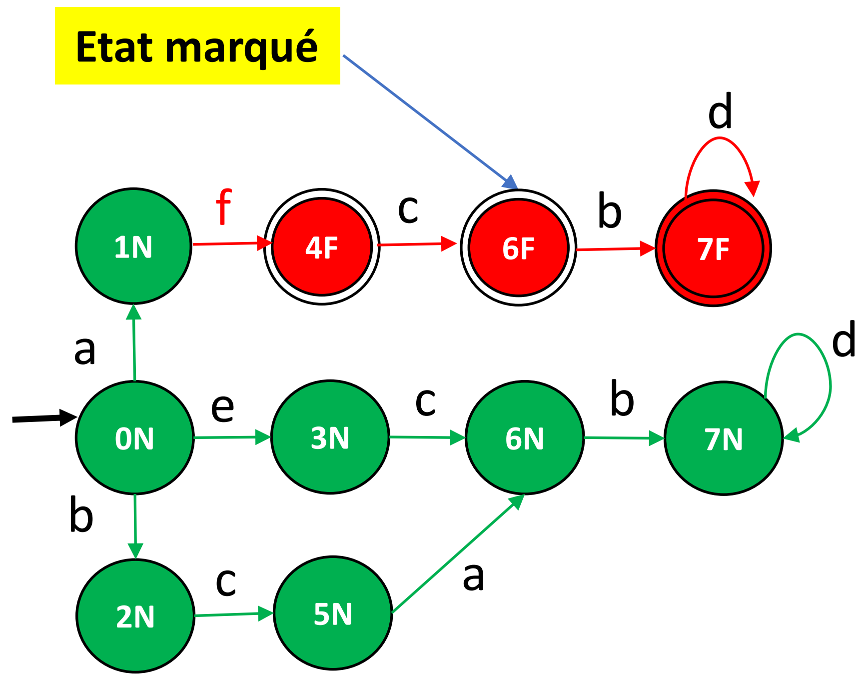


Etape 2-2 : Marquer les états défaillants



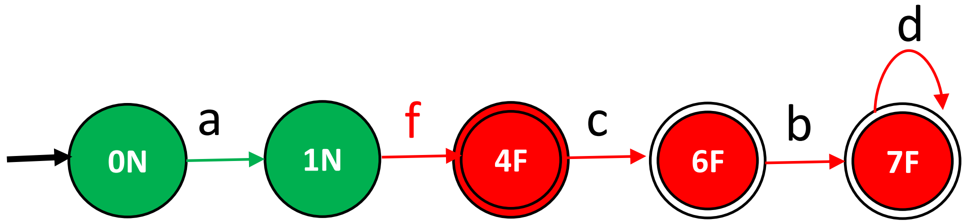
Vérification de la diagnosticabilité (4)

Etape 2-3 : Extraction de l'automate du comportement défaillant



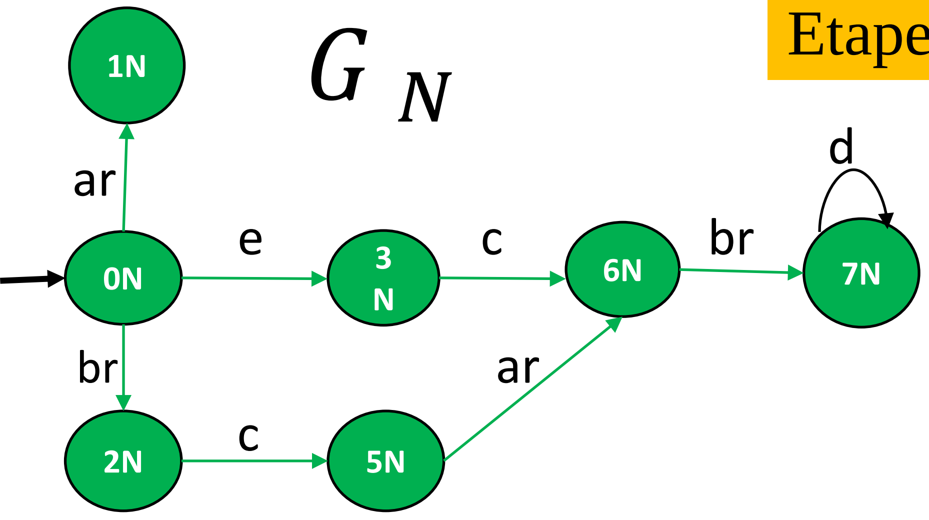
G_l

$G_F = CoAc(G_l)$



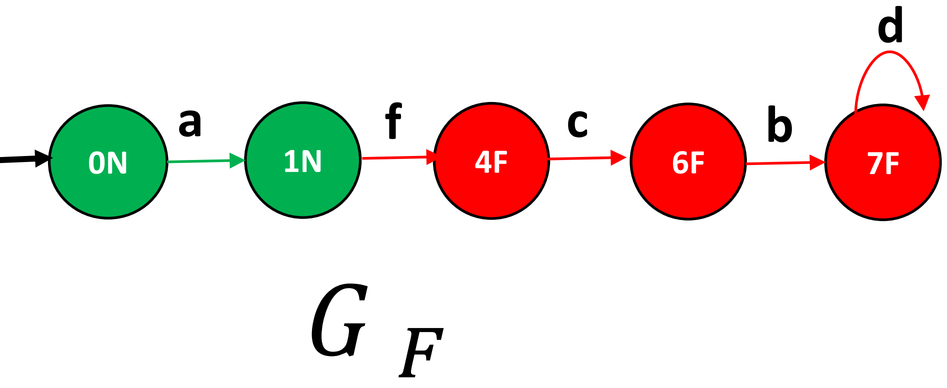
G_F

Vérification de la diagnosticabilité (5)

