Raisonnement diagnostic sur les systèmes à événements discrets

Yannick Pencolé

12 juin 2012

Plan du cours

1. Introduction
2. Systèmes à événements discrets (SED)
3. Formalisme de modélistion pour SED
4. Diagnostic de SED
   - Approche Diagnostiqueur
5. Architecture de Diagnostic
   - Diagnostic coordonné
   - Diagnostic décentralisé
   - Diagnostic distribué

Concept de diagnostic

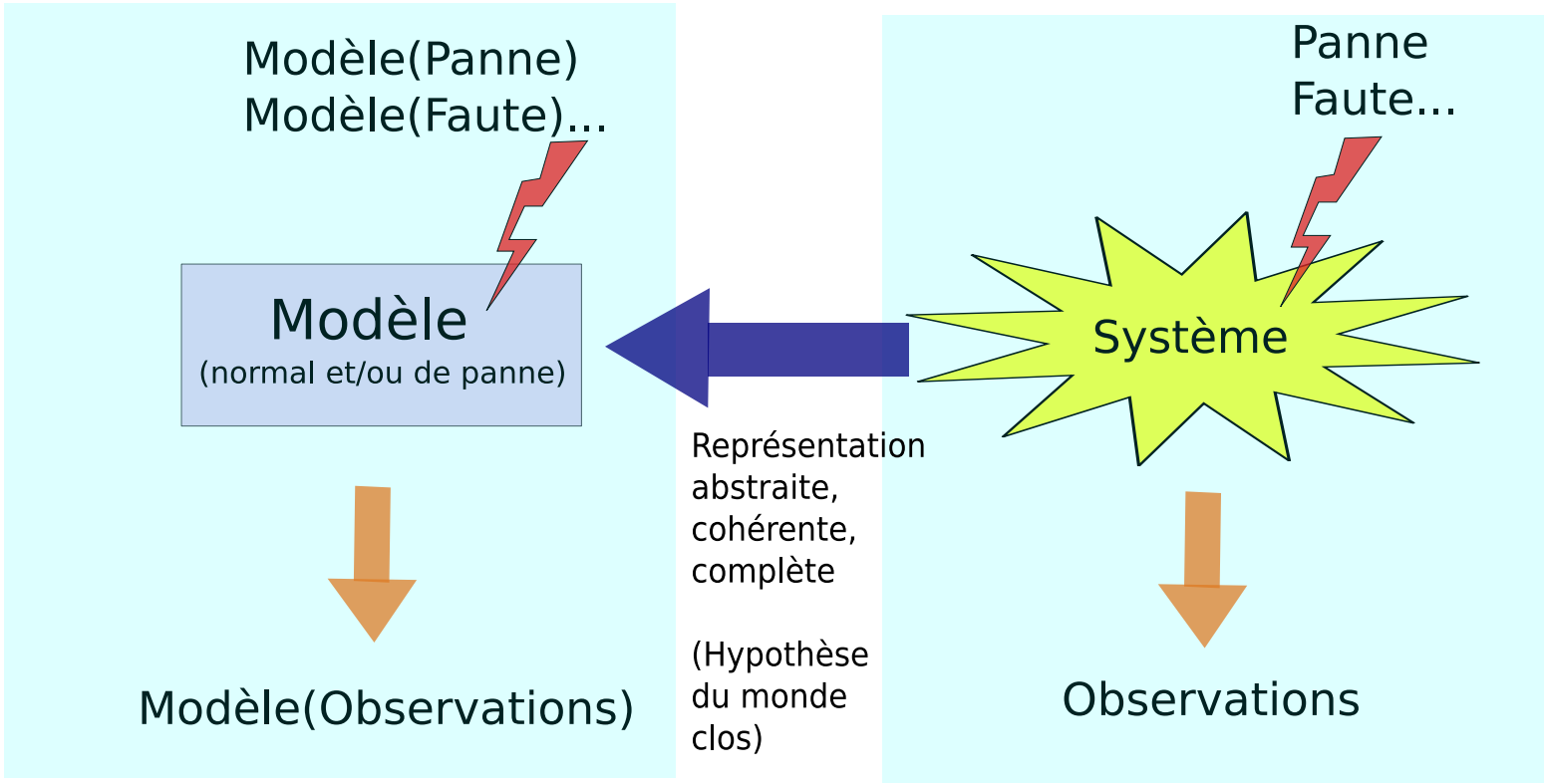
VOIR **COMPRENDRE** **DÉCIDER**

Acquérir les observations → **Détecter** un problème / **Identifier** la panne → **Modifier** le système

Infos nécessaires à un bon diagnostic

Infos nécessaires à une bonne décision

Concept du diagnostic à base de modèle



Diagnostic à base de modèle =

1) Confronter les observations au modèle (cohérence)
2) "Remonter" à la cause (abduction, modèle de panne)

Classification des systèmes et modèles DYNAMIQUES étudiés en diagnostic

- **Continu**
  - Modèle en terme de mode
  - Variables continues, valeurs réelles, temps continu
  - Équations différentielles
  - Modèles numériques : Automatique
  - Modèles qualitatifs (ou semi-quantitatifs)
- **Discret**
  - Variables discrètes et temps discret
  - Notion d' événement
- **Hybride**
  - Mode (continus), Changement de modes (discret)

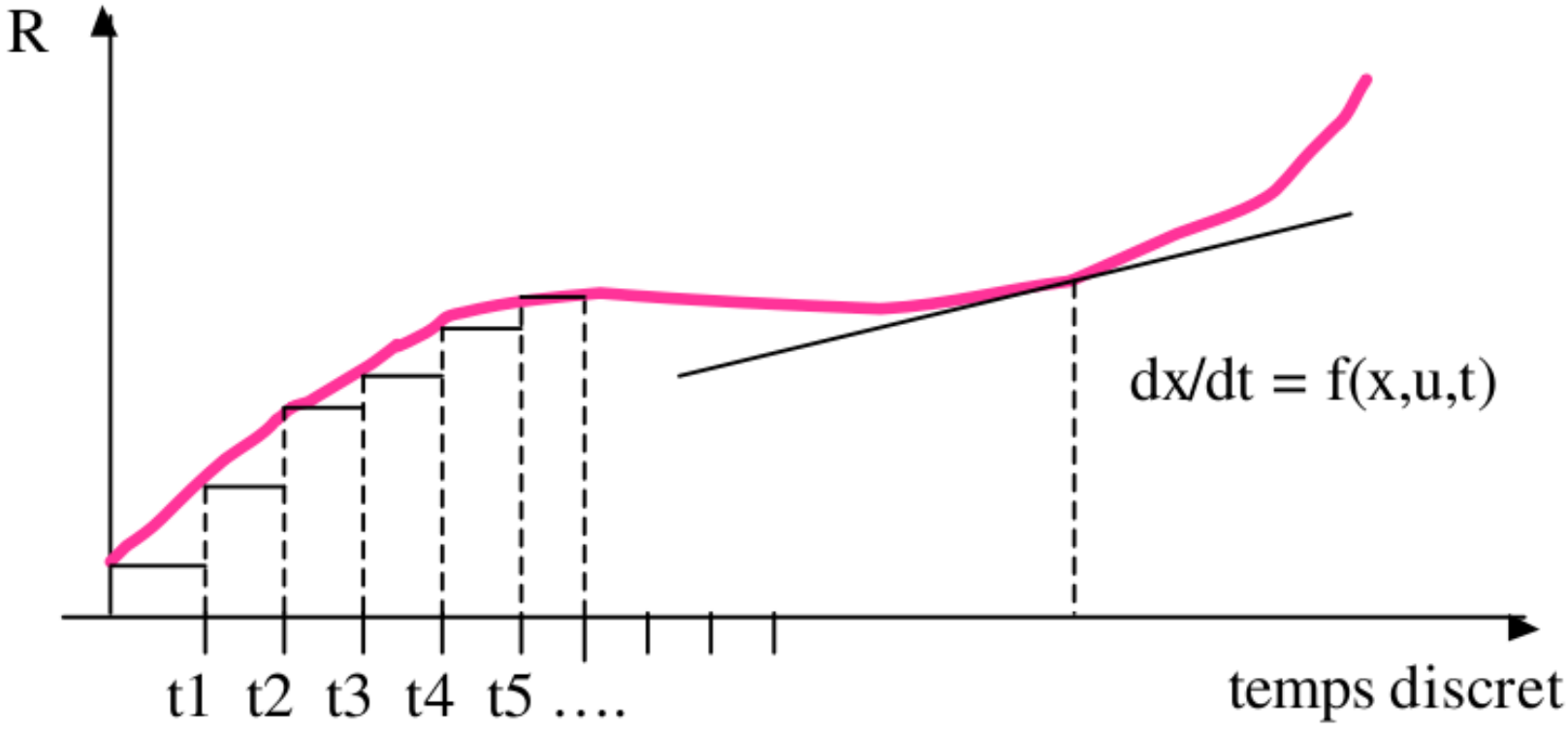
Pourquoi étudier les systèmes à événements discrets ?

- Engouement né de la complexification des systèmes à aborder
- Apparition des ordinateurs et du traitement informatique (numérique)
- Les SED sont en générale des conceptions humaines
  - réseaux de transport,
  - reseaux informatiques,
  - ordinateurs, calculateurs,
  - protocoles manufacturiers,
  - gestion d'entreprises, procédures...
- Utile aussi lorsqu'on peut discretiser un système quelconque (modélisation qualitative)

