



Identification des systèmes à événements discrets : Etat de l'art et perspectives

Anaïs Guignard
LURPA, ENS Cachan

Projet VACSIM

Plan

1. Contexte

2. Identification des SED

- 2.1 Définition
- 2.2 Etat de l'art
- 2.3 Illustration

3. Perspectives de recherche

Laboratoire de recherche

Laboratoire Universitaire de Recherche en Production Automatisée



Equipe : Ingénierie des Systèmes Automatisés

Axes de recherche :

- Analyse des SED
- Identification des SED
- Synthèse des SED

Projet VACSIM

Tâche 1 : Validation par test progressif par parties de systèmes logiques

Tâche 2 : Validation par simulation de partie opérative

Tâche 3 : Apport des techniques d'identification des SED à la validation des systèmes critiques

Tâche 4 : Validation formelle de propriétés quantitatives : approche par automates

Tâche 5 : Validation formelle de propriétés quantitatives : approche par contraintes

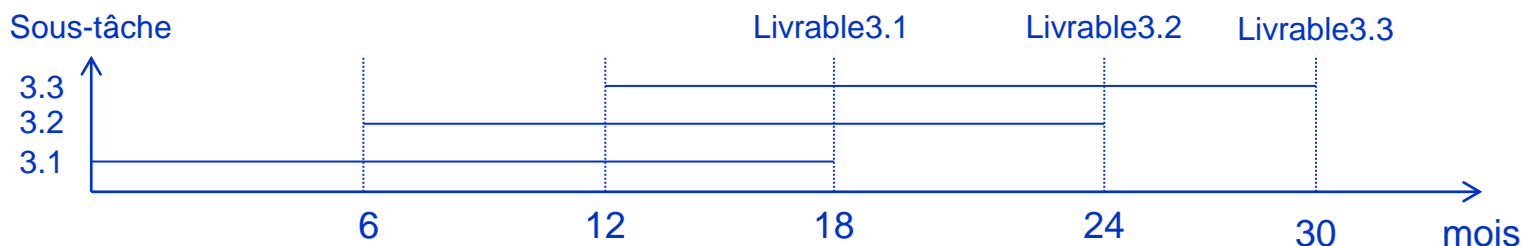
Tâche 6 : Démonstrateur et traitement de cas industriels

Tâche 3

Sous-tâche 3.1 : Identification d'un système bouclé contrôleur – simulateur de partie opérative.

Sous-tâche 3.2 : Validation par identification d'un système bouclé contrôleur – simulateur de partie opérative.

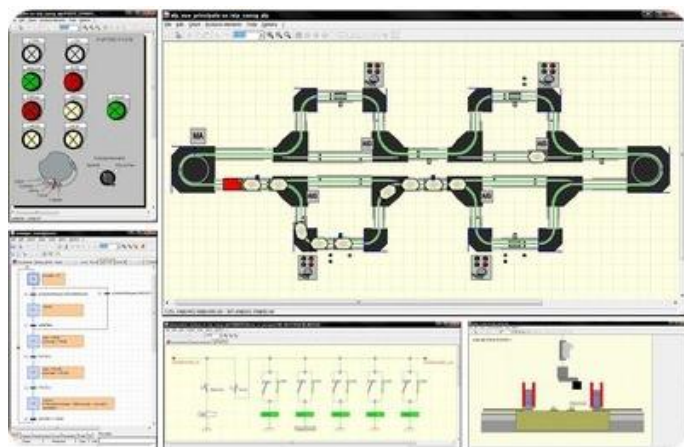
Sous-tâche 3.3 : Validation formelle par couplage identification – test de conformité.



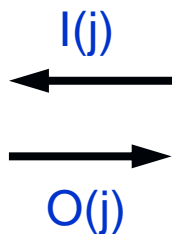
Système considéré

Système bouclé :

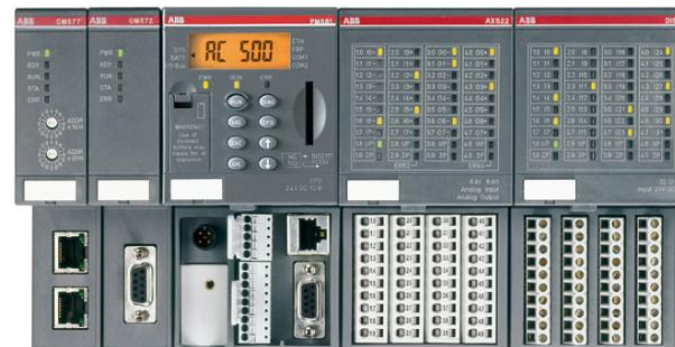
Procédé simulé



ControlBuild



Contrôleur réel



Automate Programmable Industriel
à scrutation cyclique des entrées

Spécifications Grafcet

Plan

1. Contexte

2. Identification des SED

- 2.1 Définition
- 2.2 Etat de l'art
- 2.3 Illustration

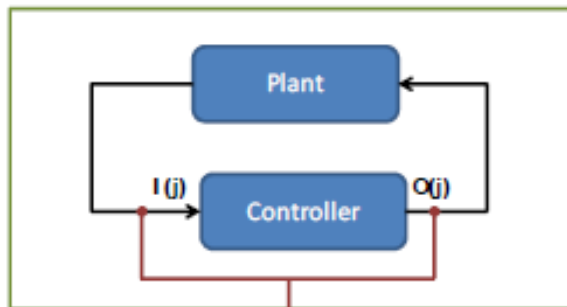
3. Perspectives de recherche

Définition

Identification [Klein, 05] :

« To determine an internal model of a system based on its external observed behaviors »

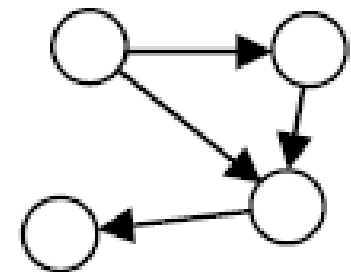
Système bouclé



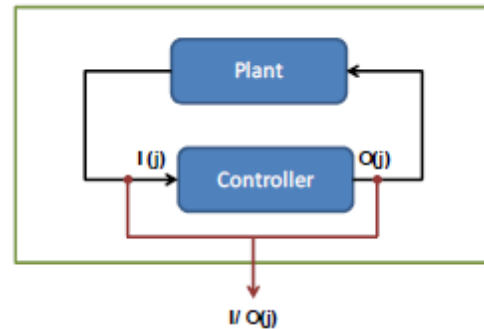
$I/O(j)$

$$\left(\begin{array}{c} 0 \\ 0 \\ 0 \\ 1 \end{array} \right) \cdot \left(\begin{array}{c} 1 \\ 0 \\ 0 \\ 1 \end{array} \right) \cdots \left(\begin{array}{c} 1 \\ 0 \\ 1 \\ 0 \end{array} \right) \left(\begin{array}{c} 1 \\ 0 \\ 0 \\ 1 \end{array} \right)$$