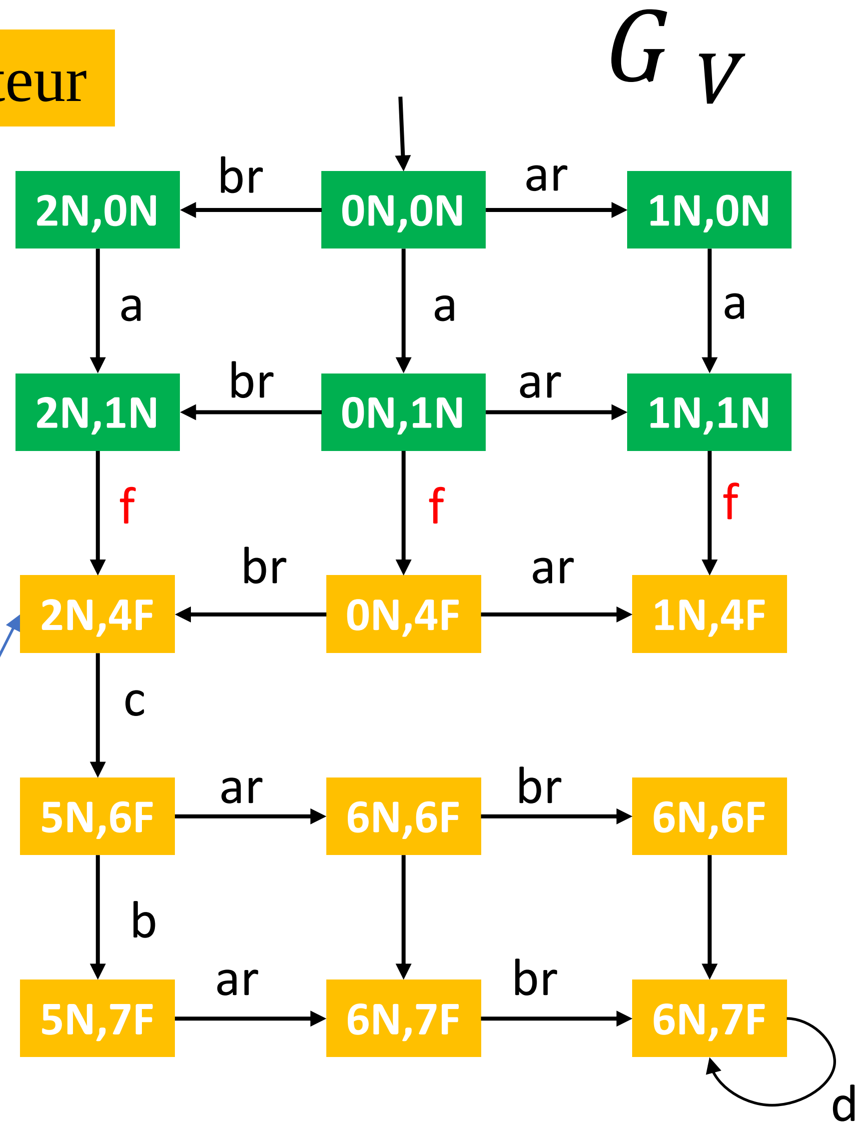


Etape 3 : Construction du vérificateur

$$G_V = G_N || G_F$$



Etat F-incertain

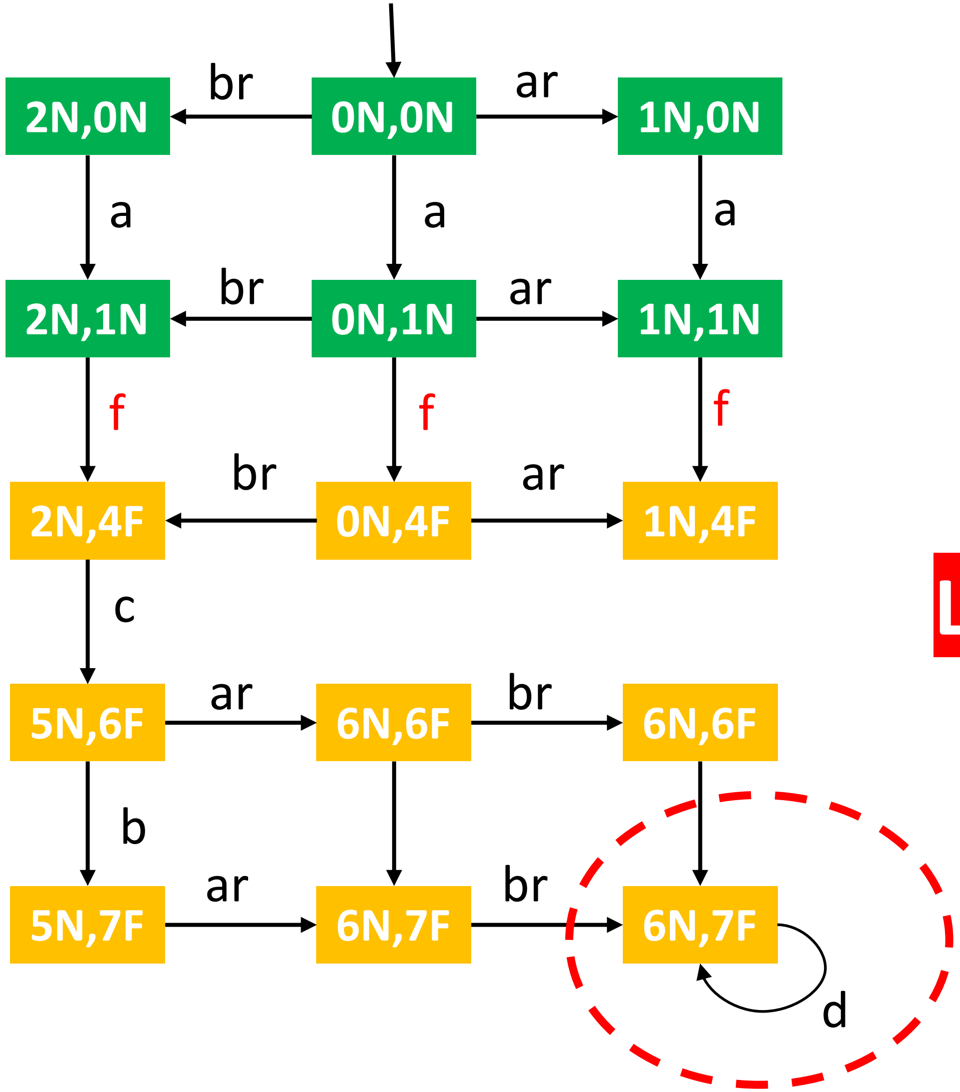


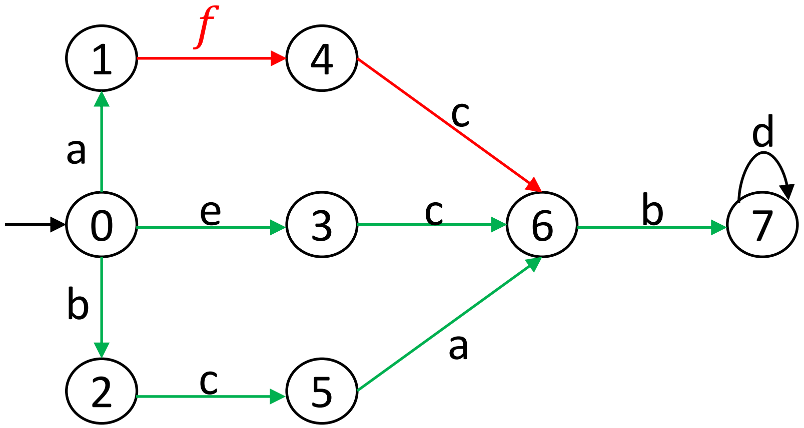
Vérification de la diagnosticabilité (6)

- **Définition : Cycle F-ambigu**
  - Un cycle du vérificateur est dit F-ambigu, s'il est formé d'états F-incertains.
- **Théorème 1 (Moreira et al. , 2011) : Diagnosticabilité**
  - Un système G est diagnosticable, si et seulement s'il n'existe pas de cycle F-ambigus dans le vérificateur de Moreira.
- Remarque :
  - Le vérificateur de Moreira a la meilleure complexité parmi tous les outils de vérification de la diagnosticabilité (Complexité polynomiale),
  - Pour assurer le diagnostic en ligne, il est nécessaire de construire un diagnostiqueur (peut être construit à la volée)

Vérification de la diagnosticabilité (7)

G_V





$$\Sigma_o = \{c, d, e\} \ et \ \Sigma_{uo} = \{a, b, f\}$$

L(G) n’est donc pas diagnosticable

Présence d’un cycle d’états F-ambigu

Plan

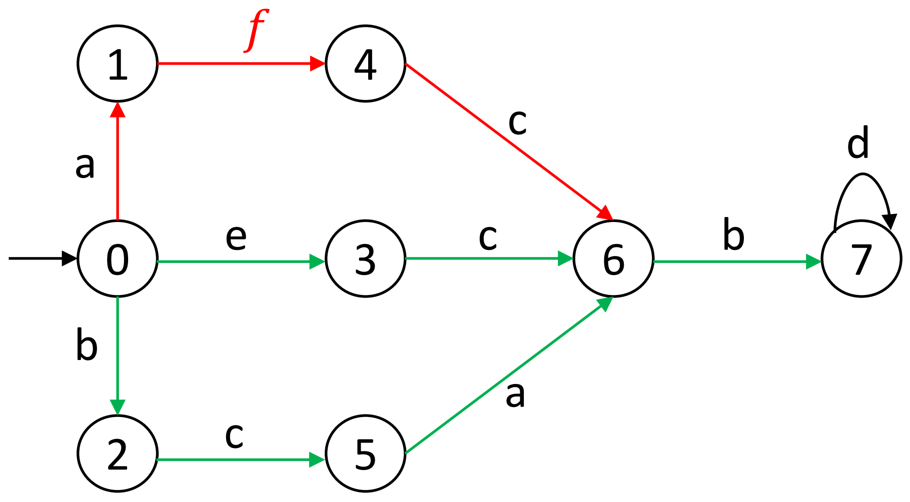
Pronostic

Pronostic SED

- La notion de pronostic dans le contexte des SED a été définie formellement pour caractériser la classe des langages pour lesquels :
  - (i) chaque faute est pronostiquée (c'est-à-dire prédite) **avant qu'elle ne se produise** ;
  - (ii) après qu'une faute ait été pronostiquée, elle se produit **au bout d'un nombre fini d'évènements**.