Temps discret vs Etat discret

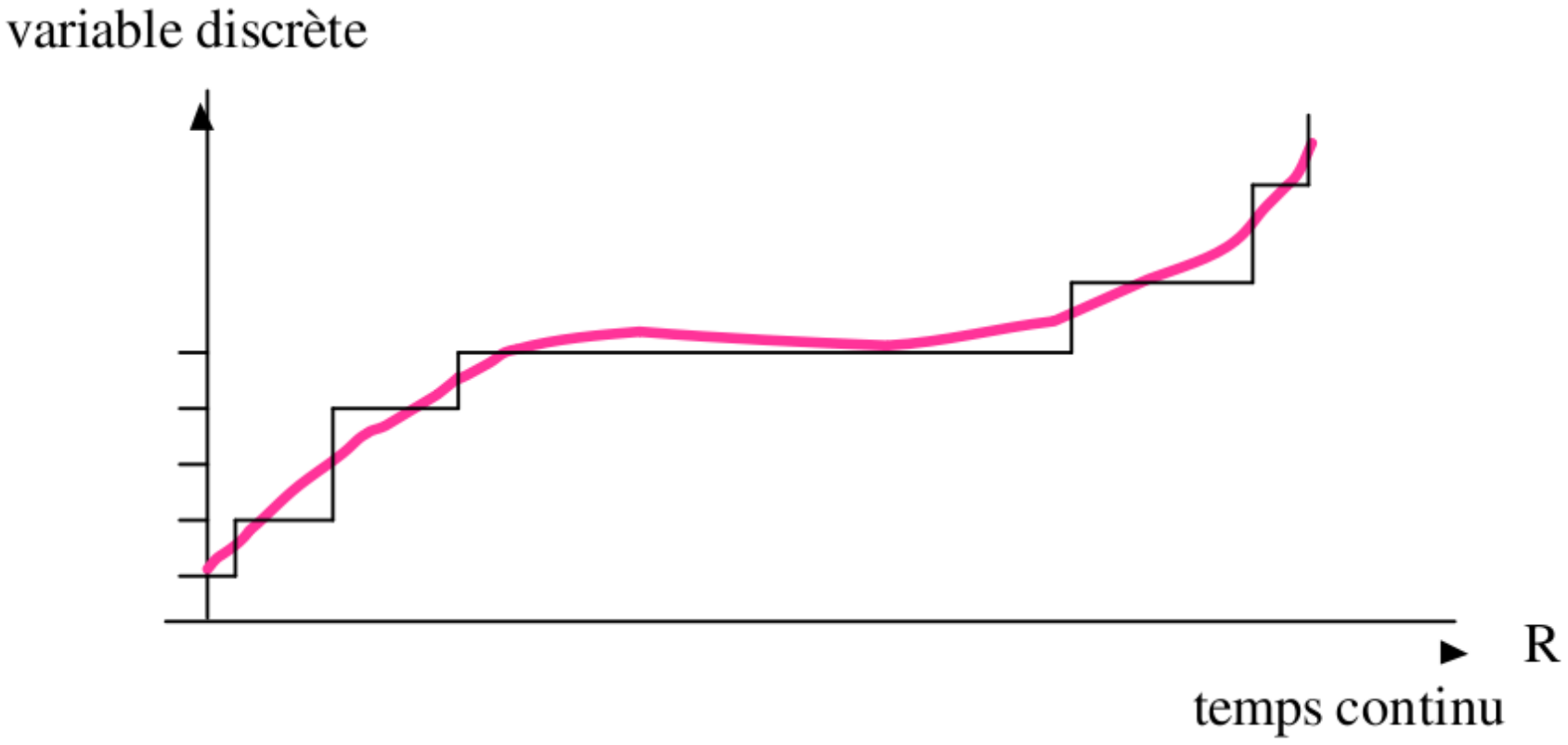
- 2 types de discrétisations
- Discretisation du temps
  - échantillonage d'un signal continu par exemple
- Discrétisation de l'état
  - Abstraction d'états en classe discrète d'états

Temps discret



Temps = ensemble d'instant (horloge)
Etat = continu

État discret



Temps = à valeur continue
Etat = discret

Dynamiques temporelle vs événementielle

- Dynamique temporelle
  - à chaque tic d'horloge, on regarde ce qui se passe
  - modèle *time-driven*
  - Synchronisation des composants par l'horloge
  - Dynamique synchrone
- Dynamique événementielle
  - C'est l'occurrence d'un événement sur le système qui sert d'horloge
  - Modèle *event-driven*
  - Dynamique asynchrone (échange de messages, rendez-vous)

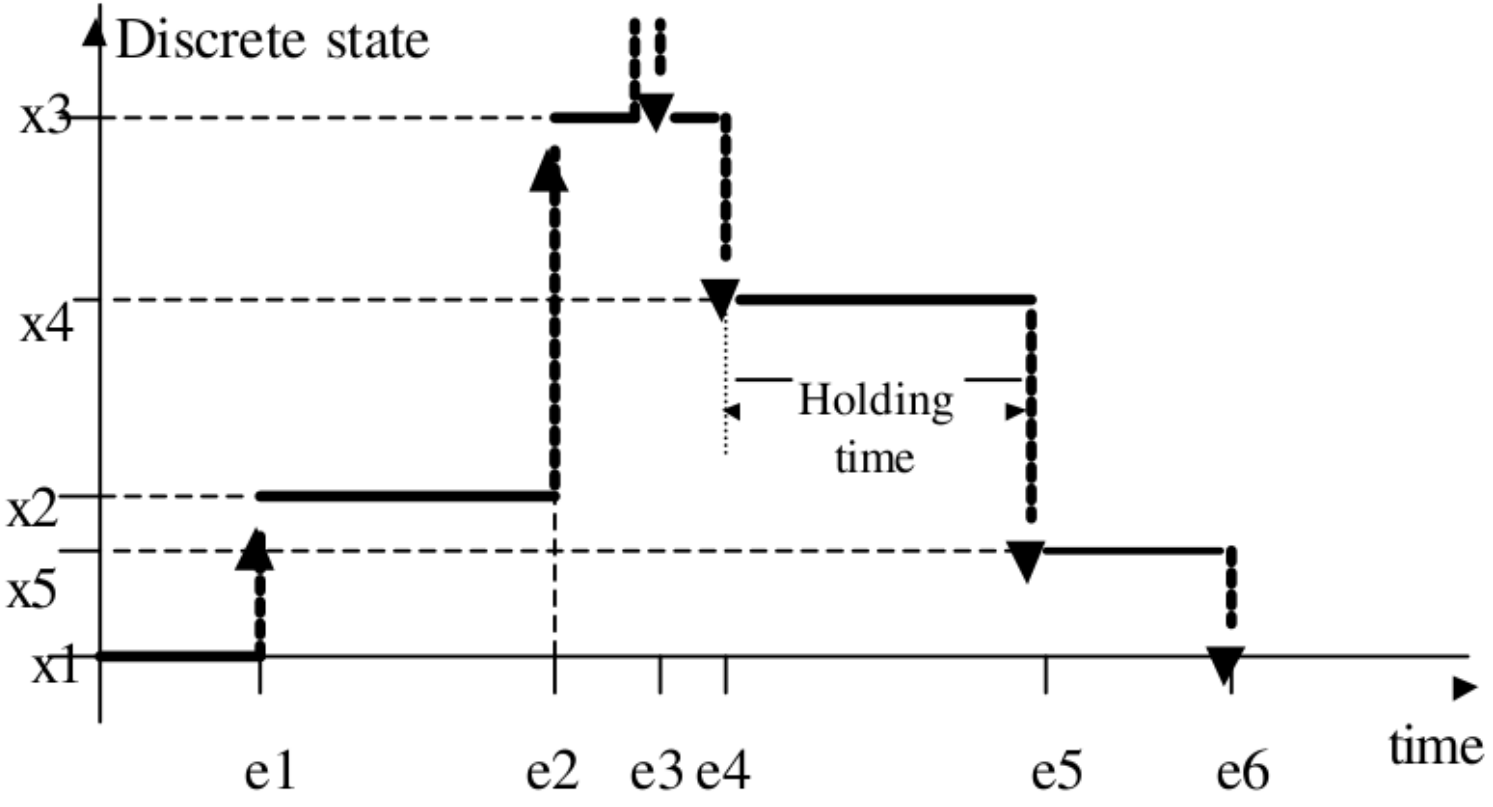
Concept d'événements

- Caractéristiques
  - instantané
  - provoque (éventuellement) un changement d'état
  - contrôlable (action) ou spontané (panne)
  - exogène ou endogène

Exemple

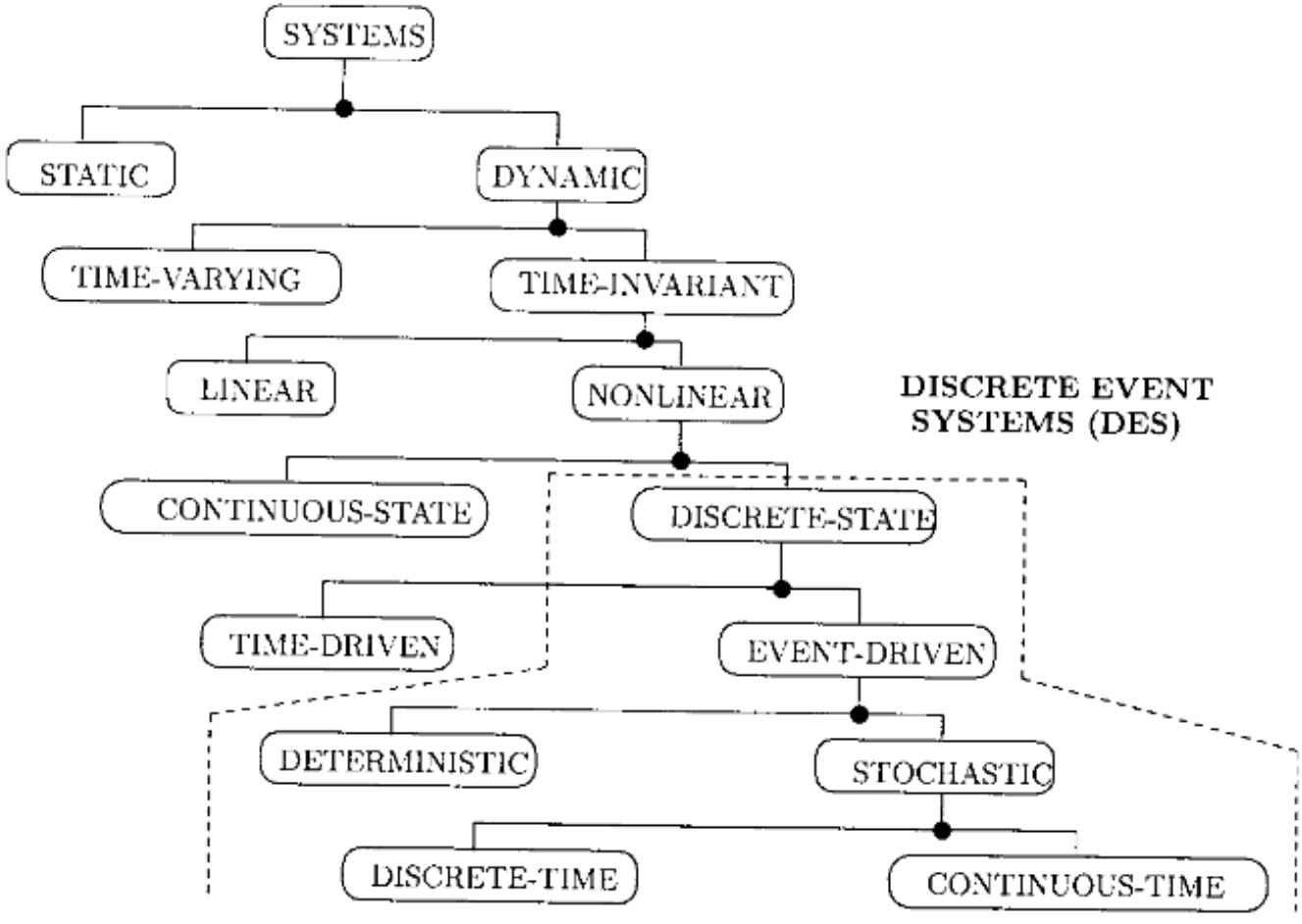
- Une personne presse un bouton. Elle ouvre une vanne.
- La machine se réinitialise. Court-circuit.
- Le niveau maximal de liquide dans le récipient est atteint.