Modèle de SED
Non-temporisé



Classes de modèle générées lors de l'identification



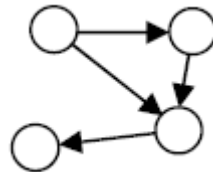
Automate à états finis

non-temporisé

[DeSmet99]

[Klein05]

[Roth10]



Réseaux de Petri interprété

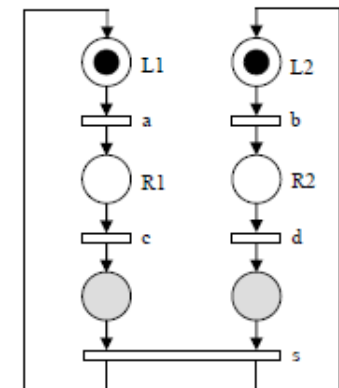
non temporisé

[Meda98]

[Cabasino07]

[Dotoli08]

[Estrada11]



Principaux champs applicatifs

- **Diagnostic de fautes :**
 - Capacité à détecter, identifier et isoler une faute avant qu'elle ne conduise à une défaillance
 - Fin de la phase de conception, pendant utilisation.
- **Reverse – engineering :**
 - Capacité à retrouver un modèle d'un système à partir de l'observation de sa structure, de ses fonctions et de ses opérations.
 - Lorsque les informations concernant la conception sont perdues ou obsolètes

Nouveau champ applicatif

- **Validation :**
 - Utilisation des méthodes d'identification pour la validation de systèmes à évènements discrets.
 - Pendant la phase de validation

Plan

1. Contexte

2. Identification des SED

- 2.1 Définition
- 2.2 Etat de l'art
- 2.3 Illustration

3. Perspectives de recherche

Critères de classification

Classés selon :

- **Le système**
 - Boite noire / Boite grise.
 - Présence d'indéterminisme ou non.
 - Prise en compte des phénomènes de concurrence ou non.
 - Evolutions possibles sans changement des variables d'entrée ou non.
- **La méthode**
 - Enumérative / Constructive.
 - Garantie d'exactitude de la solution sous certaines conditions

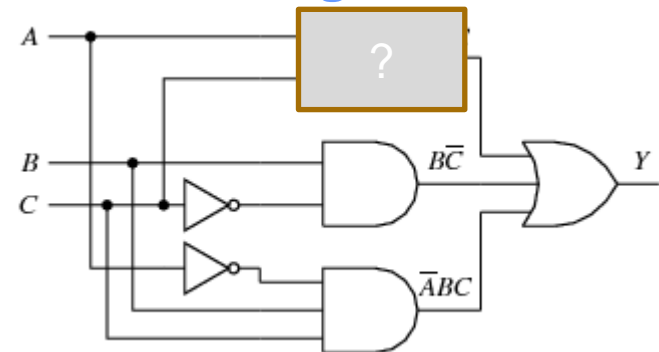
Boite noire / boite grise

Possède-t-on des connaissances concernant le système à identifier ?

Boite noire



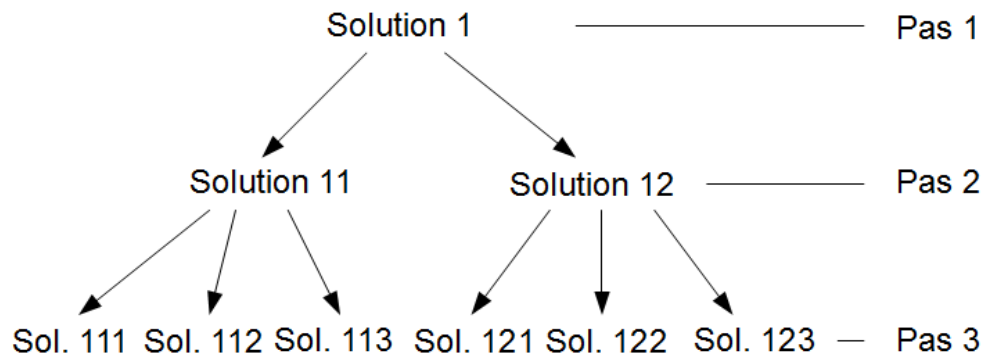
Boite grise



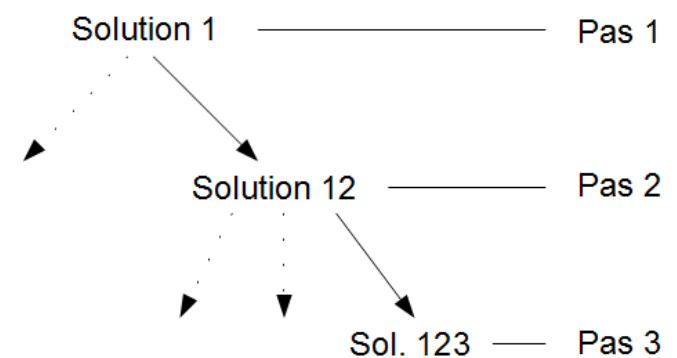
Enumératif / constructif

Est-ce qu'une seule ou toutes les solutions sont construites ?

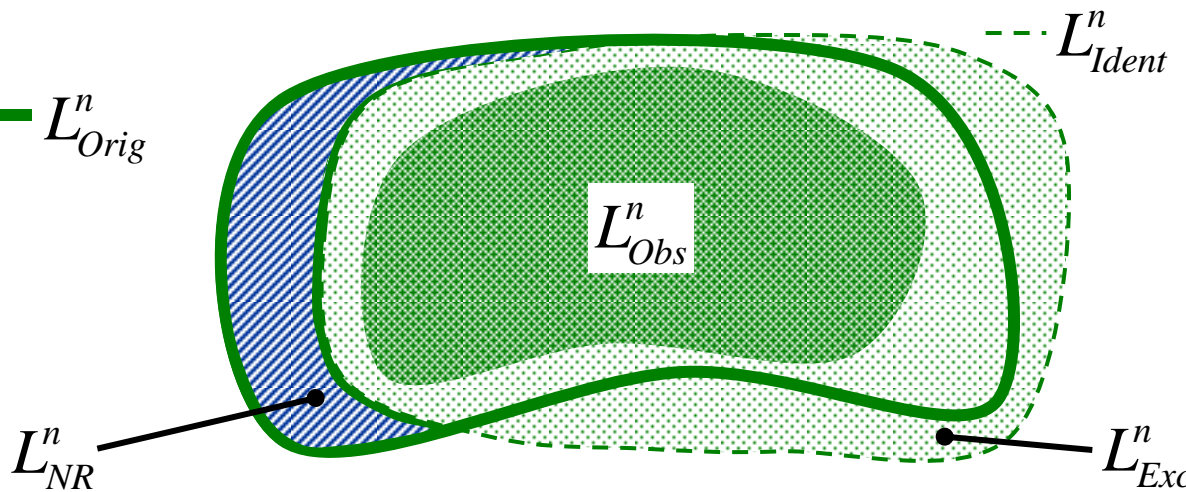
Enumératif



Constructif



Exactitude de la solution



L^n_{Orig} : Langage original (réel)
du système bouclé

L^n_{Obs} : Langage observé

L^n_{Ident} : Langage identifié

L^n_{Exc} : Langage en excès

L^n_{NR} : Langage non
reproductible

n : longueur des mots

[Klein, 05] :