- Remarque : A ne pas confondre avec le pronostic en automatique qui consiste à évaluer le temps résiduel de fonctionnement d'un système après l'occurrence d'une faute dans le système.

Pronostic



G

$$s_f = afcbd^n$$
$$s_n = ecbd^n$$
$$s'_n = bcabd^n$$

$\Sigma_o = \{a, c\}\ et\ \Sigma_{uo} = \{b, d, e, \sigma_f\}$ alors $P_o(S_f) = ac$
Et $P_o(S_n) = c\ et\ \ P_o(S'_n) = ca$

L(G) est donc pronosticable

Pronostic : pronosticabilité

Pronosticabilité (1)

- **Définition 13 (Genc & Lafortune, 2009 ; Watanabe et al., 2021 ; Chouchane et al., 2024)**
  - Un langage préfix-clos et vivant $L(G)$, généré par un automate G, est pronosticable par rapport à une projection P_o et $\Sigma_f = \{f\}$ si les propriétés suivantes sont vraies :

P1 : $(\exists n \in \mathbb{N})(\forall s \in \Psi(\Sigma_f))(\exists t \in \overline{s})(f \notin t)(\wedge P)$,

Où la condition de pronosticabilité P est

P2 : $(\forall u \in L(G))(\forall v \in {}^{L}/_{u})\ [P_o(u) = P_o(t)])(f \notin u) \wedge (\|v\| \geq n \Longrightarrow f \in v)$

Pronosticabilité (2)

- **Définition 14 (Watanabe et al.,2017).**
  - Si $G_d = (X^N \cup X^I \cup X^F, \Sigma, \delta, x_{d0})$ est un diagnostiqueur, $\boldsymbol{FI}$ **est l'ensemble des premiers états incertains atteints à partir de l'état initial**, en considérant toutes les chaînes existantes.

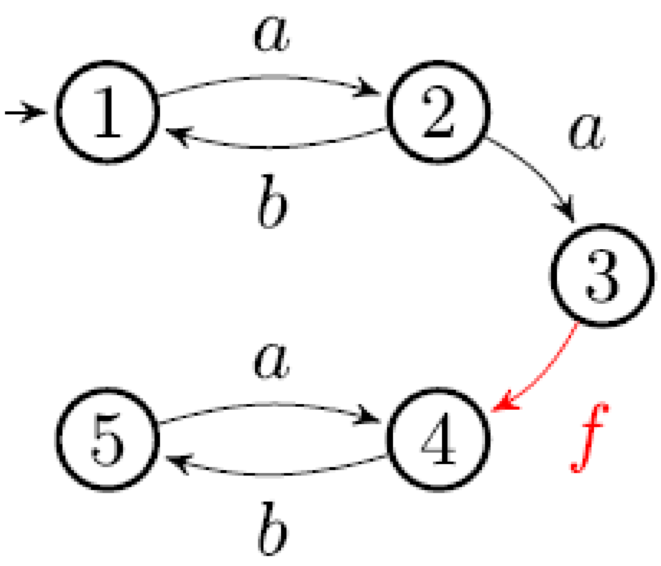
- $FI = \{x_d \in X^I : (\exists s_o \in \Sigma_o^*)\ t.q. (\delta(x_{do}, s_o) = x_d)\ et\ (\forall t_o < s_o)(\delta(x_{do}, t_o) \notin X^I)\}$ où $X^I = \{x_d \in X_d\ t.q. x_d\ est\ incertain\}$

Pronosticabilité (3)

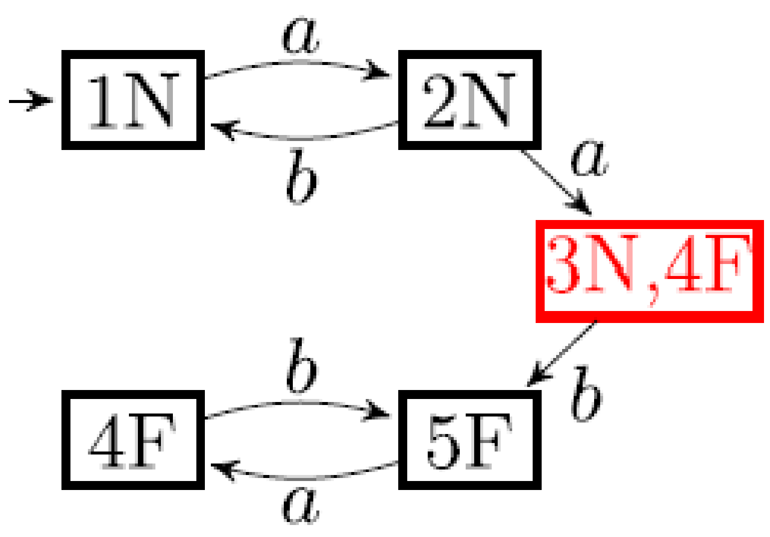
- **Théorème (Watanabe et al., 2017)**
  - Soit $G = (X, \Sigma, \delta, x_o)$ un automate qui génère L un langage prefix-clos et vivant. Soit $G_d = (X_d, \Sigma_o, \delta_d, x_{do})$ le diagnostiqueur de G par rapport à f et P_o. Les occurrences de f sont pronosticables dans L par rapport à P_o si et seulement si pour tout $x_d \in FI$, la condition $\mathbb{C}$ *où*
  - $\mathbb{C}$: tout cycle dans $A_c(G_d, x_d)$ est un cycle d'états certains du diagnostiqueur.

Pronosticabilité (4) : exemple 1

$\Sigma_o = \{a, b\}$ et $\Sigma_{uo} = \{f\}$



G



G_d

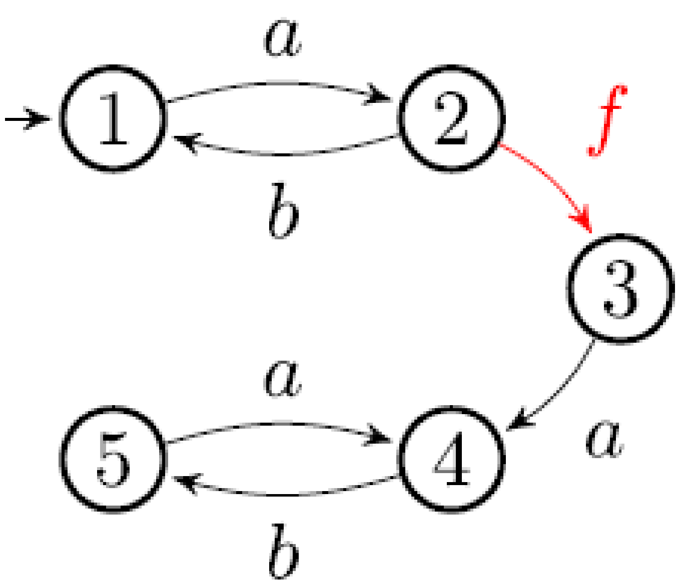
Après toute séquence $s_0 \in (ab)^* aa$ on est sûr d'atteindre un état défaillant