Caractéristiques des SED



Etat discret, dynamique événementielle

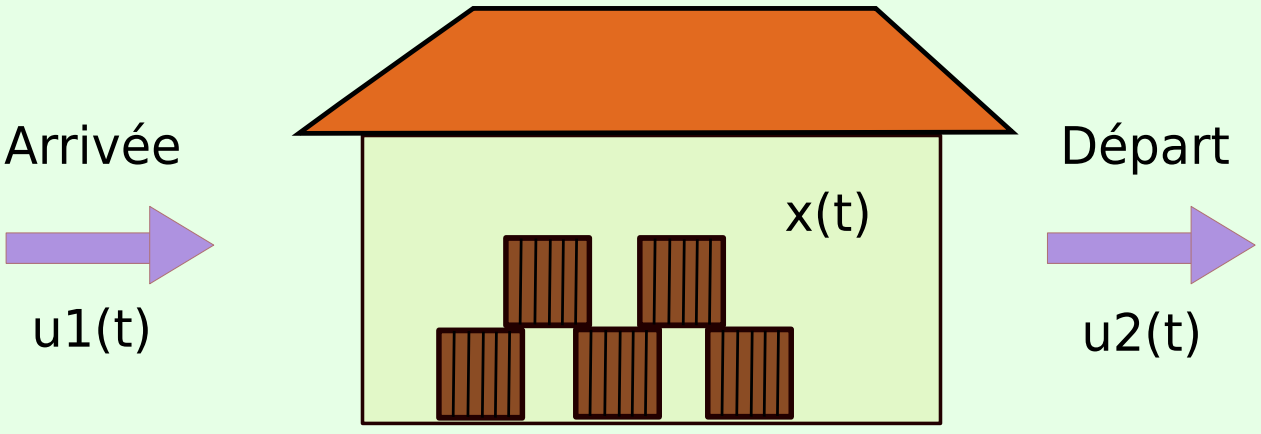
Classification des systèmes



Système discret par nature (1)

Exemple

Entrepôt

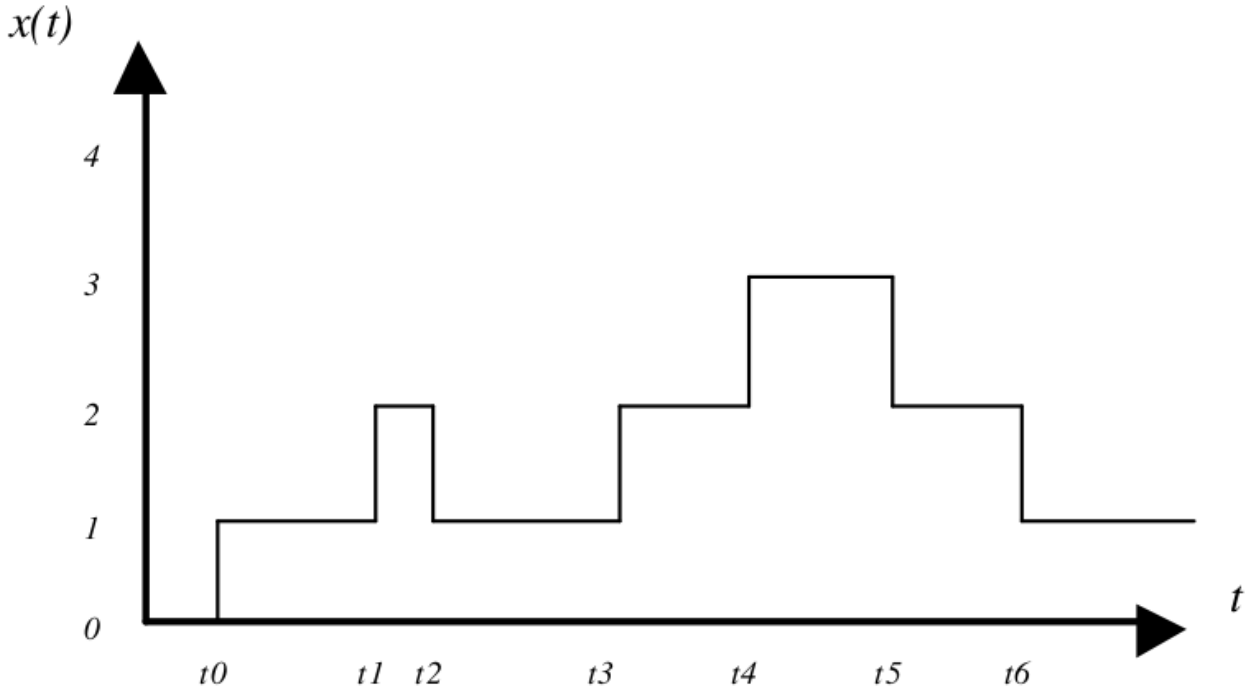


- $x(t)$ nombre de caisses dans l'entrepôt à l'instant t
- $u1(t) = 1$ si une caisse arrive au temps t, 0 sinon
- $u2(t) = 1$ si un caisse part au temps t, 0 sinon

Système discret par nature (2)

Exemple

Si u1(t) alors x(t) = x(t-1) +1
Si u2(t) alors x(t) = x(t-1) -1



Système discrétisable (1)

- Système continu mais modèle discret
  - Selon la tâche à effectuer sur le système
  - Modèle qualitatif

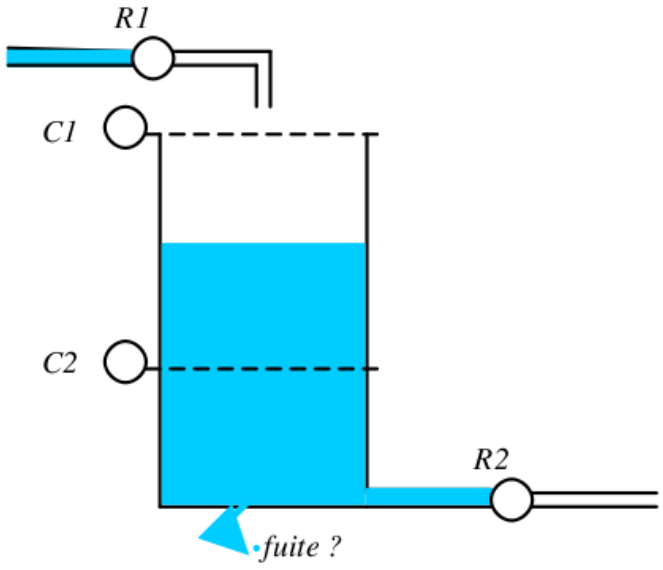
Exemple

Le réservoir

Système discrétisable (2)

Exemple

- Récipient



R1 : {ouvert, fermé}
R2 : {ouvert,fermé}
C1 : {niv_atteint,niv_pas_atteint}
C2 : {niv_atteint,niv_pas_atteint}

Espace d'états discrets :
R1 x R2 x C1 x C2

Exemples d'événements :
Ouverture de R1
Fermeture de R2
Le niveau passe au dessous de C2
Ca déborde!!

- Détection d'une fuite :
  - *R1=fermé, R2=fermé, C2=niv_atteint*
  - *événement : « le niveau devient inférieur à C2 »*

Choix du formalisme de modélisation

- Avant de modéliser :
  1. quelle est la nature de mon système ?
     - monolithique, distribué, continu, discret
  2. quel est l'objectif du diagnostic ?
     - détection, localisation, identification,...
  3. quel est le degré de connaissance de mon système ?
     - connaissance de surface, comportement nominal, comportement de panne
- Choix du formalisme de modélisation
  - Dépend de la réponse aux questions précédentes
- De nombreux formalismes sont possibles
  - Algèbre de processus, Système de transitions (automate), Langage, Règles, Réseau de Petri

Système de transition

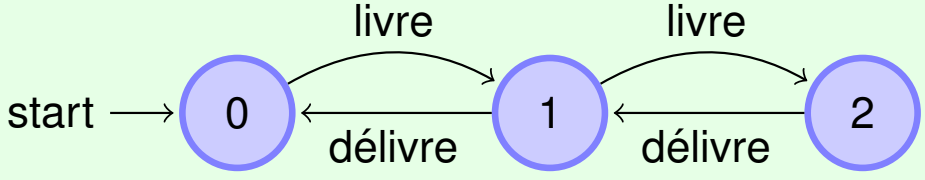
Définition

Système de transition : $G = (X, E, f, x_0)$

- X : ensemble d'états
- E : ensemble des événements
- f : fonction de transition $X \times E \rightarrow 2^X$
- x_0 : état initial

Exemple

Un entrepôt contenant deux caisses au maximum.



Système de transition : modélisation compositionnelle

- Système : ensemble de composants
- Modélisation monolithique d'un système
  1. compliqué (recensement de tous les comportements possibles)
  2. souvent irréaliste (trop nombreux, même pas codable)
  3. ne met pas à profit le fait qu'un système est un ensemble de composants
- Modélisation compositionnelle
  - Modèle du comportement de chaque composant (plus simple, moins gros)
  - Spécification d'une loi de composition des modèles
  - Modèle du système $\equiv$ Modèles des composants $+$ Loi de composition
- Loi de composition
  - Fondé en général sur un produit de système de transitions
  - Fonction de règles de synchronisation entre événements

Loi de composition (1)

Soit $G_1, \ldots, G_n$ avec $G_i = (X_i, E_i, f_i, x_{0i})$ les modèles locaux, soit ε l'événement vide tel que $\forall i, f_i(x, \varepsilon) = \{x\}$, soit $Sync \subseteq (E_1 \cup \{\varepsilon\}) \times \cdots \times (E_n \cup \{\varepsilon\})$ la relation de synchronisation.