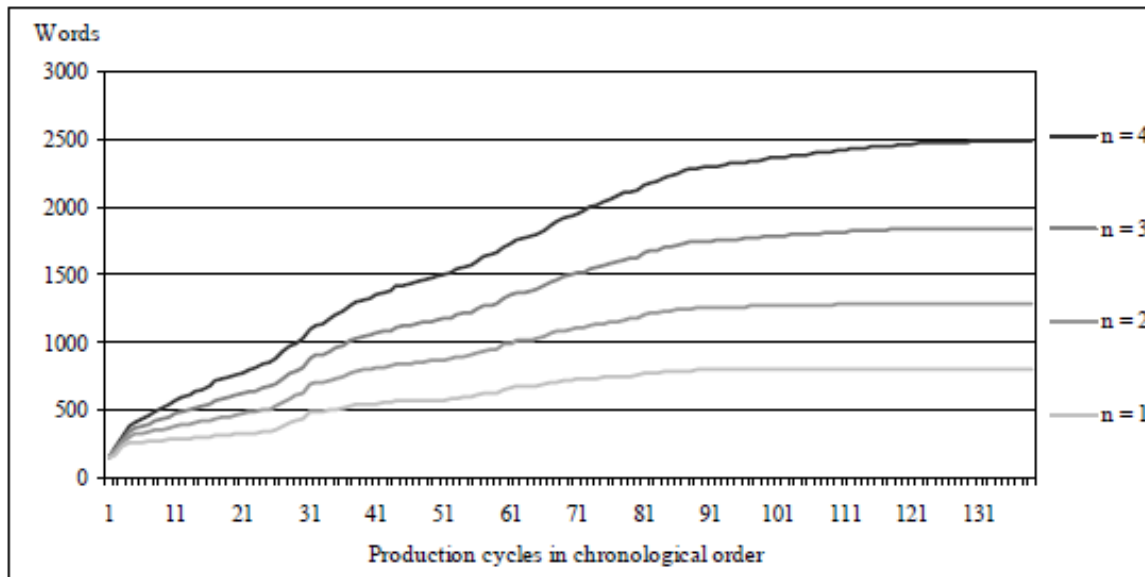
Theorem: $k+1$ completeness: $L^{k+1}_{Ident} = L^{k+1}_{Obs} \rightarrow L^{k+1}_{Exc} = \{\}$

Theorem: if $L^{k+1}_{Orig} = L^{k+1}_{Obs}$ then $L^{n+k}_{Ident} \supseteq L^{n+k}_{Orig} \rightarrow L^{n+k}_{NR} = \{\}$

Exactitude de la solution

Pour garantir l'exactitude, il est nécessaire de :

- Observer la totalité du langage original.
- Générer un langage identifié couvrant exactement le langage observé.



Convergence pour $n=3$:

$$L_{Orig}^{n=3} = L_{Obs}^{n=3}$$

Avec $k = n-1 = 2$

Il est donc possible, avec un algorithme imposant :

$$L_{Ident}^{k+1} = L_{Obs}^{k+1}$$

D'obtenir :

$$L_{NR}^5 = \{ \}$$

$$L_{Exc}^3 = \{ \}$$

Soit : $L_{Ident}^3 = L_{Orig}^3$

Travaux considérés

Travaux considérés :

M. E. Meda, A. Ramirez, A. Malo : Identification in Discrete Event Systems, 1998

O. De Smet, J. M. Roussel, N. Hévin : Identification de machine séquentielle binaire : application à un système réactif, 1999

S. Klein, L. Litz, J. J. Lesage : Fault Detection of Discrete Event Systems using an Identification Approach, 2005

M. P. Cabasino, A. Giua, C. Seatzu : Identification of Petri nets from knowledge of their language, 2007

M. Dotoli, M. P. Fanti, A. M. Mangini : Real Time Identification of Discrete Event Systems Using Petri Nets, 2008

A. P. Estrada-Vargas, J. J. Lesage, E. Lopez-Mellado : Stepwise Identification of Automated Discrete Manufacturing Systems, 2011

Bilan

	Meda	DeSmet	Klein	Cabasio	Dotoli	Estrada
Boite	Noire	Noire	Noire	Grise	Grise	Noire
Indétermi- nismes	Oui	Non	Oui	Non	Non	Oui
Concurrence	Oui	Non	Oui	Non	Non	Oui
Evolution possible sans changement des entrées	Oui	Non	Oui	Non	Non	Oui
Enumératif / Constructif	Constructif	Constructif	Constructif	Constructif	Constructif	Constructif
Exactitude de la solution	Non	Non	Oui	Non	Non	Oui

Plan

1. Contexte

2. Identification des SED

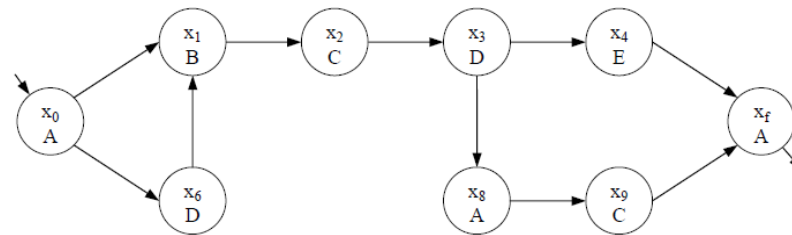
- 2.1 Définition
- 2.2 Etat de l'art
- 2.3 Illustration

3. Perspectives de recherche

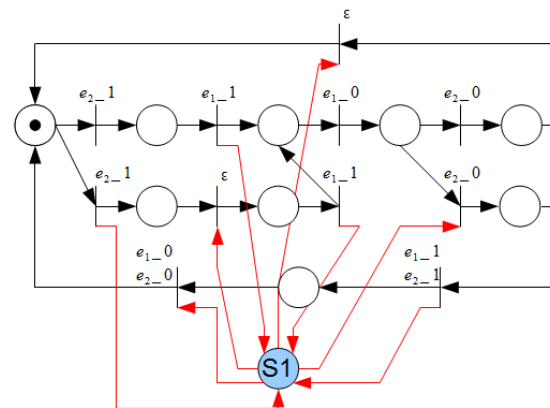
Choix des méthodes illustrées

Description détaillée de deux méthodes d'identification :

- Identification d'un automate à états finis [Klein, 05]



- Identification d'un réseau de Petri sauf interprété [Estrada, 11]



Exemple illustratif

- Variables d'entrée : $\{e_1, e_2\}$
- Variables de sortie : $\{s_1\}$
- Vecteur d'entrée-sortie de la forme $w(i) = [e_1(i) \ e_2(i) \ | \ s_1(i)]^T$
- Séquences permettant l'identification :

$$\sigma_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\sigma_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

- Garantie de l'exactitude de la solution pour des mots de longueur $k = 2$.