FI = {(3N, 4F)} et Ac (Gd, (3N, 4F))={(4F),(5F)}

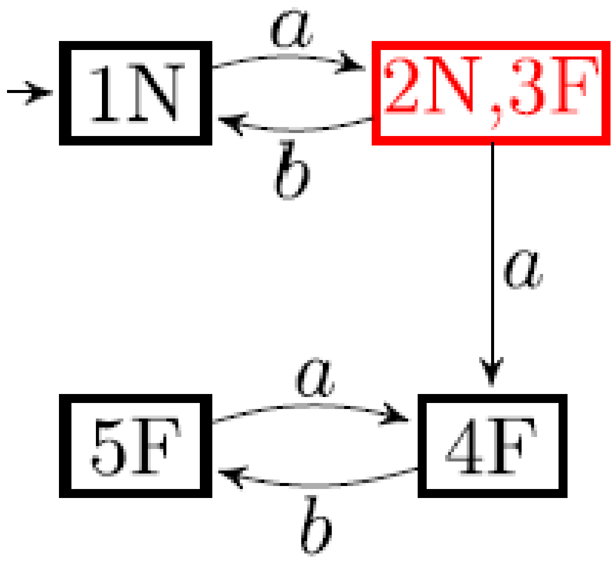
L(G) est donc pronosticable

Pronosticabilité (5) : exemple 2

$$\Sigma_o = \{a, b\}\ et\ \Sigma_{uo} = \{f\}$$



G

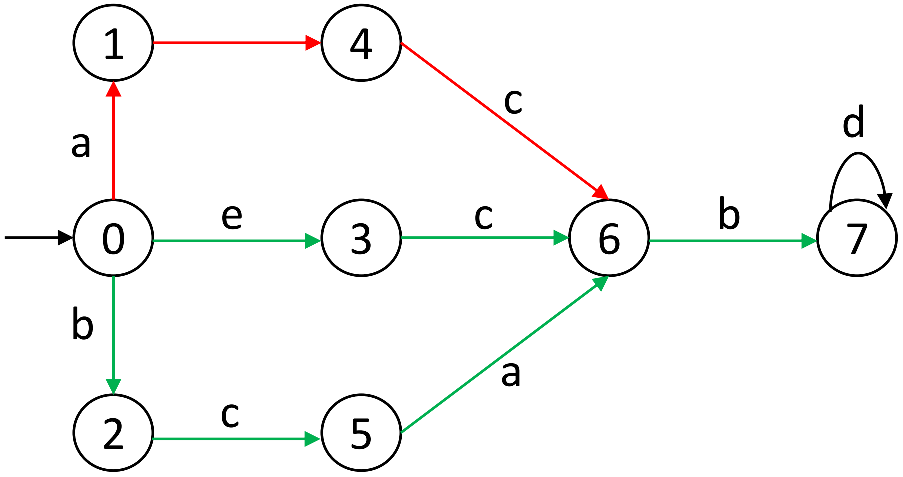


G_d

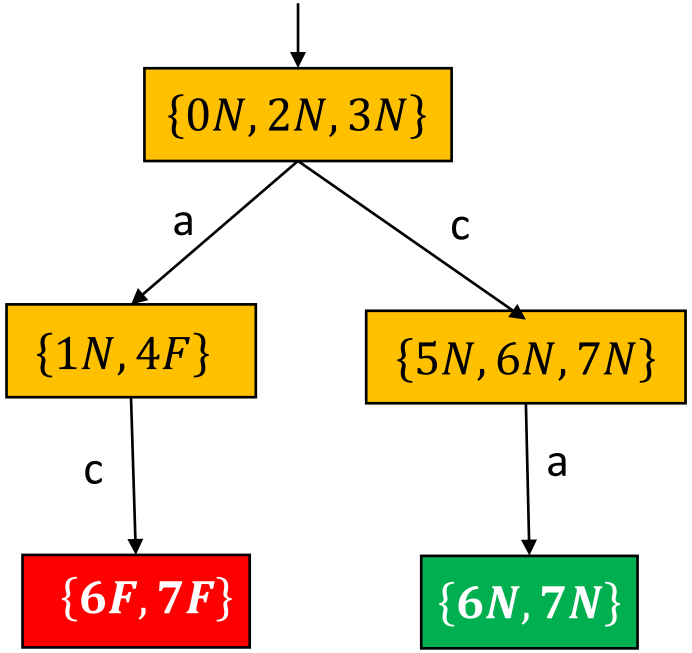
FI = {(2N, 3F)} et Ac (Gd, (2N, 3F))={((1N),(2N,3F)), (5F,4F)}

L(G) n'est pas pronosticable

Pronosticabilité (6) : exemple 3



G



FI = {(0N,*2N*,*3N*)} et Ac (Gd, (0N,*2N*,*3N*))={ }

Diagnostiqueur
$G_{d,2}$
pour $\Sigma_o = \{a, c\}$

L(G) est donc pronosticable

Plan

Opacité

Opacité (1)

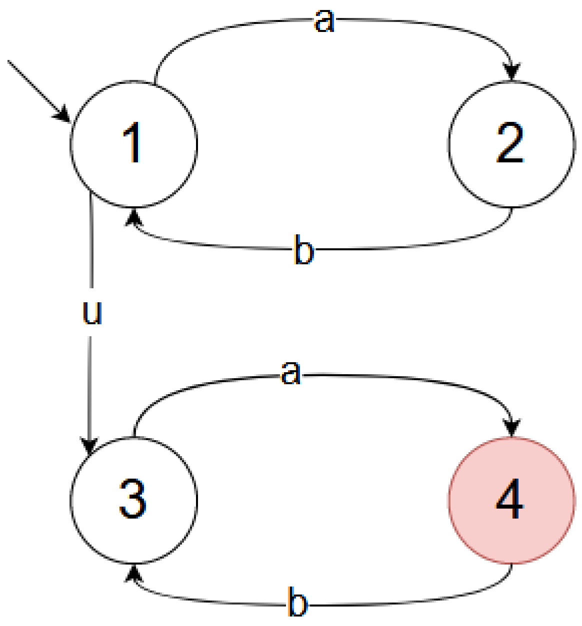
- **Propriété fondamentale** : Garder "secret" un ensemble d'états du système.
- **Définition 15 (Système opaque)** : Un intrus est incapable de déduire un "secret" sur le comportement du système.
- **Hypothèses par rapport à l'intrus (observateur)** :
  - **H1** : connaît la structure du système
  - **H2** : a accès à une base des observations

→ cherche à savoir si le système passe par certains états secrets

- **Principe d'une solution** : Pour assurer l'opacité du système, il faut que pour tout comportement secret, il existe au moins un autre comportement non secret qui lui soit équivalent d'un point de vue observationnel

Opacité (2)

- État initial = {1},
- État secret = {4},
- Événements observables Σ_o = {a, b}
- Événements non observables Σ_{uo} = {u}
- Comportement secret : (ab)*ua



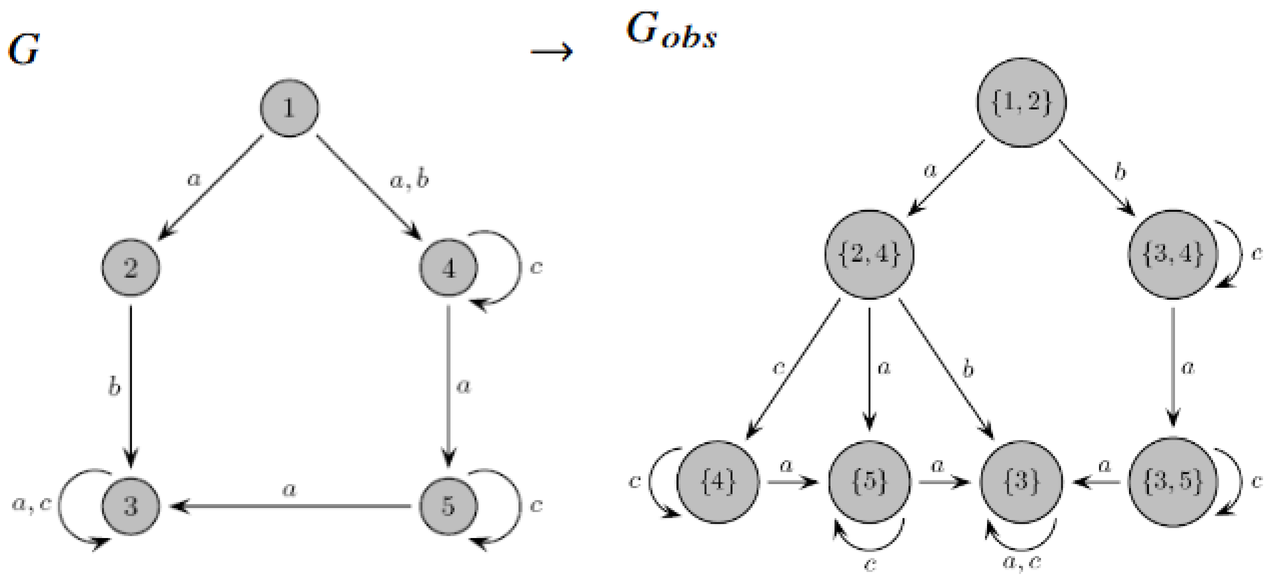
Opacité (3) : à l'état courant

- G est un automate à état finis, G = < X, Σ, δ, X0, > avec :
  - X : ensemble fini d'états;
  - Σ : ensemble fini d'événements ;
  - δ : X × Σ → X ensemble des transitions ;
  - $X_0 \subseteq X$ ensemble des états initiaux;
- L(G) est le langage généré par l'automate G,
- $L(G) = \{s \in \Sigma^* \mid \exists i \in X_0, \delta(i, s)\}$
- $P_o : \Sigma^* \rightarrow \Sigma_o{}^*$ est une projection
- $X_S \subset X$ est l'ensemble d'états secrets