Définition

Le produit synchronisé $G = G_1 ||_{Sync} \ldots ||_{Sync} G_n$ est le système de transition : $G = (X, E, f, x_0)$

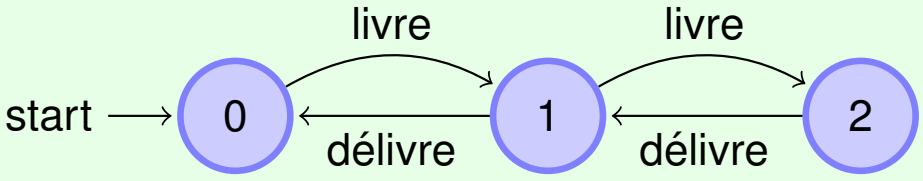
- $X \subseteq X_1 \times \cdots \times X_n$ (ensemble des états accessibles)
- $E = Sync \subseteq (E_1 \cup \{\varepsilon\}) \times \cdots \times (E_n \cup \{\varepsilon\})$
- $x_0 = (x_{01}, \ldots, x_{0n})$
- f fonction de transition :

$$f((x_1, \ldots, x_n), (e_1, \ldots, e_n)) = \{f_1(x_1, e_1) \times \cdots \times f_n(x_n, e_n)\}$$

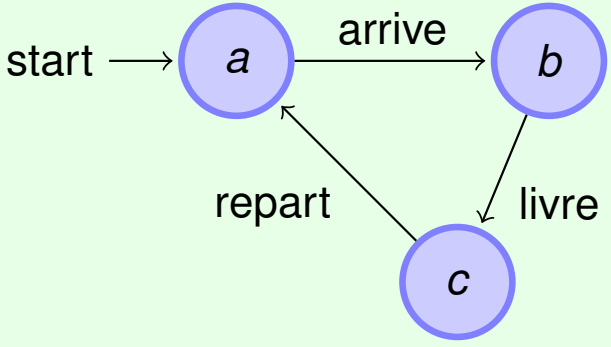
Loi de composition (2)

Exemple

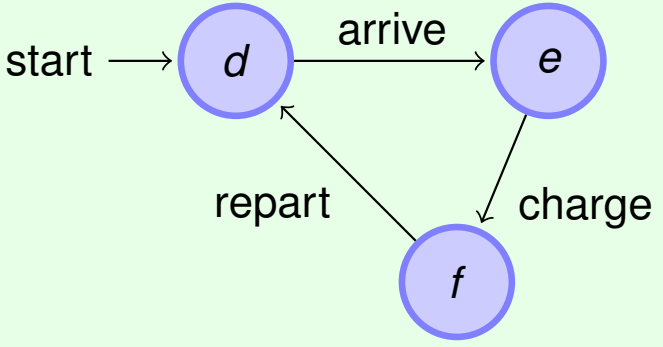
Entrepôt :



Livreur :



Client :

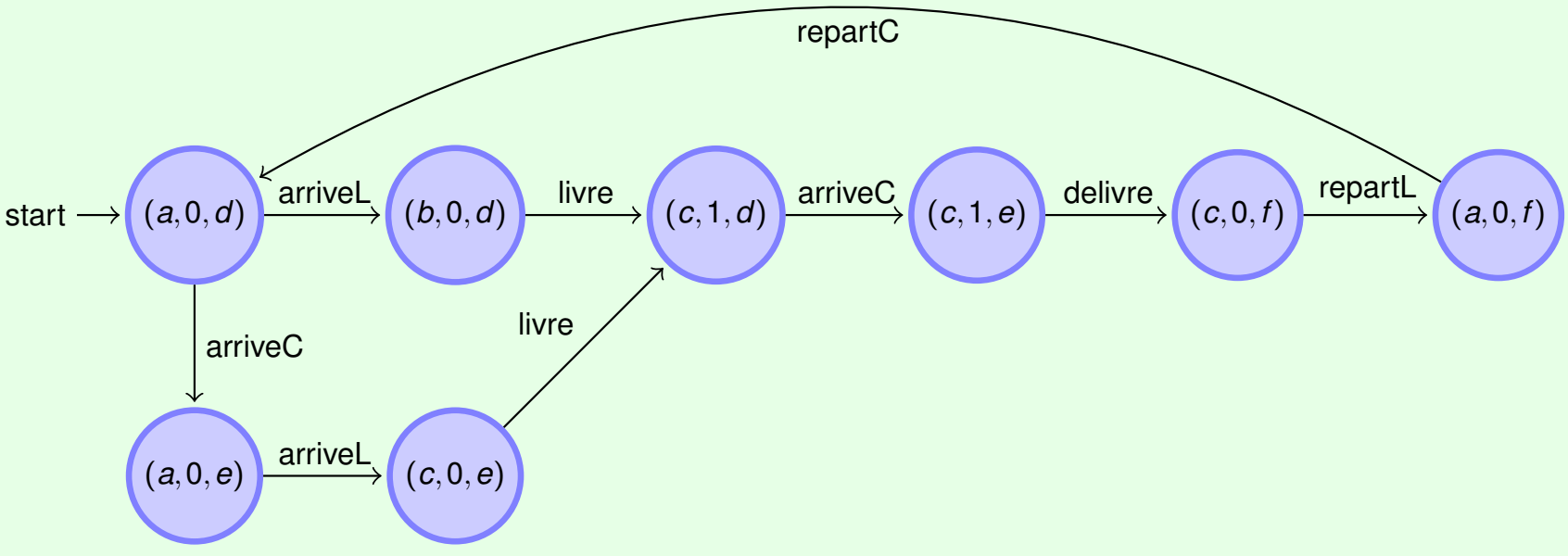


Relation de

synchronisation *Sync* : $arriveL = (arrive, \varepsilon, \varepsilon)$, $repartL = (repart, \varepsilon, \varepsilon)$, $arriveC = (\varepsilon, \varepsilon, arrive)$, $repartC = (\varepsilon, \varepsilon, repart)$, $livre = (livre, livre, \varepsilon)$, $delivre = (\varepsilon, delivre, charge)$.

Loi de composition (3)

Exemple

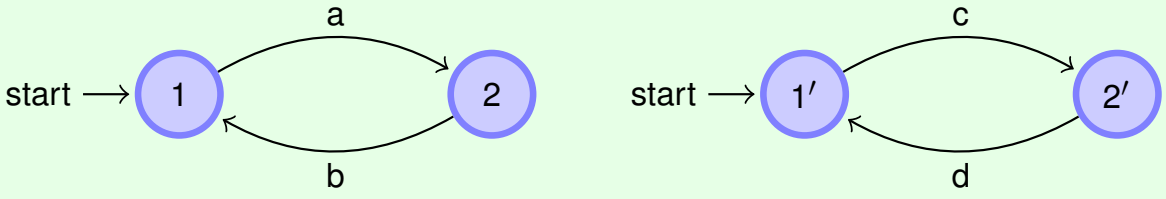


Différentes lois de composition

- Fonction de *Sync*
  1. $\forall (e_1, \ldots, e_n) \in Sync, e_i \neq \varepsilon \Rightarrow$ produit synchrone (tous les composants évoluent en même temps)
  2. $\forall (e_1, \ldots, e_n) \in Sync, \exists! e_i \neq \varepsilon \Rightarrow$ produit libre (tous les composants évoluent indépendamment les uns des autres)
  3. Tout ensemble *Sync* définit un produit synchronisé particulier
- Modèle global $G = G_1 ||_{Sync} \ldots ||_{Sync} G_n$
  - Au pire cas (et en pratique), taille exponentielle en le nombre de composants
  - Explosion combinatoire
  - Problème de calcul

A vous de jouer

Exemple



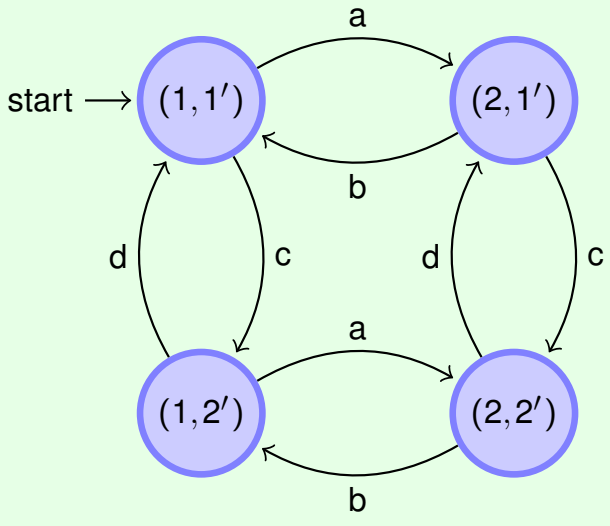
Ecrire le produit libre : $Sync = \{(a,\varepsilon),(b,\varepsilon),(\varepsilon,c),(\varepsilon,d)\}$

Ecrire le produit synchrone : $Sync = \{(a,c),(b,d)\}$

Ecrire le produit synchronisé : $Sync = \{(a,\varepsilon),(b,c),(\varepsilon,d)\}$

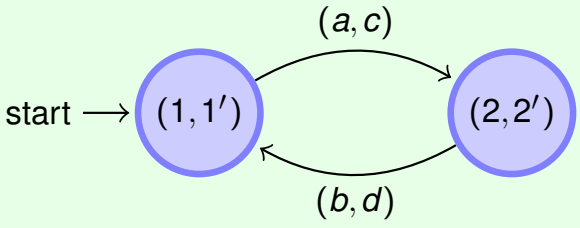
Résultat du jeu : produit libre

Exemple



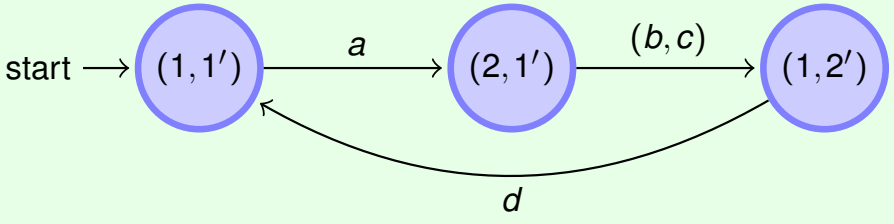
Résultat du jeu : produit synchrone

Exemple



Résultat du jeu : produit synchronisé

Exemple



Réseau de Petri

Définition

Réseau de Petri : $R = (P, T, Pré, Post)$

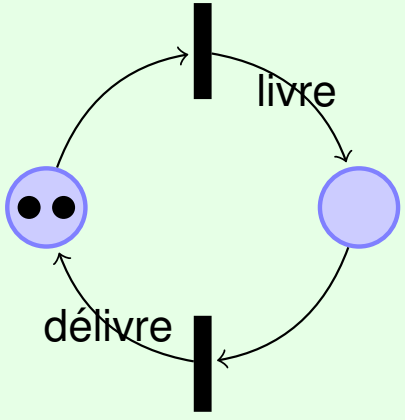
- P : ensemble de places
- T : ensemble de transitions
- $Pré$: fonction incidence arrière $Pré : P \times T \rightarrow \mathbb{N}$
- $Post$: fonction incidence avant $Post : P \times T \rightarrow \mathbb{N}$

Réseau de Petri marqué : (R, M_0) où M_o est le marquage initial.
Fonction de marquage $M : P \rightarrow \mathbb{N}$

Réseau de Petri (2)

Exemple

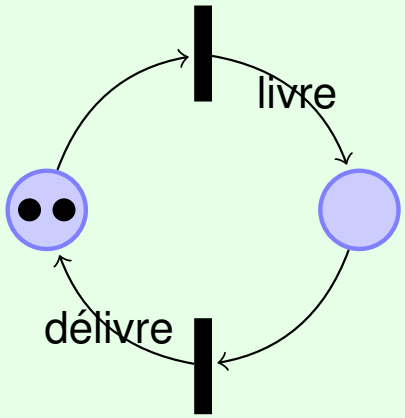
Un entrepôt contenant deux caisses au maximum.