Application à [Klein, 05]

Modèle identifié : Automate Autonome Non-Déterministe avec Sorties

$NDAAO = (X, \Omega, f_{nd}, \lambda, x_0, x_f)$ avec :

- X , ensemble fini d'états.
- Ω , alphabet de sortie.
- $f_{nd} : X \rightarrow 2^X$, fonction de transition.
- $\lambda : X \rightarrow \Omega$, fonction de sortie.
- x_0 , état initial.
- x_f , état final.

Application à [Klein, 05]

Chaque lettre de la séquence est identifiée par :

$$A = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

$$D = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

$$E = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Les deux séquences peuvent alors s'écrire :

$$\sigma_1 = (A, B, C, D, E, A)$$

$$\sigma_2 = (A, D, B, C, D, A, C, A)$$

Application à [Klein, 05]

Etape 1 : Détermination des séquences de mots de longueur k

La séquence de mots de longueur k=2 issue de σ_1 s'écrit :

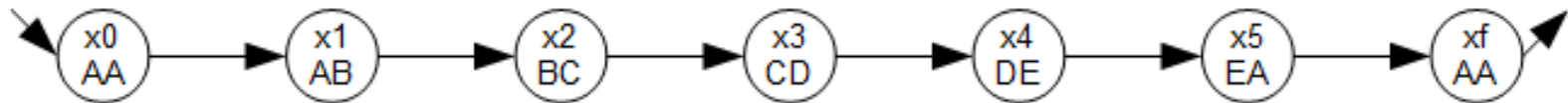
$$\sigma_1^2 = ((A, A), (A, B), (B, C), (C, D), (D, E), (E, A), (A, A))$$

De même, σ_2 devient :

$$\sigma_2^2 = ((A, A), (A, D), (D, B), (B, C), (C, D), (D, A), (A, C), (C, A), (A, A))$$

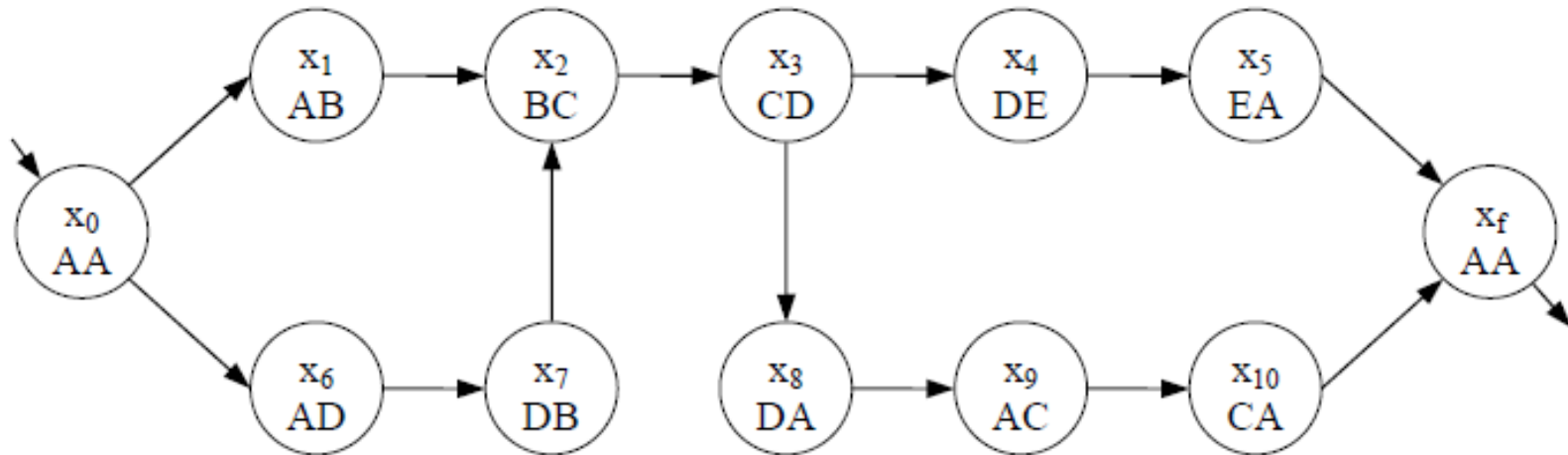
Etape 2 : Construction pas à pas de la structure de l'automate

On construit l'automate décrivant la séquence σ_1^2 :



Application à [Klein, 05]

En ajoutant σ_2^2 à la première on obtient :

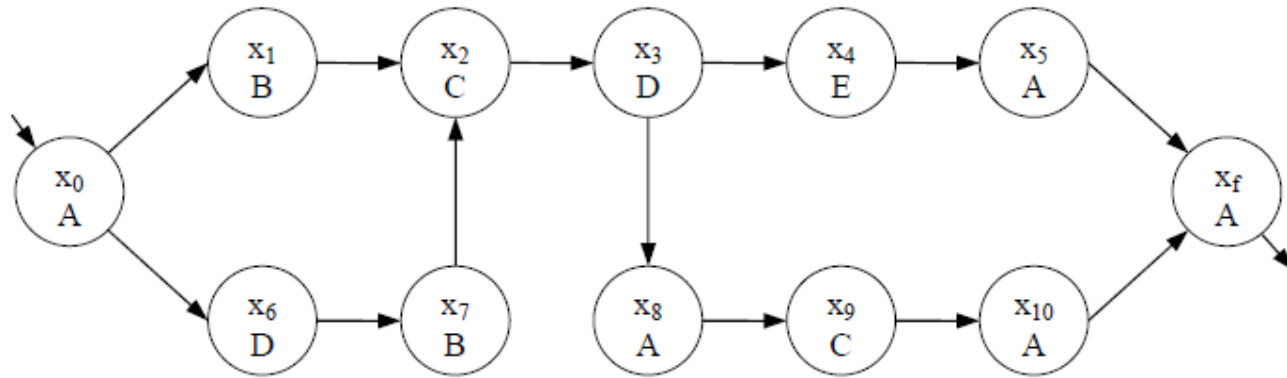


Chacun des états portant le même identifiant sont fusionnés en un unique état.