G est **opaque** à l'état courant si

$$(\forall x_0 \in X_{0)}\ (\forall w \in L(G) : \delta(x_0, w) \in X_S)$$
$$(\exists x'_0 \in X_0)(\exists w' \in L(G) : [\delta(x'_0, w') \notin X_S \wedge P_o(w) = P_o(w')]$$

Opacité (4) : Exemple



$X = \{1, 2, 3, 4, 5\}$, $\Sigma = \Sigma_o = \{a,b,c\}$, $X_0 = \{1, 2\}$

Etats secrets	Opacité vérifiée	Contre exemple
$S_1=\{5\}$	non	(ac*ac*)
$S_2=\{4\}$	Non	(acc*)
$S_3=\{2\}$	Oui	

Plan

1. Introduction
   Entraves à la sûreté de fonctionnement
   Moyens pour la tolérance aux fautes
2. Diagnostic
3. Pronostic
4. Opacité
5. Conclusion
   Diagnostic des SED basés sur les Réseaux de Petri
   Extensions de la diagnosticabilité

Plan

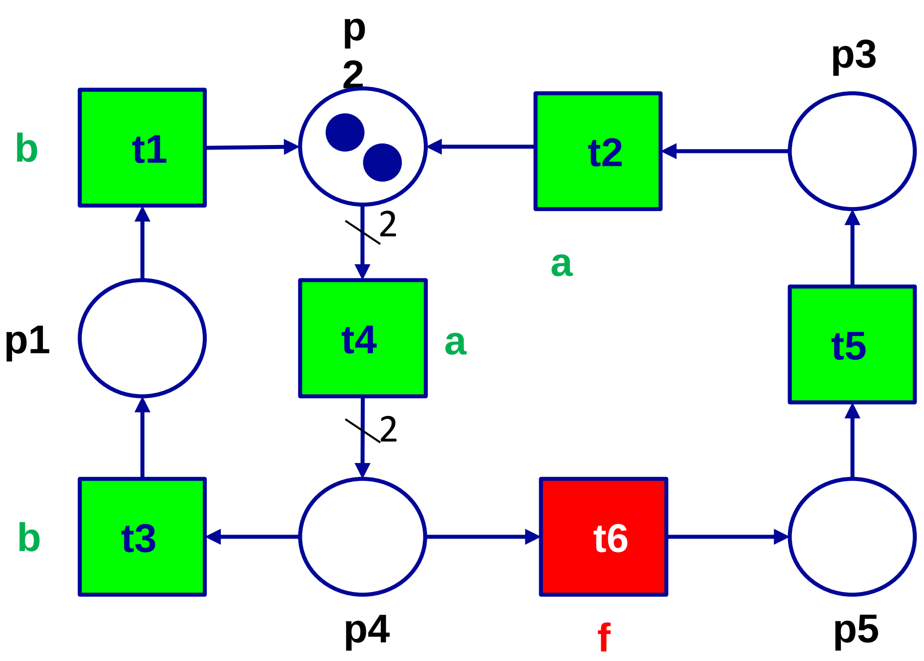
1. Introduction
2. Diagnostic
3. Pronostic
4. Opacité
5. Conclusion
    - Réseaux de Petri étiquetés
    - Extensions diagnosticabilité
    - Exercice sur Desuma

Conclusion

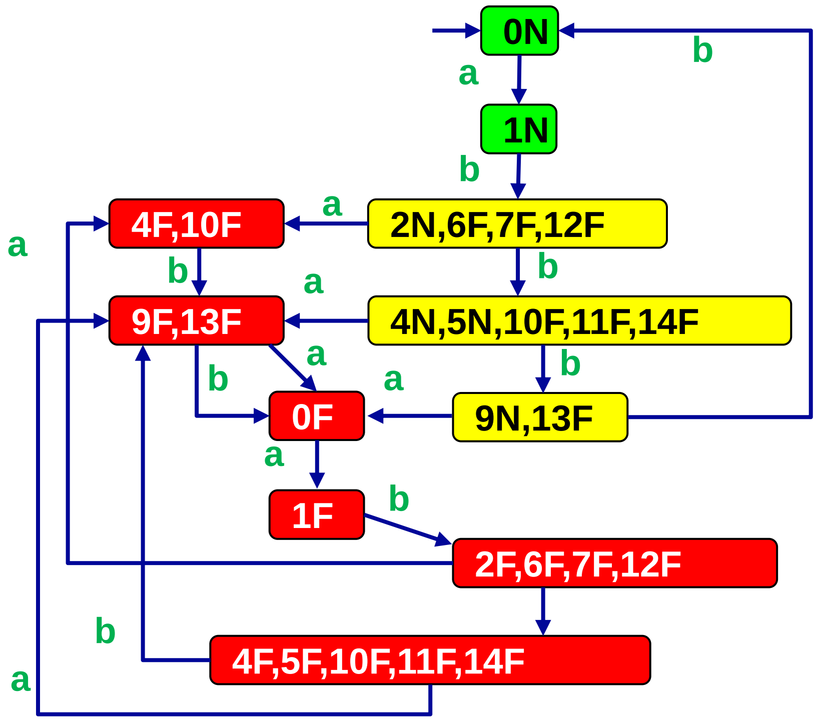
Réseaux de Petri étiquetés (1)

- **Définition 16 : Fonction d'étiquetage (labeling function)**
  - Étant donné un RdP N avec un ensemble de transitions T, une fonction d'étiquetage $\varphi : T \rightarrow \Sigma \cup \{\varepsilon\}$ attribue à chaque transition $t \in T$ un symbole, provenant d'un alphabet Σ donné, ou lui attribue la chaîne vide.
- **Définition 17 (Cabasino et al., 2010) : RdP étiqueté (L-RdP pour labeled RdP)**
  - Un RdP étiqueté (L-RdP) est triplet $G=(N, M0, \varphi)$, où $N =(P, T, Pre, Post)$ est un RdP, M0 est son marquage initial, et $\varphi : T \rightarrow \Sigma \cup \{\varepsilon\}$ est sa fonction d'étiquetage.
- **Remarque :**
  - En anglais on parle de Labeled Petri Nets ou LPN
  - Surtout à ne pas confondre avec les Réseaux de Petri Synchronisés (RdPS)