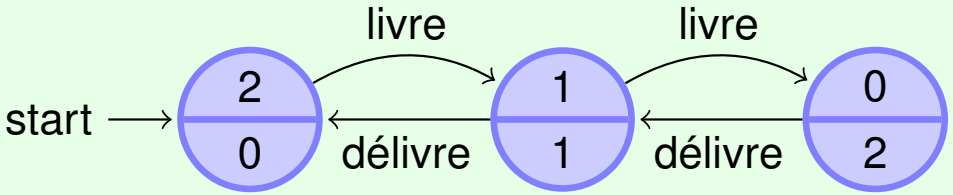


Graphe des marquages accessibles

- Etant donné un réseau de Petri marqué
- Par simulation exhaustive, on peut obtenir le graphe des marquages accessibles
- Graphe des marquages $\equiv$ Système de transition

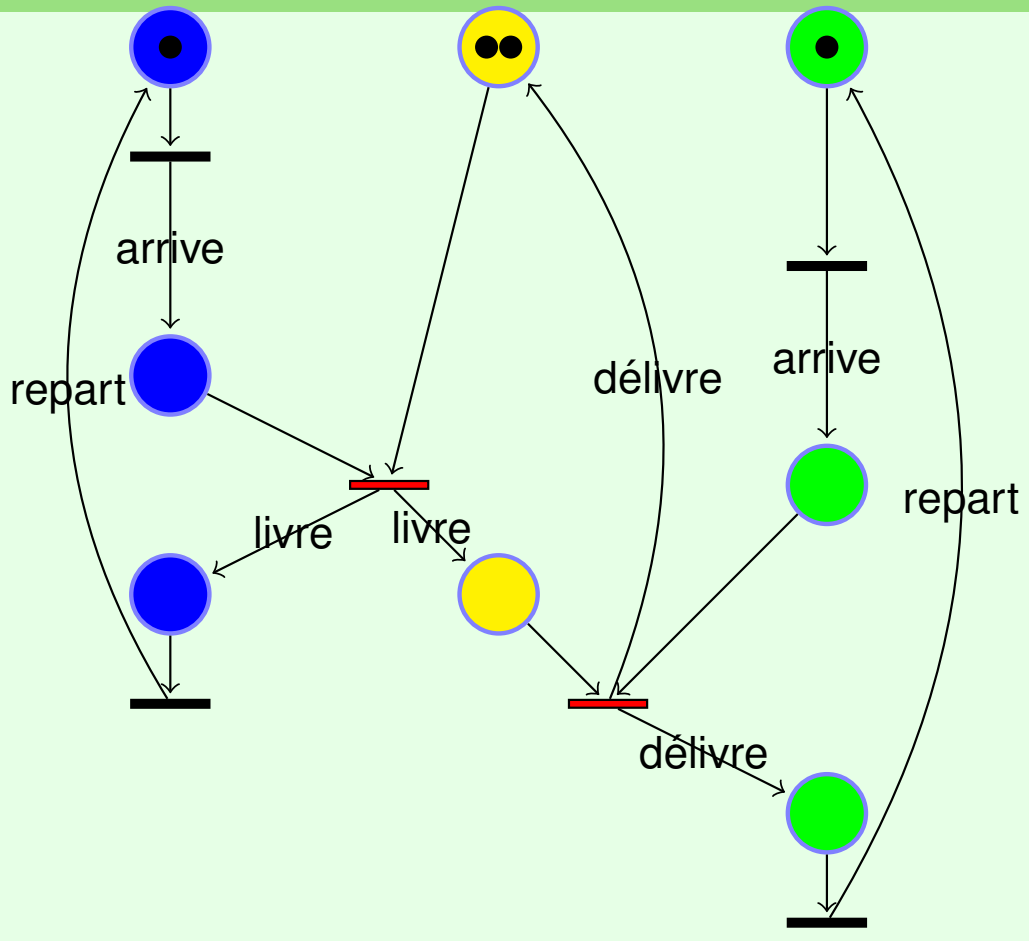
Exemple





Livreur+Entrepôt+Client en RDP

Exemple

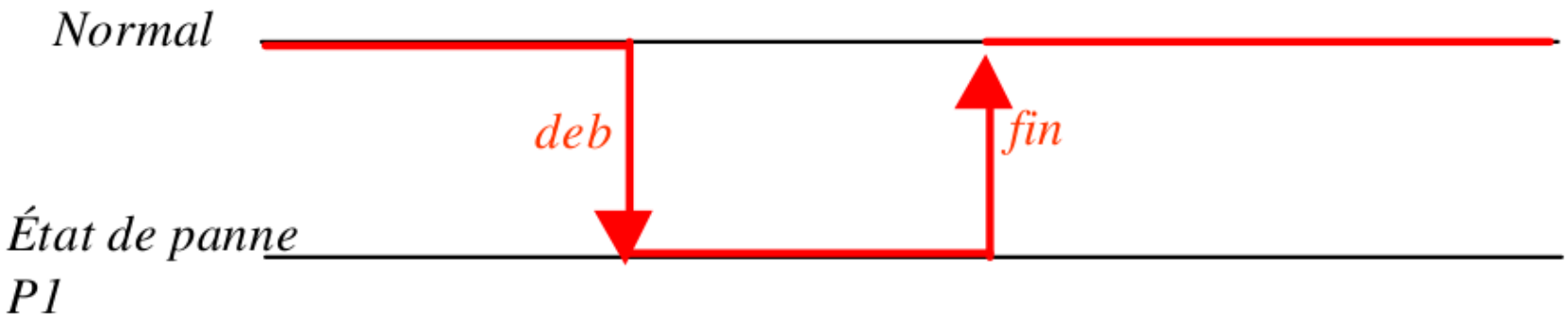


RDP vs Modélisation compositionnelle

- Tout d'abord et avant tout, c'est une affaire de goût (éducation, culture)
- Modélisation compositionnelle
  1. Plus intuitif (notion explicite de composants)
  2. Plus simple à définir pour les non-spécialistes (notion de message et de ports)
  3. Pas terrible pour exprimer la concurrence entre composants
  4. Sémantique compositionelle peu maitrisée
- Réseau de Petri
  1. Expression aisée de la concurrence
  2. Sémantique de synchronisation connue et maitrisée (elle est exprimée dans le réseau même)
  3. Pas très intuitif pour un néophite
  4. L'aspect composant n'est pas si clair

Notion de panne, de faute

- Pannes, Fautes :
  - Événements non-observables
    1. *début de panne*
    2. éventuellement *fin de panne*

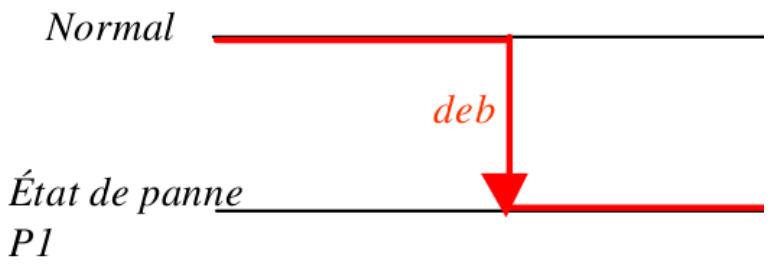


- Terminologie : panne ? faute ? Grand débat franco-français. (et même toulouso-toulousains)
- Dépend de l'application et de son propre champ terminologique

Panne permanente

Définition

Une panne est permanente s'il n'existe aucun moyen de la réparer ou aucune raison qu'elle disparaisse pendant la période de diagnostic.



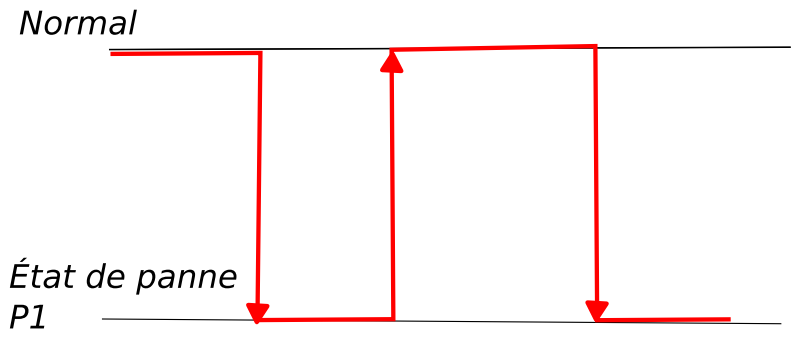
Exemple

Diagnostic en-ligne d'une automobile : une panne moteur est permanente. Cette panne ne peut pas disparaître sans une intervention extérieure (garage)

Panne intermittente

Définition

Une panne est intermittente si elle peut apparaître/disparaître pendant la période de diagnostic à plusieurs reprises.



Exemple

Diagnostic en-ligne d'une automobile : un mauvais contact filaire entre calculateurs peut apparaître et disparaître (vibration, humidité) au cours d'un même trajet.

Modèle de diagnostic

Soit *M* le modèle SED de diagnostic (ou son langage), il contient toujours :

- Des événements observables *O*

Il contient parfois :

- Des événements de pannes, de fautes (modèle de panne, de faute, de dysfonctionnement) *F*
  - Un tel événement est en général non observable (sinon problème trivial)
- Des évenements de communication (modèles de composants interagissants) *C*

Problème général du diagnostic à base de modèle de SED

- Étant donné un modèle *M*
- Étant donné des événements observables
  - Flux d'observations arrivant en-ligne (ordre total ou non)
- Quels sont les événements non-observables qui expliquent les observations selon le modèle ?

Objectif de diagnostic

Fonction des connaissances, le diagnostic a plusieurs sous-objectifs :

1. Détection : détecter la présence d'une panne
2. Localisation : identifier le composant où la panne s'est produite
3. Identification : identifier la nature, le type de la panne
4. Propagation : déterminer toutes les conséquences de la panne (relation cause-effet)

Problème de diagnostic : détection

Soit σ une séquence d'observations, soit M un modèle nominal de diagnostic

Définition

Diagnostic par cohérence $\equiv$ détection :
si $P_O(M) \cap \{\sigma\}$ est vide alors le système ne se comporte pas comme prévu $\Rightarrow$ détection d'un problème.