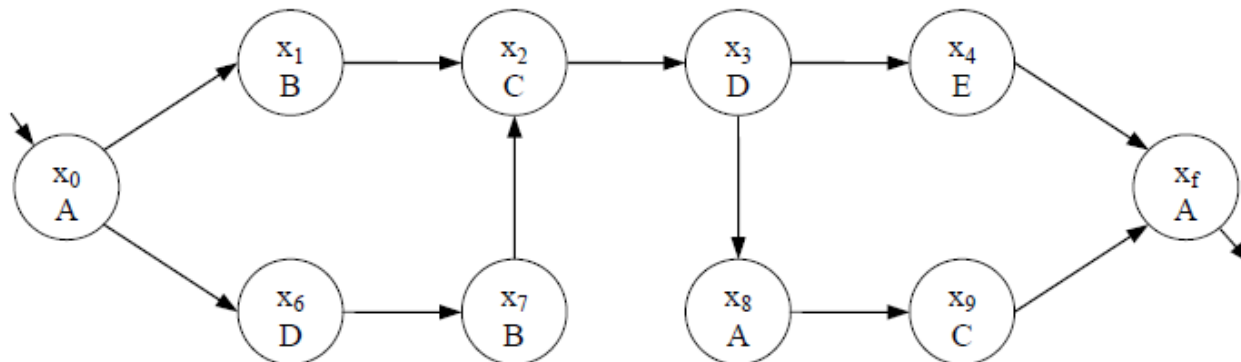
Application à [Klein, 05]

Etape 3 : Caractérisation des états par un unique vecteur E/S

Chaque état est associé au dernier vecteur de chaque mot :



Etape 4 : Simplification de l'état final :

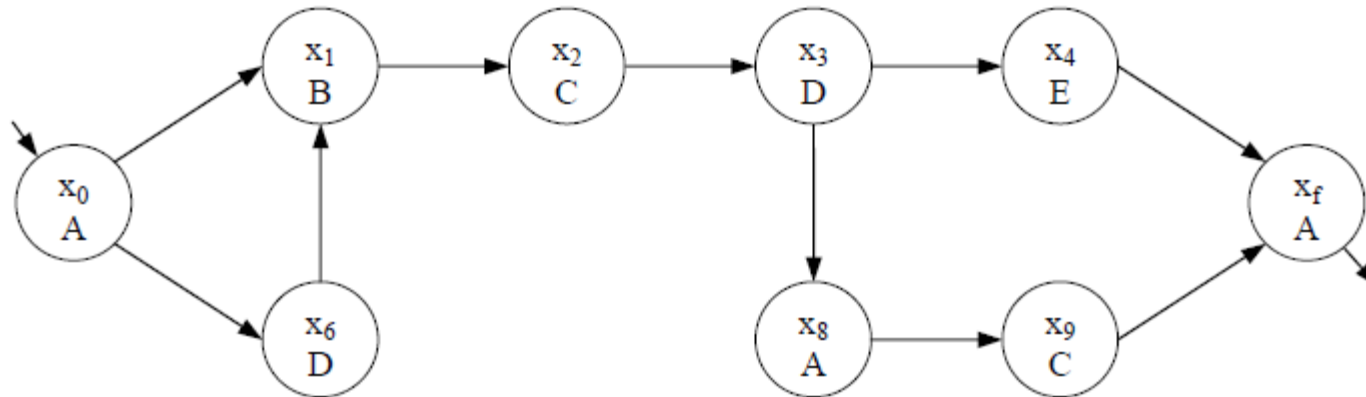


Application à [Klein, 05]

Etape 5 : Fusion des états

Deux états x_i , x_j sont équivalents si :

- Leur fonction de sortie est identique : $\lambda(x_i) = \lambda(x_j)$
- Leur ensemble d'états successeurs sont identique : $f_{nd}(x_i) = f_{nd}(x_j)$



Cas d'étude

Cas d'étude : Station de bobinage

336 E/S – 130 cycles observés – 350 mots différents

Notations :

$L^n(NDAAO)$: Mots de longueur n générés par le NDAAO.

L_{Obs}^n : Mots de longueur n observés.

X : Etats du NDAAO.

$f_{nd} : X \rightarrow 2^X$: Fonction de transition.

Cas d'étude

Critères :

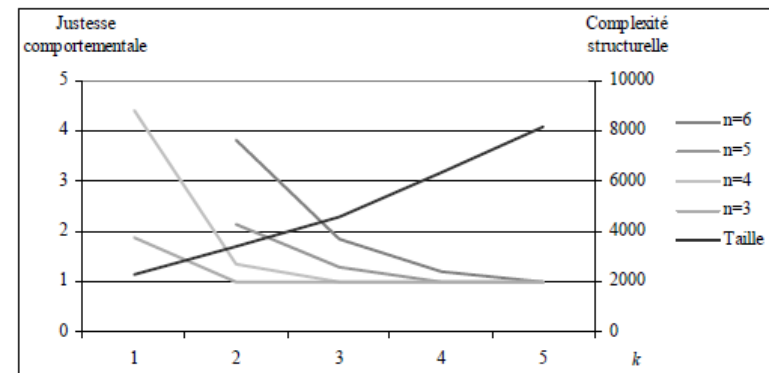
Justesse comportementale

$$C_B^n = \frac{|L^n(NDAAO)|}{|L_{Obs}^n|}$$

Complexité structurelle

$$C_{S1} = |\mathbf{X}| \quad C_{S2} = \frac{\sum_{x_i \in X} |f_{nd}(x_i)|}{|\mathbf{X}|}$$

k	Justesse comportementale						Complexité structurelle	
	C_B^1	C_B^2	C_B^3	C_B^4	C_B^5	C_B^6	C_{S1}	C_{S2}
1	1	1	1,87	4,40	n.c.	n.c.	2266	1,80
2	1	1	1	1,36	2,15	3,82	3390	1,65
3	1	1	1	1	1,29	1,85	4591	1,52
4	1	1	1	1	1	1,21	6347	1,39
5	1	1	1	1	1	1	8172	1,30



Application à [Estrada, 11]

Application de la méthode [Estrada 11] à l'exemple.

Modèle identifié : Réseau de Petri Interprété Sauf

$RdPIS = (Q, M_0)$ avec :

- $Q = (G, \Sigma, \Phi, \lambda, \varphi)$ la structure du réseau avec :
 - $G = (P, T, I, O)$ une structure ordinaire de réseau de Petri avec :
 - P l'ensemble des places.
 - T l'ensemble des transitions.
 - $I : P \times T \rightarrow \{0, 1\}$ la fonction des arcs entrants.
 - $O : P \times T \rightarrow \{0, 1\}$, la fonction des arcs sortants.
 - Σ , l'alphabet d'entrée.
 - Φ , l'ensemble des variables de sortie.
 - $\lambda : T \rightarrow \Sigma \cup \{\varepsilon\}$, la fonction de transition.
 - $\varphi : R(Q, M_0) \rightarrow \{0, 1\}^q$, la fonction de sortie, avec $q = |\Phi|$.
- M_0 , le marquage initial.