Réseaux de Petri étiquetés (2) : approche classique



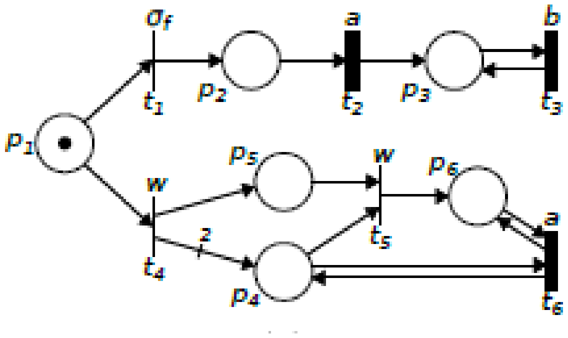
Modèle du système



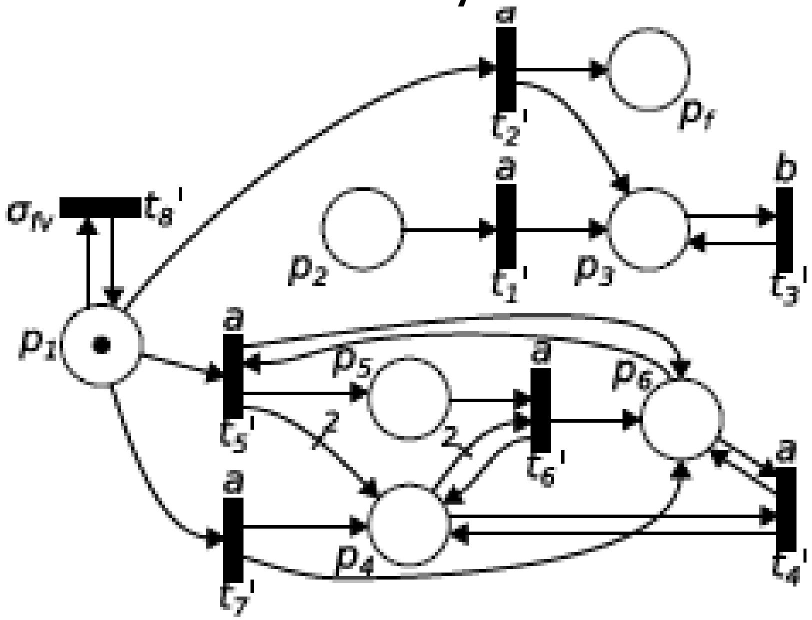
Diagnostiqueur

Réseaux de Petri étiquetés (3) : autres approches

- **Basic Reachability Graph (BRG) (Cabasino et al.,2010)**
  - Vérification de la diagnosticabilité des LPN bornés
- **LPN Verifier (Cabasino et al., 2012)**
  - Vérification de la diagnosticabilité des LPN non bornés
- **Augmented state class set graph (Liu, 2014)**
  - Vérification de la diagnosticabilité des LTPN
- **Diagnosticabilité modulaire des LPN (Li et al., 2017)**
- **Approches algébriques pour les RdP (Basile et al., 2012) (Chouchane and Ghazel, 2023)**
- **Diagnostiqueur LPN (de Freitas et al; 2022)**



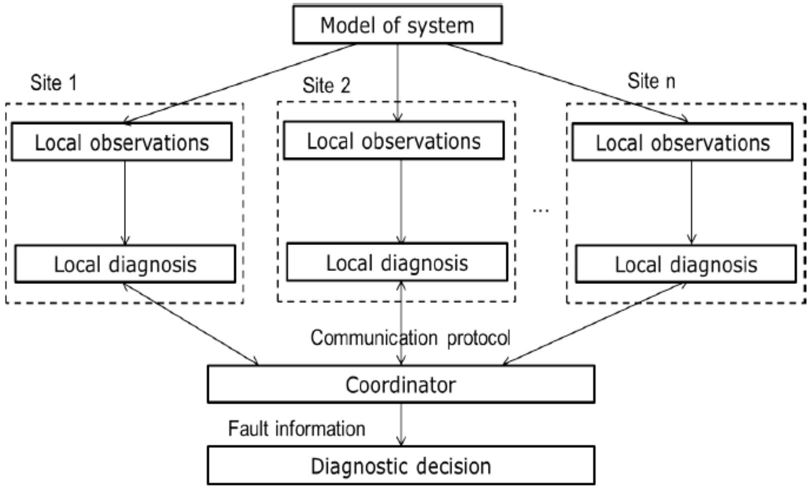
Modèle du système



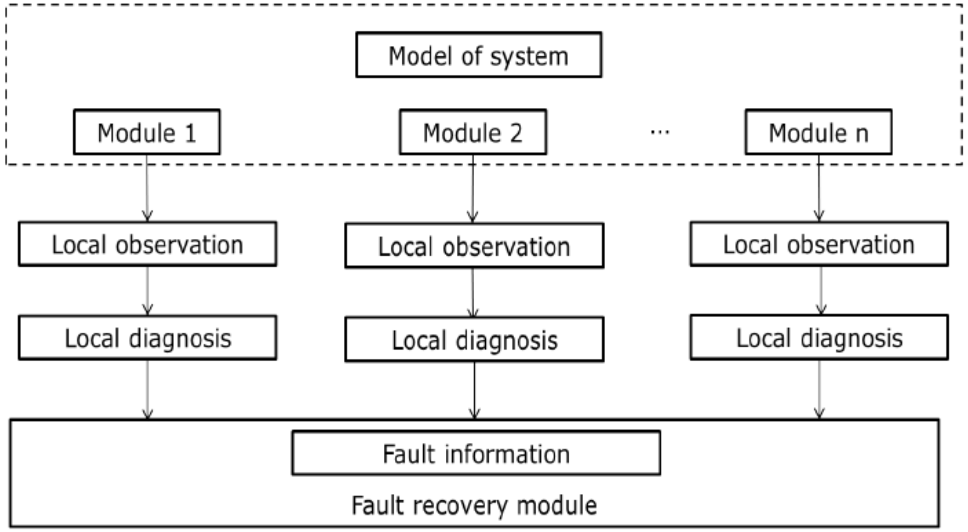
Diagnostiqueur

Extensions de la diagnosticabilité (1)

- **Diagnostic décentralisé et codiagnosticabilité** (Debouk et al., 2000 ; Moreira et al., 2011) : Toute faute appartenant à l'ensemble de fautes prédéfinies est diagnostiquée dans un délai borné **par au moins un** diagnostiqueur local.

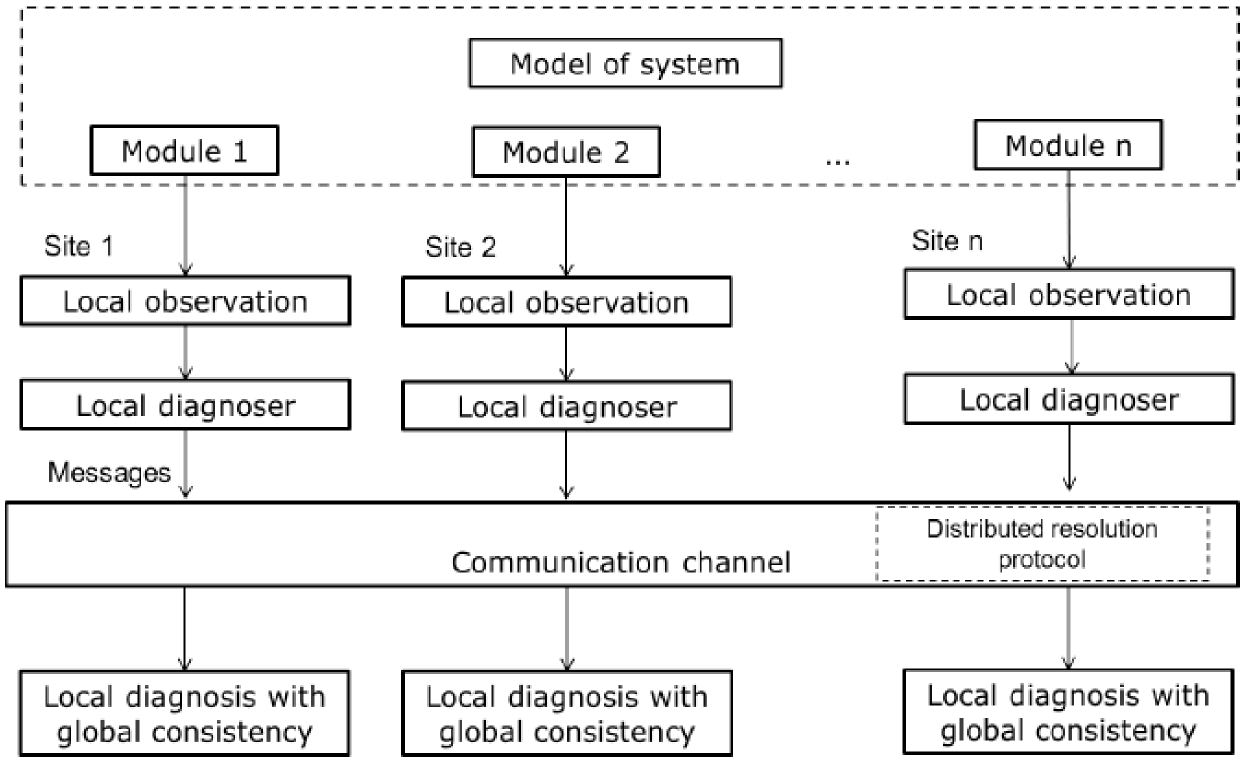


- **Diagnosticabilité Modulaire (Contant et al., 2006 ; Li et al., 2017 ; Basilio et Toguyéni 2023)** : Capacité pour tout module à diagnostiquer une faute interne après un nombre fini d'événements locaux observables.