Exemple

Système Livreur+Entrepôt+Client, $O = \{arriveC, arriveL\}$
Si $\sigma = arriveC\ arriveC$ alors problème (le client aurait dû attendre l'arrivé d'un livreur)
Si $\sigma = arriveC\ arriveL\ arriveC$ on reste cohérent : $P_O(M) \cap \{\sigma\} = \sigma$

Problème de diagnostic : localisation

Soit M un modèle nominal à base de composant $C_1, \ldots, C_n$

Définition

Localisation :
Si $P_{O,C}(M) \cap \{P_C(\sigma)\}$ est vide alors le composant C est un diagnostic candidat.

Exemple

Système Livreur+Entrepôt+Client, $O = \{arriveC, charge\}$
Si $\sigma = arriveC\ arriveC$ alors problème possible chez le client.

Problème de diagnostic : identification et propagation

Soit M un modèle de panne identifiant les pannes $f_1, \ldots, f_n$ et leurs conséquences

Définition

Propagation : ensemble de trajectoires expliquant σ

$$H = M||_O\{\sigma\}$$

Identification :

$$I = P_{Panne}(H)$$

Diagnostic en-ligne vs diagnostic hors-ligne

- Diagnostic hors-ligne
  - On connaît toutes les observations *a priori*
  - Pas de contraintes sur le temps de réponse
  - En général on ne traite que les pannes permanentes

- Diagnostic en-ligne
  - Les observations sont acquises en ligne
    - Suivi, supervision, surveillance
  - Évolution du diagnostic en fonction de nouvelles observations
  - Temps de réponse contraint

Diagnostiqueur de Sampath

- Travaux publiés en 94-95-96
  - Meera Sampath et Stéphane Lafortune (Université du Michigan)
  - Parmi les initiateurs pour la caractérisation du problème de diagnostic dans les SED
- Objectifs : identification de l'occurrence d'événements de faute pouvant expliquer une séquence d'observations
- Modèle : proche de la théorie de la contrôlabilité [Ramadge et Wonham 89]
  - Ensemble d'automates synchronisés

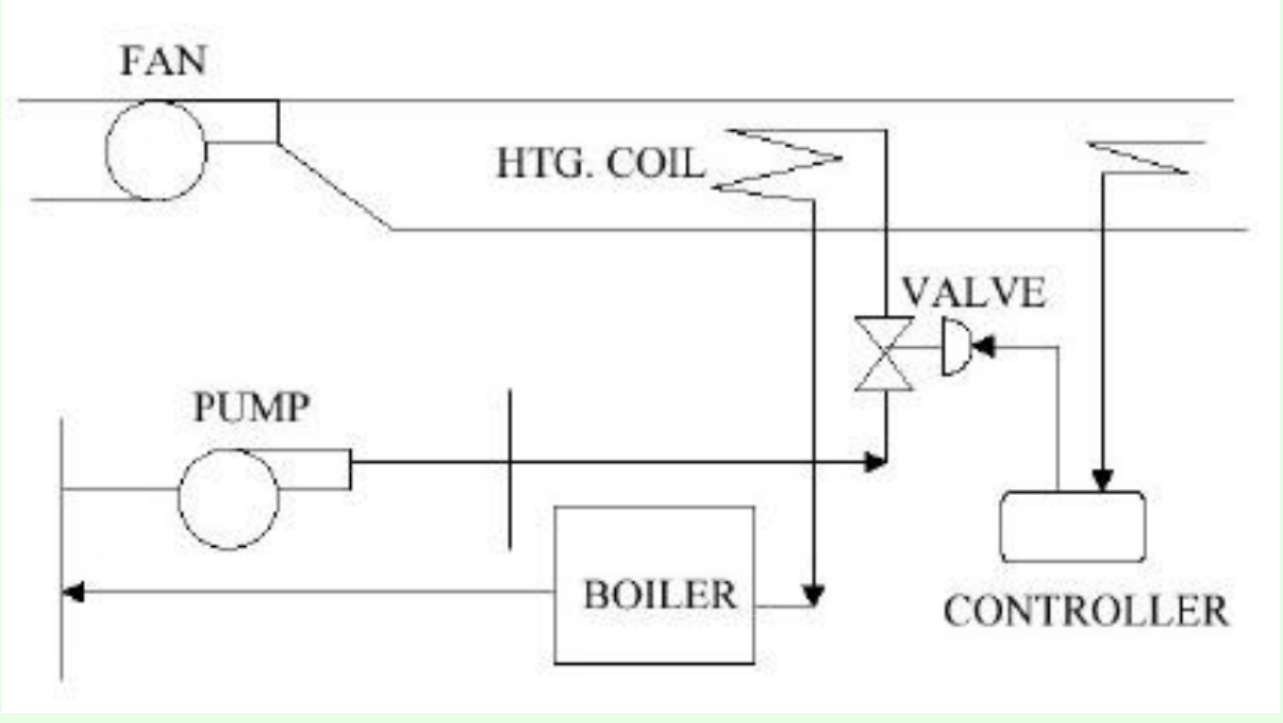
Système d'air conditionné

Exemple

Stéphane Lafortune
Dept of EECS, Detroit
Meera Sampath
Wilson Center for Research & Technology

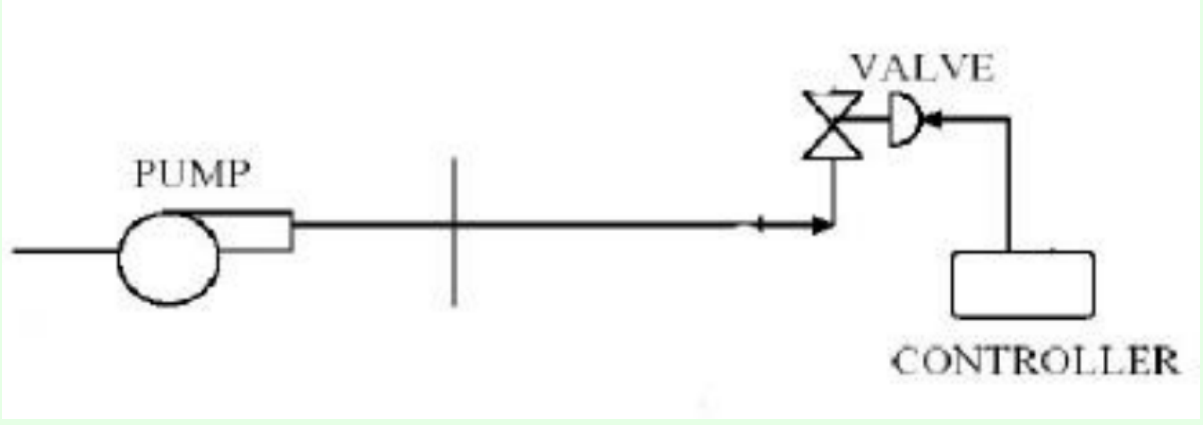
Système d'air conditionné (2)

Exemple



Système d'air conditionné (2)

Exemple



Système d'air conditionné (3)

Exemple

Évenements de pannes permanentes :

- Valve bloquée ouverte : événement spontané **bo**
- Valve bloquée fermée : événement spontané **bf**

Événements observables :

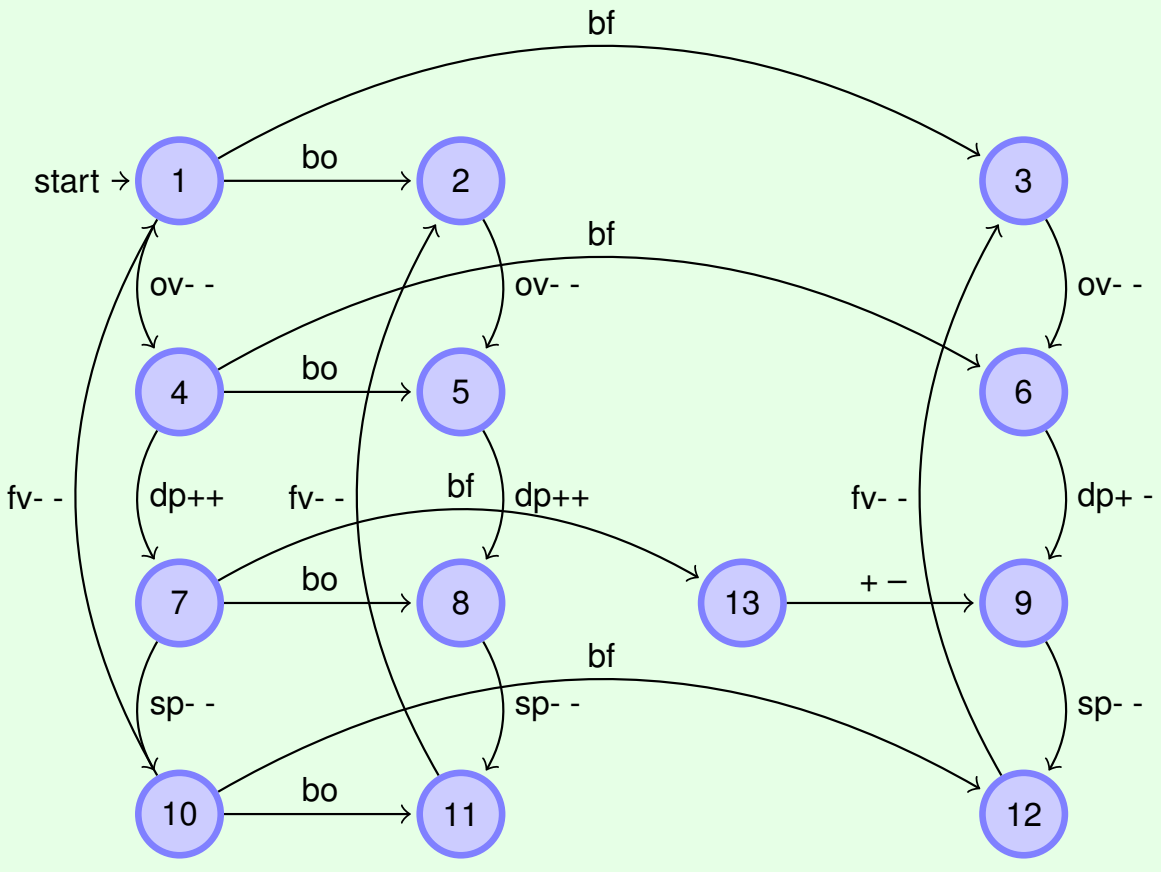
- Actions
  - **ov** : ouvre la valve, **fv** : ferme la valve
  - **dp** : démarre la pompe, **sp** : stoppe la pompe
- Observations d'un capteur de débit
  - **++** : présence d'un débit, **- -** : absence d'un débit
  - **+ -** : coupure spontanée du débit

