Synthèse Algébrique

Formation Systèmes à Evénements Discrets

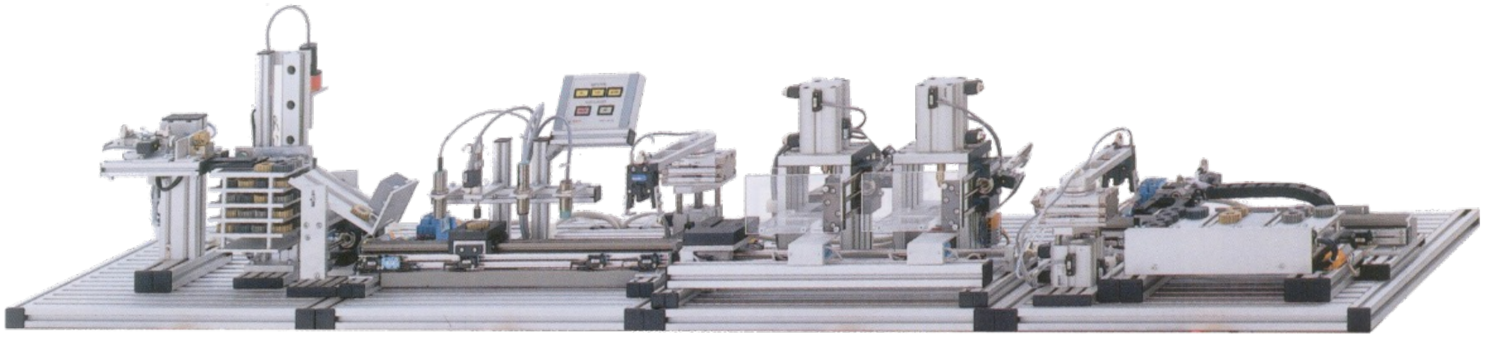
1ère édition
Janvier 2024

Société d'Automatique,
de Génie Industriel & de Productique

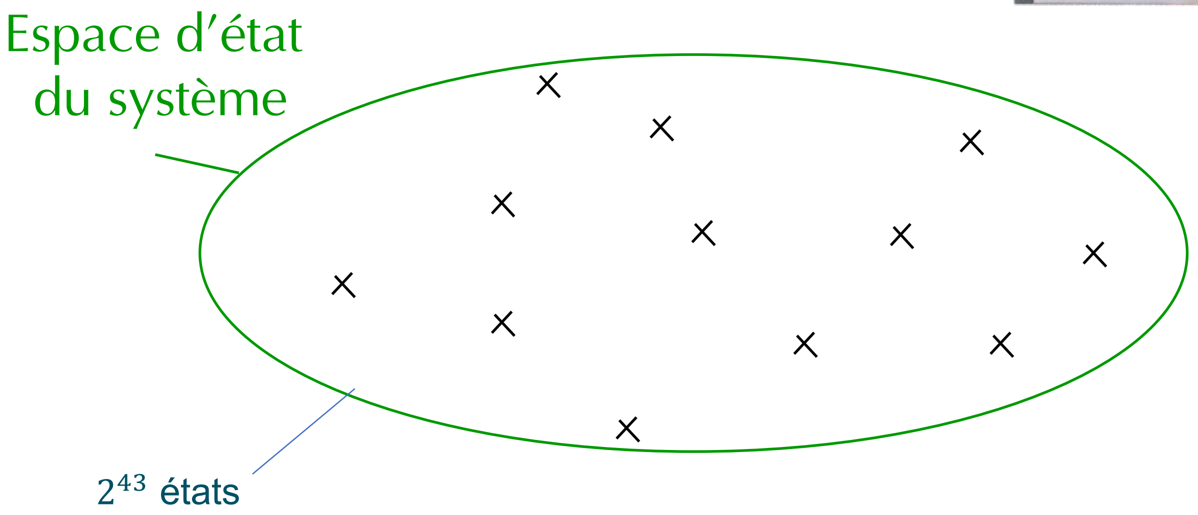
Introduction

Synthèse Algébrique

Considérons une chaine manufacturière



⇒ 43 entrées / 30 sorties

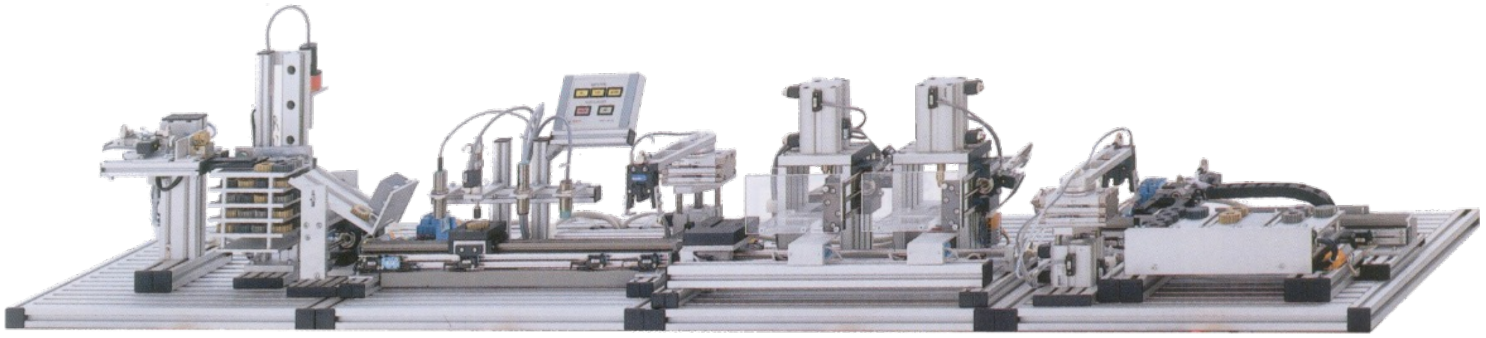


Vecteur Entrées | Vecteur Sorties

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ \vdots \\ e_i \end{bmatrix} \xrightarrow{\quad ? \quad} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ \vdots \\ s_j \end{bmatrix}$$