Extensions de la diagnosticabilité (2)

- **Diagnostic distribué (Genc et Lafortune, 2003)** : C'est la capacité pour chaque diagnostiqueur local a diagnostiquer une faute du système à l'aide d'observations locales et d'informations de diagnostic communiquées par d'autres modules.
  - Problématique de l'approche : protocole de communication

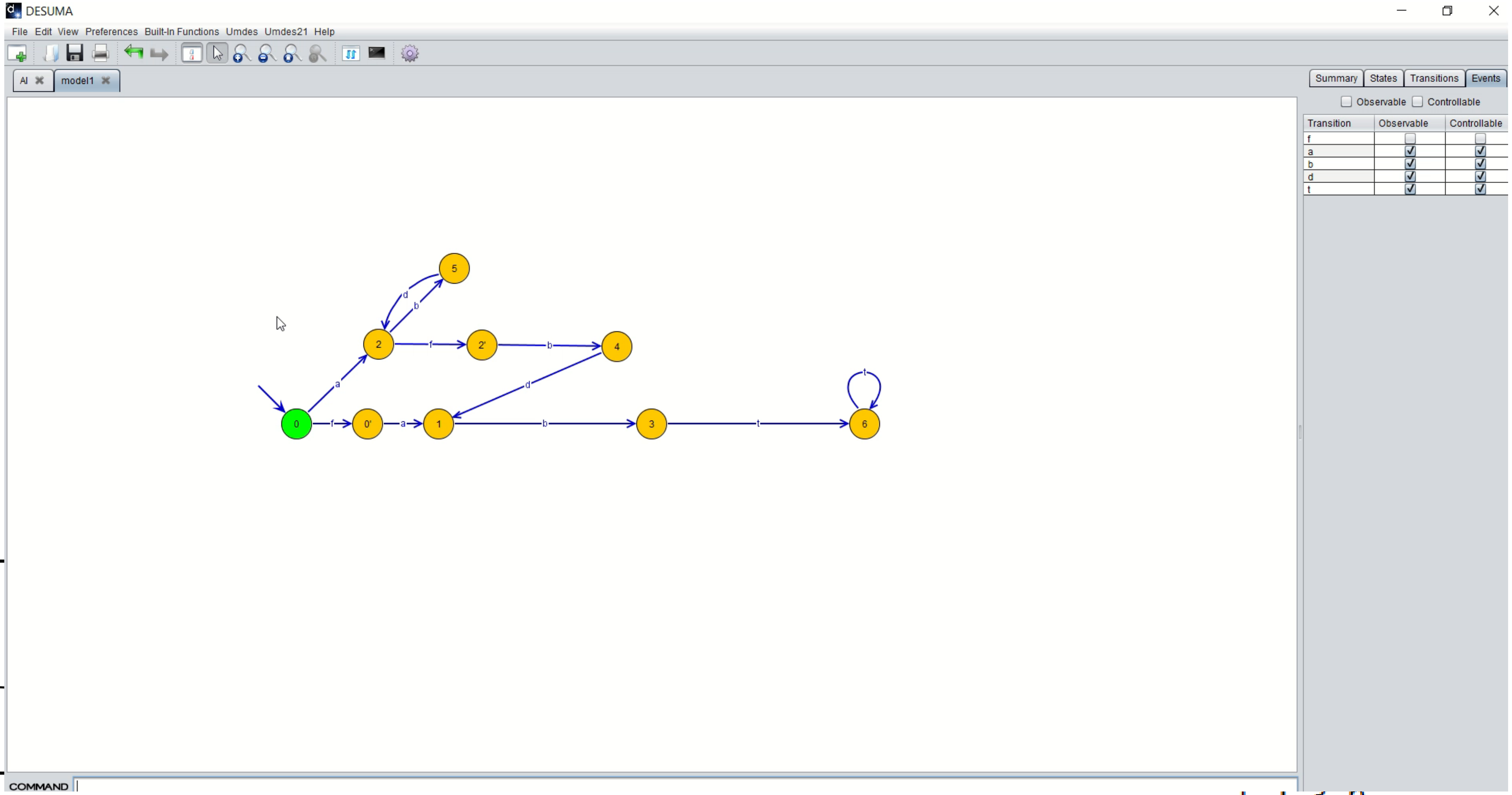


Stephane LAFORTUNE | PhD |...

Exercices sur Desuma

Exercice 1 sur Desuma

G

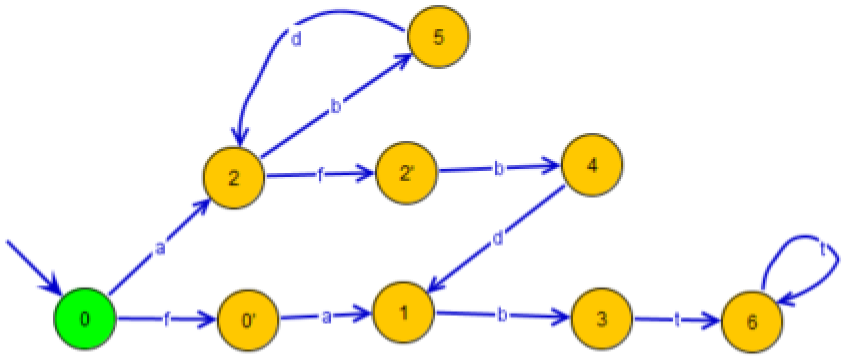


Obs(G)

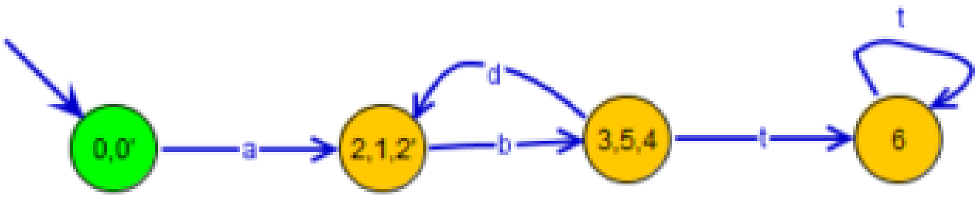
t 3 6 t
t 6 6 t

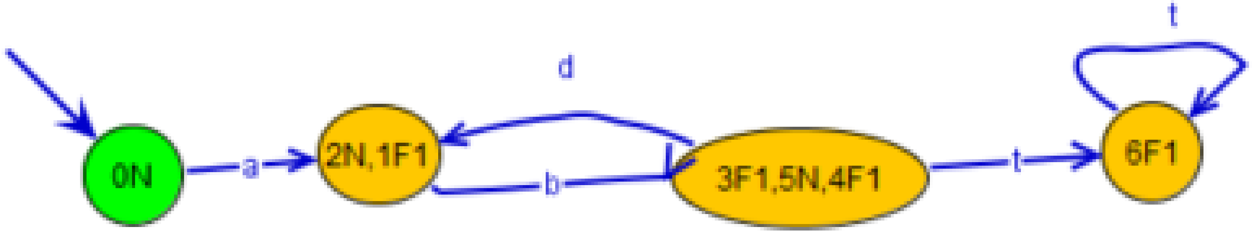
Exercice 1

G



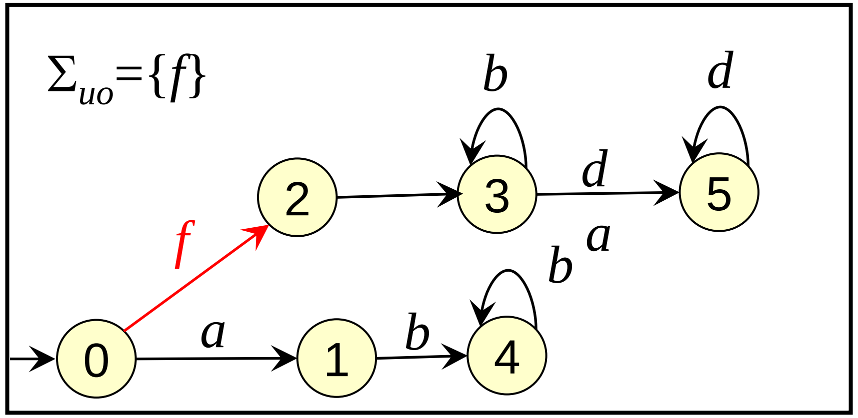
Obs(G)