Dans la suite, événements composés : **ov- -** signifie "ouverture de la valve et absence de débit"

Système d'air conditionné : Modèle global

Exemple

bo : bloqué ouverte
bf : bloqué fermé
ov : ouvre valve
fv : ferme valve
dp : démarrre pompe
fp : ferme pompe
++ : débit **-**
- : pas débit
+- : plus aucun débit



Principe du diagnostic

Exemple

1. On suppose connu l'état initial :

$$\text{État initial} = 1$$

2. On observe une séquence d'observations :

ov- -, dp ++

3. On cherche à établir l'ensemble des états courants du système

$$\text{États courants (observateur)} = \{7, 8\}$$

4. On cherche à établir si des événements de pannes se sont produits

$$\text{États courants (diagnostiqueur)} = \{7\{\}, 8\{bo\}\}$$

Principe du diagnostiqueur

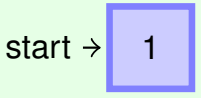
- Algorithme de diagnostic $\equiv$ Recherche de chemins de transitions dans le modèle
- Diagnostiqueur : machine à états finis
  - Précompilation hors-ligne de la recherche de chemins
  - Abstraction des événements non-observables
- Diagnostiqueur $\equiv$ Observateur enrichi
- Machine efficace
  - A chaque nouvelle observation, on tire une transition du diagnostiqueur
  - Le diagnostiqueur reconnaît le langage observable du système :

$$L_{OBS} = P_{OBS}(L(M))$$

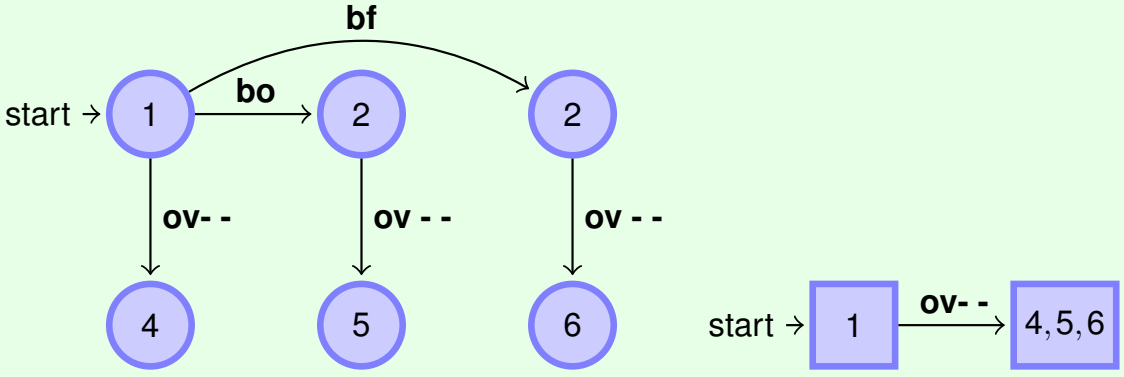
Construction de l'observateur

Exemple

1. Considérer l'état initial du modèle



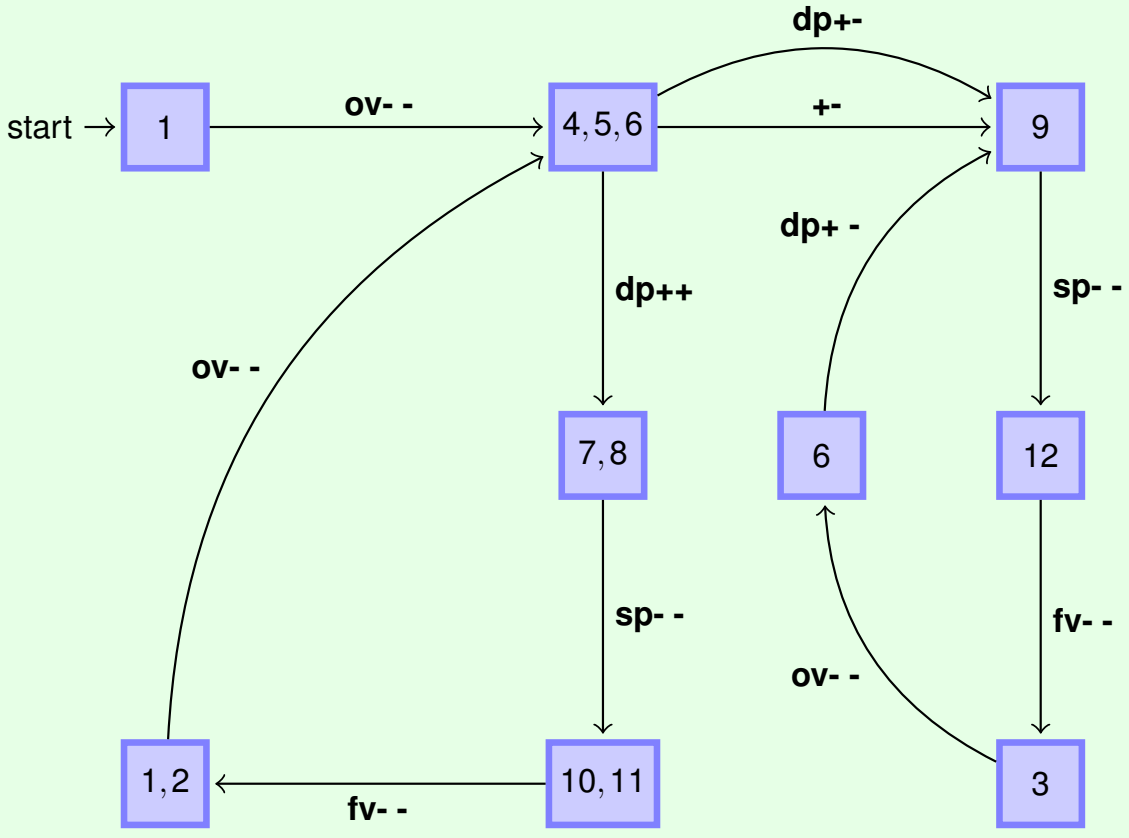
2. Rechercher à partir de ces états les chemins aboutissants à des états *Xo* cible d'une transition observable *o*



3. Considérer les états *Xo* nouvellement construits et réiterer 2 jusqu'à convergence

Observateur complet

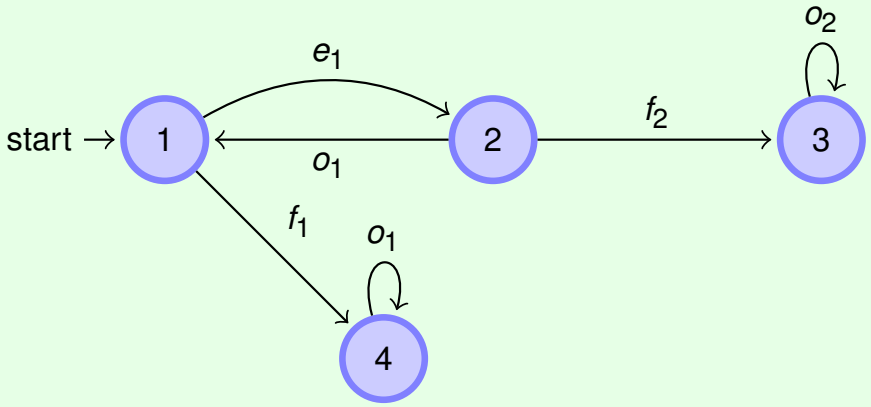
Exemple



À vous de jouer

Exemple

Modèle global :

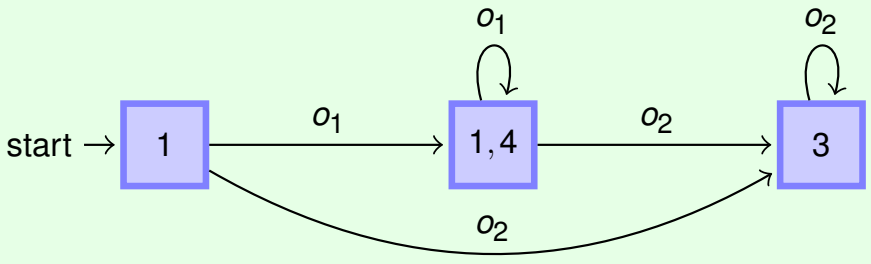


Calculer l'observateur de ce modèle : seuls o_1 ete o_2 sont observables

Résultat

Exemple

Observateur :



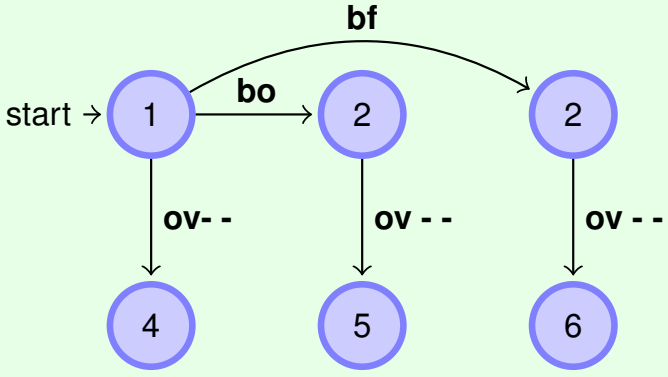
Construction du diagnostiqueur

Exemple

1. Considérer l'état initial du modèle, pas de panne

start → $1, \{\}$

2. Rechercher à partir de ces états les chemins aboutissants à des états *Xo* cible d'une transition observable *o* et propager les pannes