Application à [Estrada, 11]

Notations :

Soit la séquence $\sigma = (w_1, \dots, w_i, \dots, w_n)$ avec $w_i = [e_1(i) \ e_2(i) \ | \ s_1(i)]^T$

On définit $\tau(i) = w_i - w_{i-1}$

On définit $\tau^k(i) = (\tau(i - k + 1), \tau(i - k + 2), \dots, \tau(i))$

Pour chaque $\tau^k(i)$ apparaissant dans la séquence, un ensemble de place et transition sera défini.

La condition de franchissement de cette transition porte la valeur du dernier terme de $\tau^k(i)$: $t_r^{e_i} = \tau(i)$

Application à [Estrada, 11]

Etape 1 : Détermination des séquences

Les séquences s'écrivent :

$$\sigma_1 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \tau^2(1) \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \tau^2(2) \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \tau^2(3) \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \tau^2(4) \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \tau^2(5) \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\sigma_2 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \tau^2(6) \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \tau^2(7) \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \tau^2(8) \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \tau^2(9) \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \tau^2(10) \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \tau^2(11) \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \tau^2(12) \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

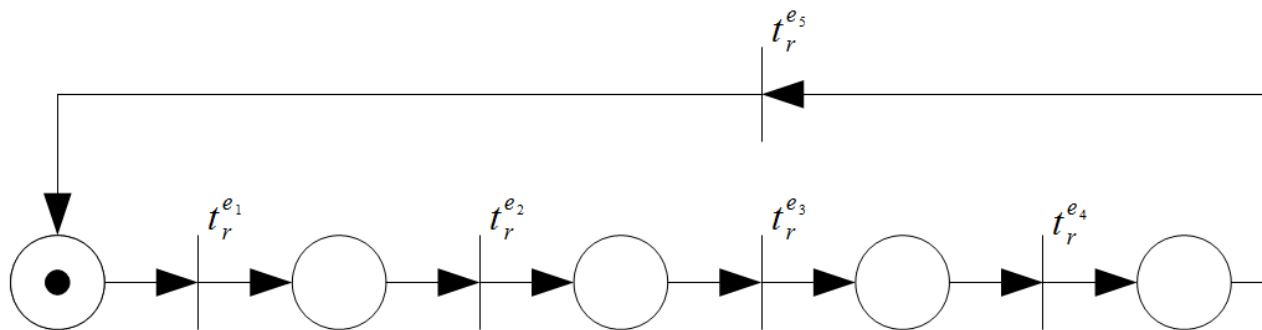
Application à [Estrada, 11]

Etape 2 : Construction de la structure du réseau de Petri

On a :

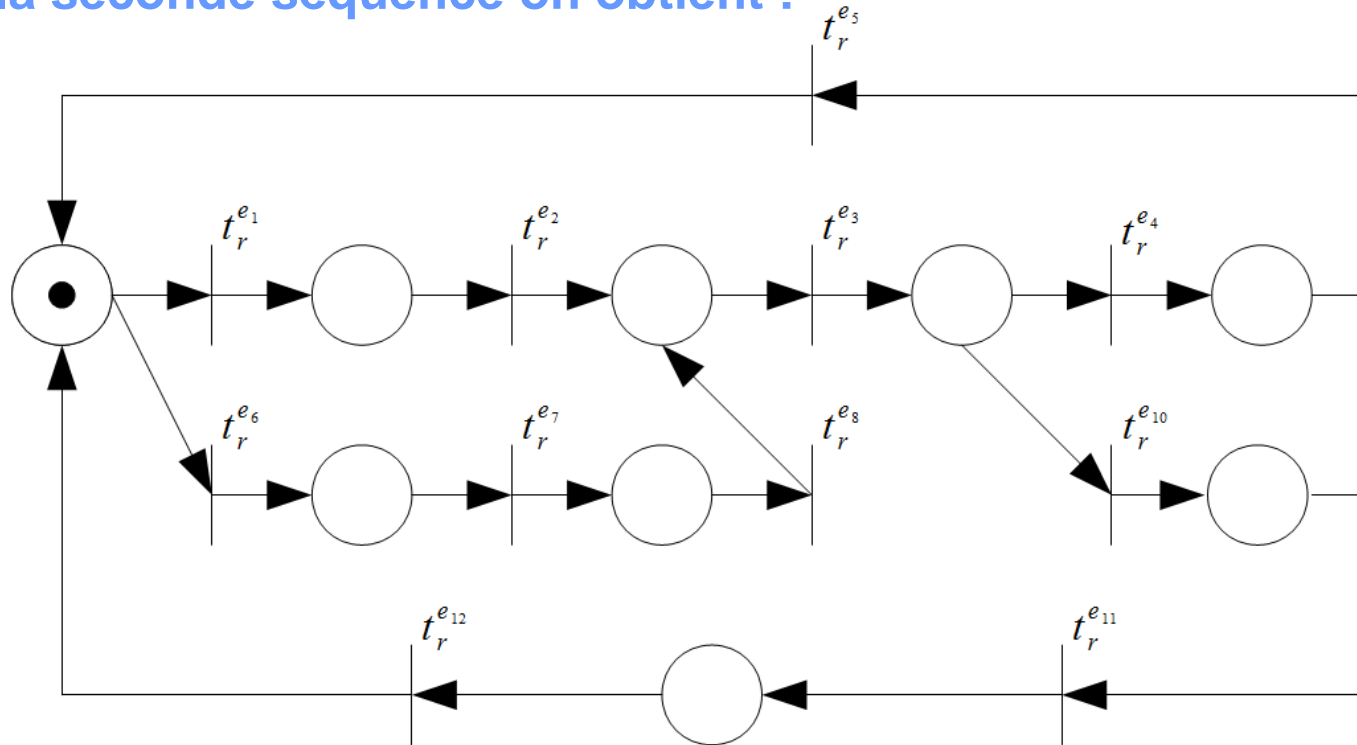
$$\sigma_1 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \tau^2(1) \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \tau^2(2) \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \tau^2(3) \begin{pmatrix} 1 & -1 \\ 0 & 0 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \tau^2(4) \begin{pmatrix} -1 & 0 \\ 0 & -1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \tau^2(5) \begin{pmatrix} 0 & 0 \\ -1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

La structure du réseau de Petri généré par la cette séquence est :



Application à [Estrada, 11]

En ajoutant la seconde séquence on obtient :