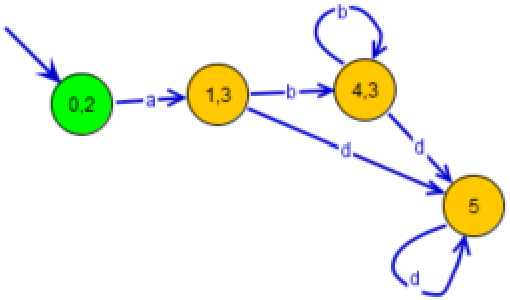


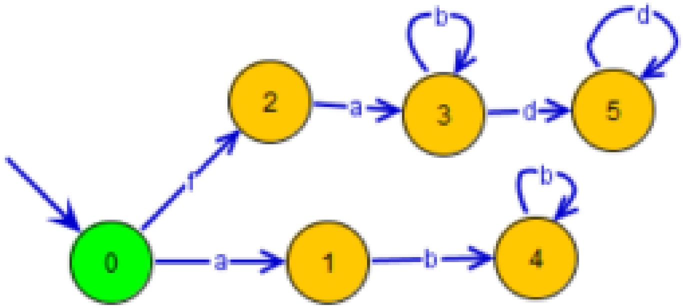
G_d

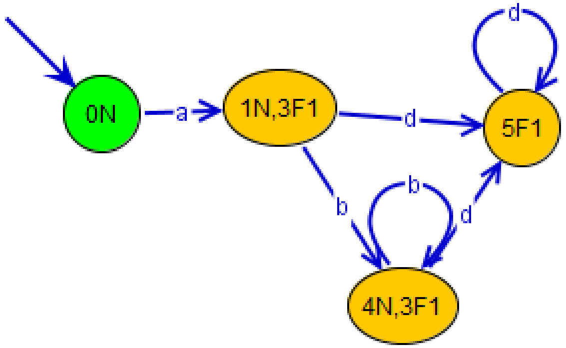
Exercice 2




```
s 0
s 1
s 2
s 3
s 4
s 5
t 0 1 a
t 1 4 b
t 4 4 b
t 0 2 f
t 2 3 a
t 3 5 d
t 3 3 b
t 5 5 d
```







Bibliographie

Références bibliographiques (1)

- Basile, F., Chiacchio, P., & De Tommasi, G. (2012). On K-diagnosability of Petri nets via integer linear programming. Automatica, 48(9), 2047-2058.
- Basilio, J. C., & Toguyéni, A. (2023). Modular diagnosability of Discrete Event Systems Synchronized by Observable or Unobservable Events. IFAC-PapersOnLine, 56(2), 4570-4575.Cabasino, M. P., Giua, A., & Seatzu, C. (2010). Fault detection for discrete event systems using Petri nets with unobservable transitions. Automatica, 46(9), 1531-1539.
- Boussif, A., Ghazel, M., & Basilio, J. C. (2021). Intermittent fault diagnosability of discrete event systems: an overview of automaton-based approaches. Discrete Event Dynamic Systems, 31, 59-102.
- Cabasino, M. P., Giua, A., Lafortune, S., & Seatzu, C. (2012). A new approach for diagnosability analysis of Petri nets using verifier nets. IEEE Transactions on Automatic Control, 57(12), 3104-3117.
- Cassandras, C. G. and Lafortune, S. (2008). *Introduction to discrete event systems.* Springer.
- Chouchane A. and Ghazel M.. Fault-prognosability, K-step prognosis and K-step predictive diagnosis in partially observed petri nets by means of algebraic techniques, Automatica, vol. 162, 2024. https://doi.org/10.1016/j.automatica.2024.111513
- Chouchane A., Ghazel M.,and Boussif A. (2023). K-Diagnosability analysis of bounded and unbounded Petri nets using linear optimization, Automatica, vol. 147, 2023.

Références bibliographiques (2)

- Clavijo, L. B., & Basilio, J. C. (2017). Empirical studies in the size of diagnosers and verifiers for diagnosability analysis. Discrete Event Dynamic Systems, 27, 701-739.
- Contant, O., Lafortune, S., & Teneketzis, D. (2004). Diagnosis of modular discrete event systems. IFAC Proceedings Volumes, 37(18), 327-332.
- de Freitas, B.I. and Basilio, J.C. (2022). Online fault diagnosis of discrete event systems modeled by labeled Petri nets using labeled priority Petri nets. IFACPapersOnLine, 55(28), 329–336.
- Debouk, R., Lafortune, S., & Teneketzis, D. (2000). Coordinated decentralized protocols for failure diagnosis of discrete event systems. Discrete event dynamic systems, 10(1-2), 33-86.
- Genc, S., & Lafortune, S. (2009). Predictability of event occurrences in partially-observed discrete-event systems. Automatica, 45(2), 301-311.
- S. Genc and S. Lafortune. “Distributed Diagnosis of Discrete-Event Systems Using Petri Nets”. In: International Conference on Application and Theory of Petri Nets. 2003, pp. 316–336 (pages 117, 118, 121, 196).
- Hadjicostis, C.N. (2021). Opacity of Discrete Event Systems. In: Baillieul, J., Samad, T. (eds) Encyclopedia of Systems and Control. Springer, Cham. https://doi.org/10.1007/978-3-030-44184-5_100093
- Lafortune, S. (2013). Diagnosis of Discrete Event Systems. In: Baillieul, J., Samad, T. (eds) Encyclopedia of Systems and Control. Springer, London. https://doi.org/10.1007/978-1-4471-5102-9_56-1
- Lafortune, S., Lin, F., & Hadjicostis, C. N. (2018). On the history of diagnosability and opacity in discrete event systems. Annual Reviews in Control, 45, 257-266.

Références bibliographiques (2)

- Laprie J.-C. (1995) . Guide de la sûreté de fonctionnement, Cépaduès, 369 pages, Toulouse, mai 1995,
- Li, B., Basilio, J. C., Khlif-Bouassida, M., & Toguyéni, A. (2017). Polynomial time verification of modular diagnosability of discrete event systems. IFAC-PapersOnLine, 50(1), 13618-13623.
- Liu, B. (2014). An Efficient Approach for Diagnosability and Diagnosis of DES Based on Labeled Petri Nets, Untimed and Timed Contexts (Doctoral dissertation, Ecole Centrale de Lille).
- Moreira, M. V., Jesus, T. C., & Basilio, J. C. (2011). Polynomial time verification of decentralized diagnosability of discrete event systems. IEEE Transactions on Automatic Control, 56(7), 1679-1684
- Saboori, A., & Hadjicostis, C. N. (2007, December). Notions of security and opacity in discrete event systems. In 2007 46th IEEE Conference on Decision and Control (pp. 5056-5061). IEEE.
- Zaytoon J., On Fault Diagnosis Methods of Discrete Event Systems, Plenière https://unidad.gdl.cinvestav.mx/wodes-12/downloads/slides/plenaryProfZaytoon.pdf
- Zwingelstein G (1995). Diagnostic des défaillances : théorie et pratique pour les systèmes industriels, Edition Hermès, Traité des Nouvelles Technologies “Série Diagnostic et Maintenance”, 601 pages, Paris, 1995,

Equipe pédagogique et technique

Auteurs : Mohamed Ghazel, Dimitri Lefevre, Hervé Marchand, Ramla Saddem, Armand Toguyéni

Intervenants : Ramla Saddem, Armand Toguyéni