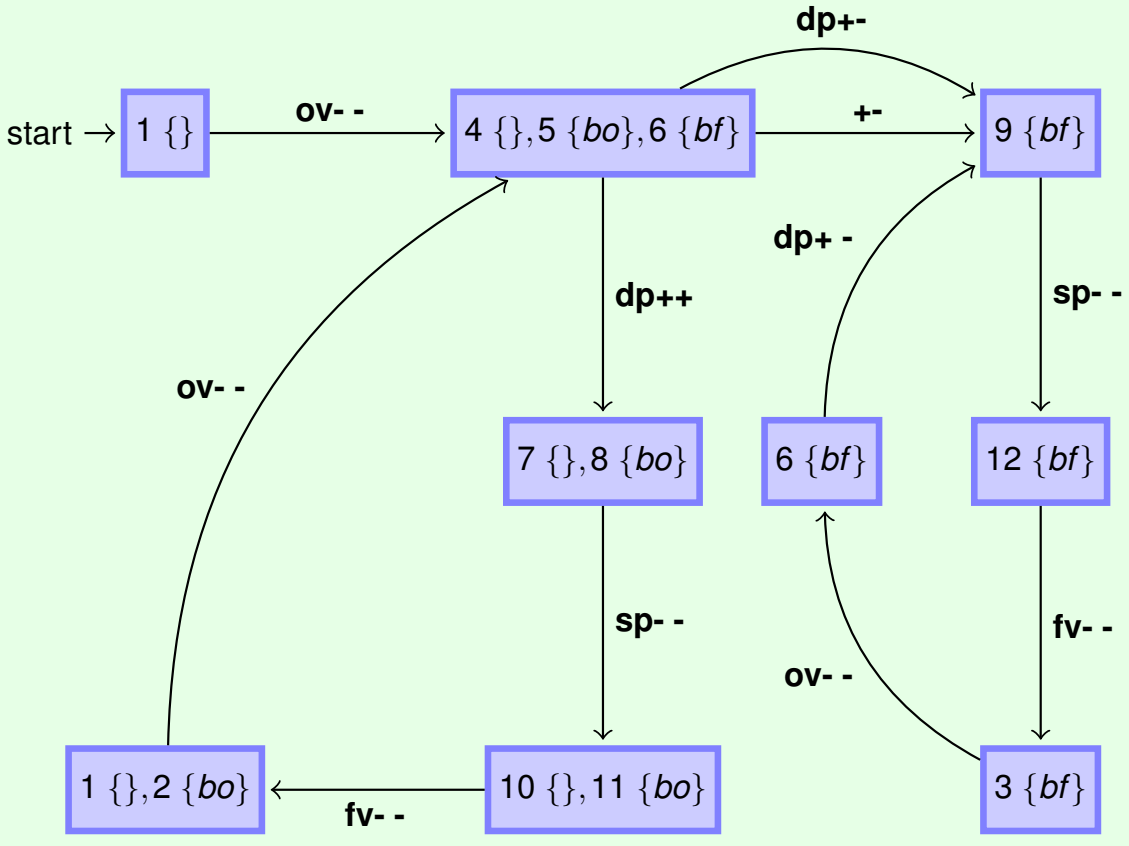


start → $1, \{\}$ —**ov- -**→ $4\{\}, 5\{bo\}, 6\{bf\}$

3. Considérer les états *Xo* nouvellement construits et réiterer 2 jusqu'à convergence

Diagnostiqueur complet

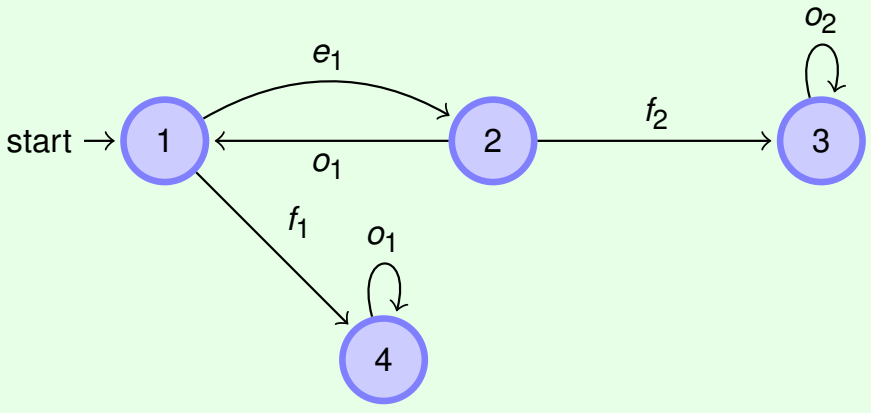
Exemple



À vous de jouer

Exemple

Modèle global :

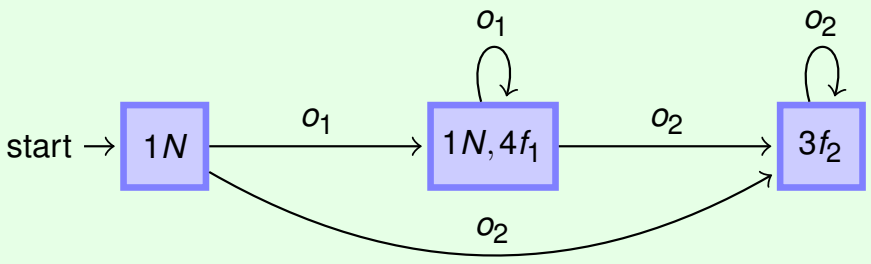


Calculer le diagnostiqueur de ce modèle : seuls o_1 ete o_2 sont observables. f_1 et f_2 sont les événements de pannes.

Résultat

Exemple

Diagnostiqueur :



Diagnostiqueur : la machine idéale et utopique

- Idéal : disposer d'un diagnostiqueur c'est :
  1. avoir une caractérisation complète d'un problème de diagnostic dans le SED
     - tout état du diagnostiqueur est un diagnostic possible
     - chaque diagnostic possible est un état du diagnostiqueur
  2. avoir un algorithme de diagnostic efficace (temps constant)
     - Adapter un diagnostic en fonction d'une nouvelle observation, c'est franchir une transition

- Utopique : Construire un diagnostiqueur, c'est :
  1. Disposer d'un modèle de panne (à base de composant) complet et correct
  2. Être capable de calculer le modèle global (produit synchronisé des composants)
  3. Être capable de calculer l'observateur associé

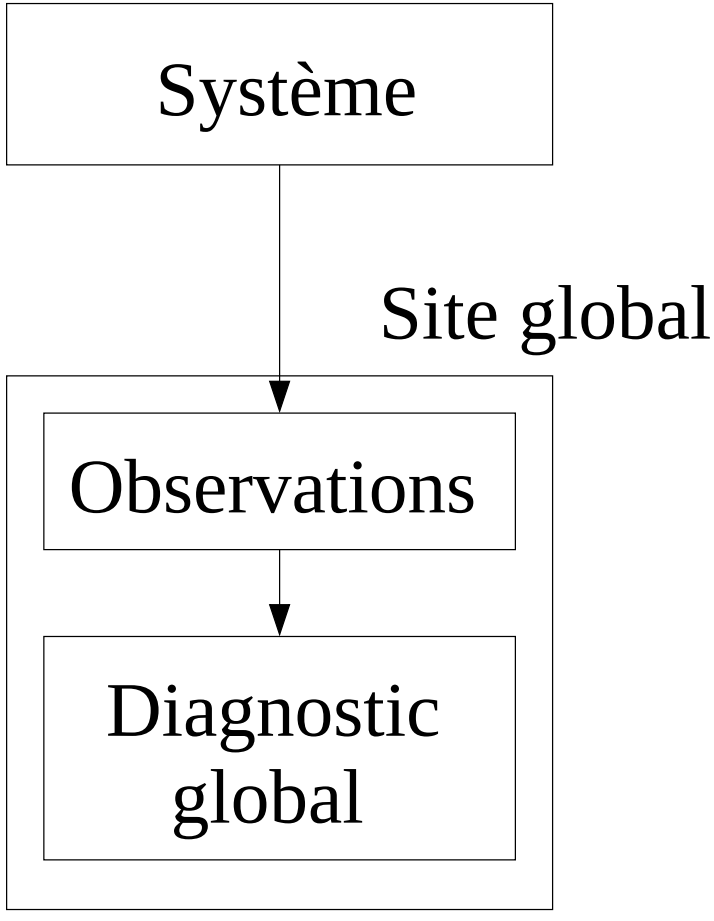
Un peu de complexité algorithmique

- Soit n le nombre de composants, on suppose que chaque composant contient m états, soit P le nombre de pannes dans le système
- Calculer le modèle global, c'est :
  - Déterminer tous ses états : au pire $N = m^n$ états !
  - Algorithme de calcul exponentiel en le nombre de composants : $O(2^n)$
- Calculer le diagnostiqueur, c'est :
  - Déterminer tous ses états : au pire $N_D = 2^N \times 2^P$ états !
  - Algorithme de calcul exponentiel en le nombre de pannes : $O(2^P)$
  - Doublement exponentiel en le nombre de composants : $O(2^{2^n})$

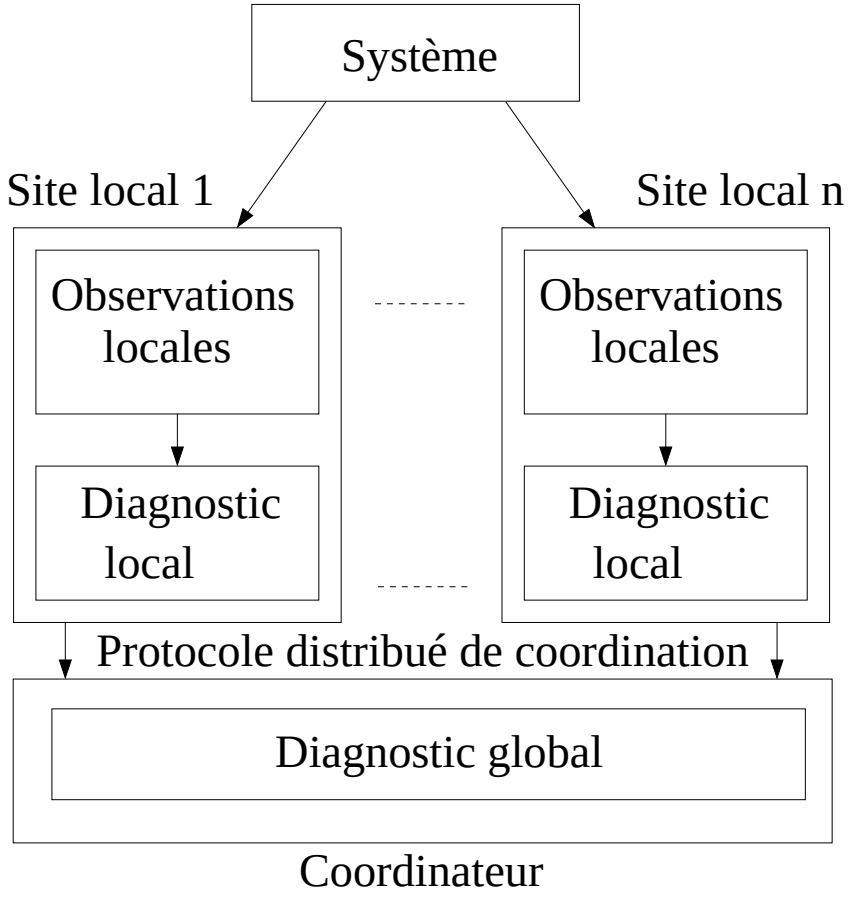
Exemple

Soit $n = 4$, $m = 10$, $f = 5$, au pire $N = 10000$ et $N_D = 10^{10000}$. A titre de comparaison, le nombre d'atomes dans l'univers n'est que de 10^{80}.

Architecture de diagnostic centralisée



Architecture de diagnostic coordonnée



Architecture de diagnostic distribuée