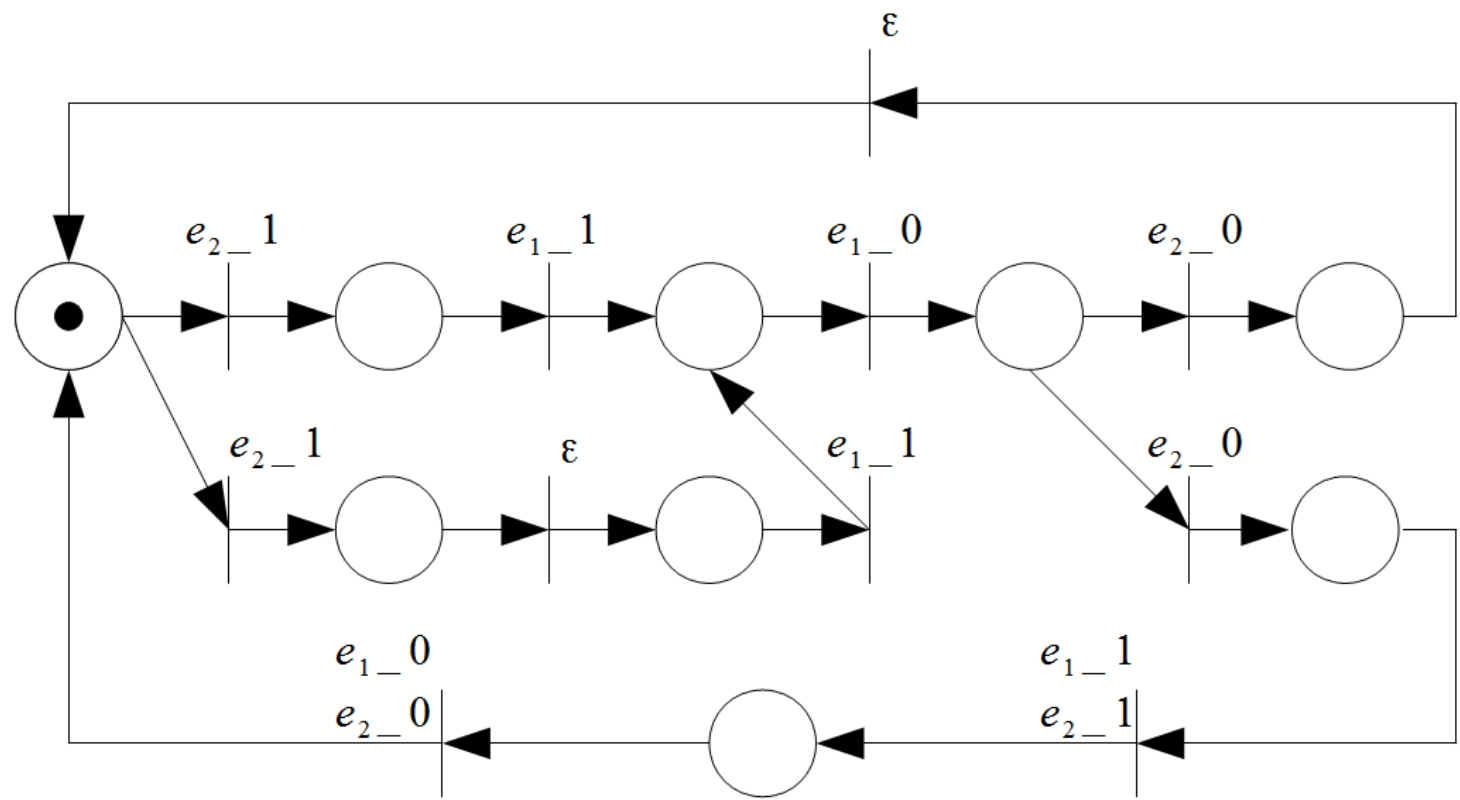
Etape 3 : Prise en compte de la concurrence

Cette étape simplifie le réseau de Petri en mettant en évidence les parallélismes s'il y a présence de phénomènes de concurrence.

Application à [Estrada, 11]

Etape 4 : Condition de franchissement des transitions

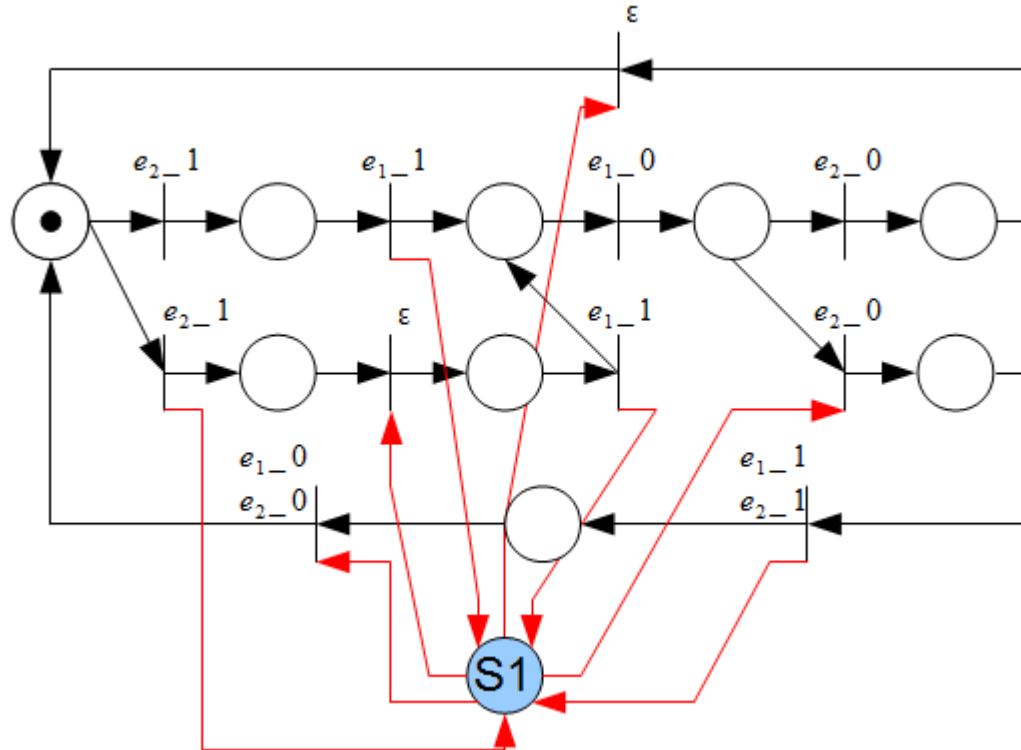
Les transitions représentent des variations des variables d'entrée.



Application à [Estrada, 11]

Etape 5 : Ajout des sorties

Création d'une place par sortie et ajout des arcs associés



Plan

1. Contexte

2. Identification des SED

- 2.1 Définition
- 2.2 Etat de l'art
- 2.3 Illustration

3. Perspectives de recherche

Rappel

Réalisation de chacune des sous-tâches :

- **Sous-tâche 3.1 : Identification d'un système bouclé contrôleur – simulateur de partie opérative.**
- **Sous-tâche 3.2 : Validation par identification d'un système bouclé contrôleur – simulateur de partie opérative.**
- **Sous-tâche 3.3 : Validation formelle par couplage identification – test de conformité.**

Perspectives

Sous-tâche 3.1 :

- **Prise en main de l'outil de simulation de partie opérative : ControlBuild.**
- **Développement d'une méthode d'identification adaptée à ce type de système bouclé.**
 - Modèle identifié de type NDAAO (Klein, Roth).
 - Modèle identifié de type Réseau de Petri Interprété Sauf (Estrada).
 - Modèle identifié de type Grafcet.

Sous-tâche 3.2 :

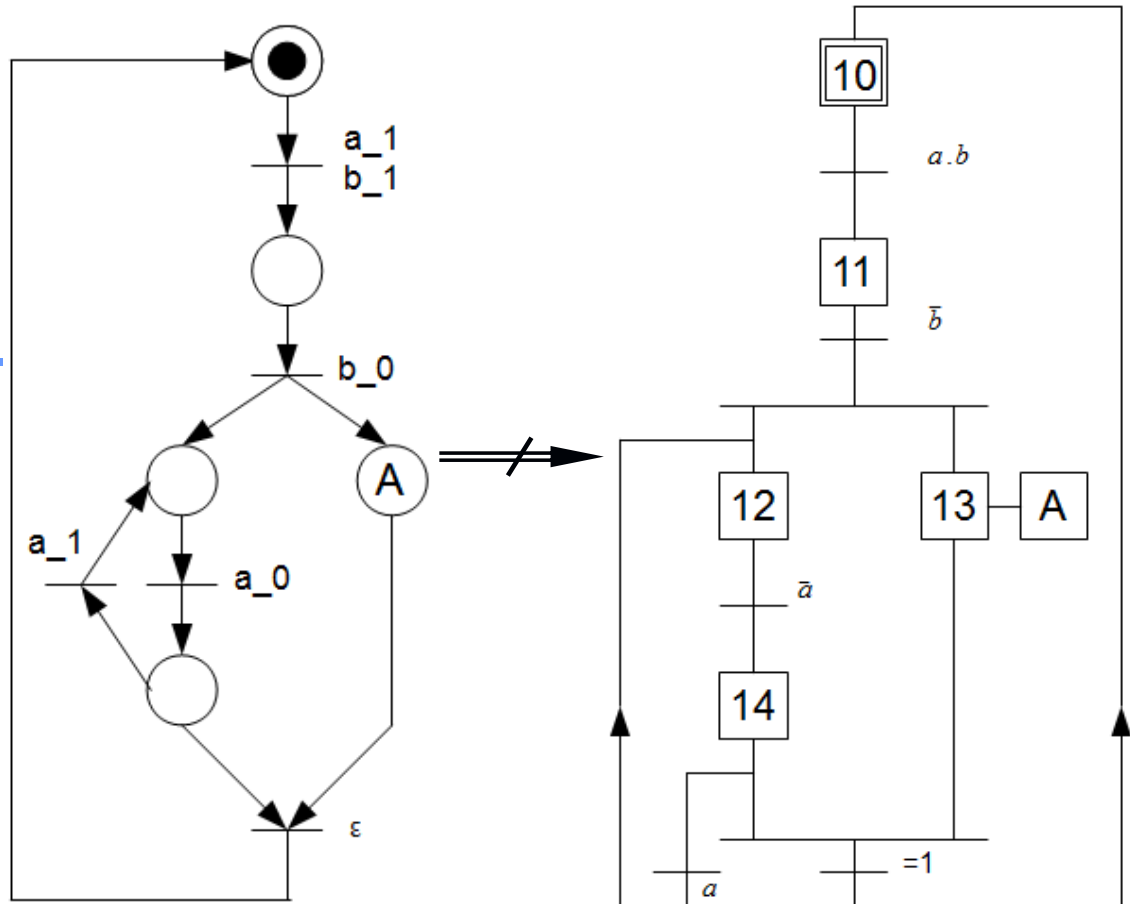
- **Choix d'une méthode de validation du modèle identifié :**
 - Par comparaison entre NDAAO et machine de Mealy.
 - Par comparaison entre réseau de Petri et Spécification Grafcet.
 - Par comparaison entre Grafcet identifié et Spécification Grafcet.
 - Par utilisation de la méthode d'apprentissage.

Verrous scientifiques

Modèle d'un réseau de Petri sauf proche d'un modèle Grafcet

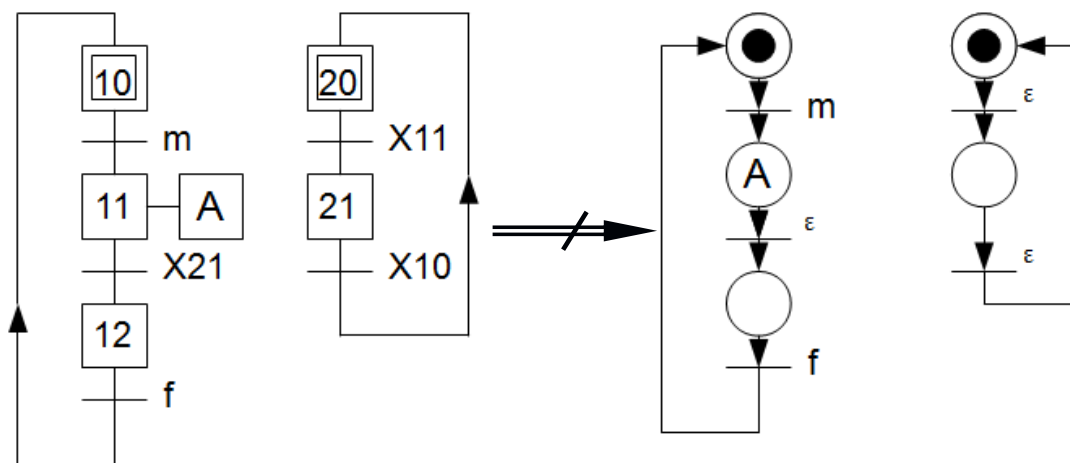
Cependant :
Règles d'évolutions
sensiblement différentes.

Peut amener à des
comportements non-
équivalents.



Verrous scientifiques

Ou bien à des situations
non-prises en charge
par un des modèles



Développement d'une méthode d'identification générant un Grafcet ?

Merci pour votre attention

Avez-vous des questions supplémentaires ?