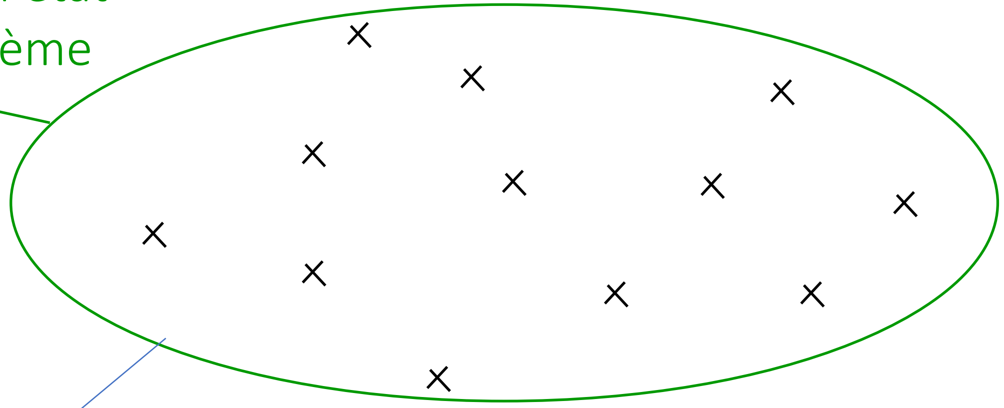
Synthèse Algébrique

Considérons une chaine manufacturière



$\Rightarrow$ 43 entrées / 30 sorties

Espace d'état du système



2^{43} états

Vecteur Entrées | Vecteur Sorties

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ \vdots \\ e_i \end{bmatrix} \qquad \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ \vdots \\ s_j \end{bmatrix}$$

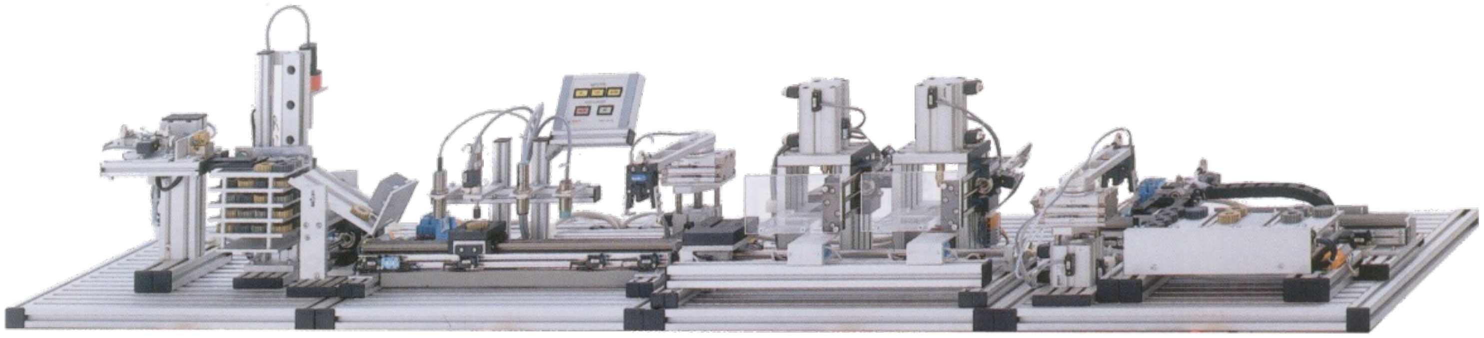
$$\boldsymbol{s_2 = F(e_1, e_2, \ldots, e_i)}$$

Equation algébrique

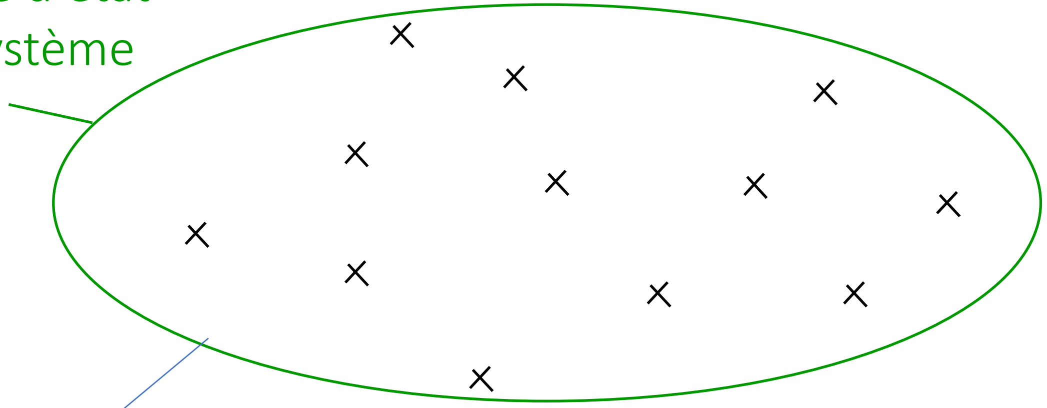
Synthèse Algébrique

Considérons une chaine manufacturière

⇒ 43 entrées / 30 sorties



Espace d'état du système



2^{43} états

L'évolution d'un système n'est qu'un ensemble d'équations algébriques

Vecteur Entrées — Vecteur Sorties

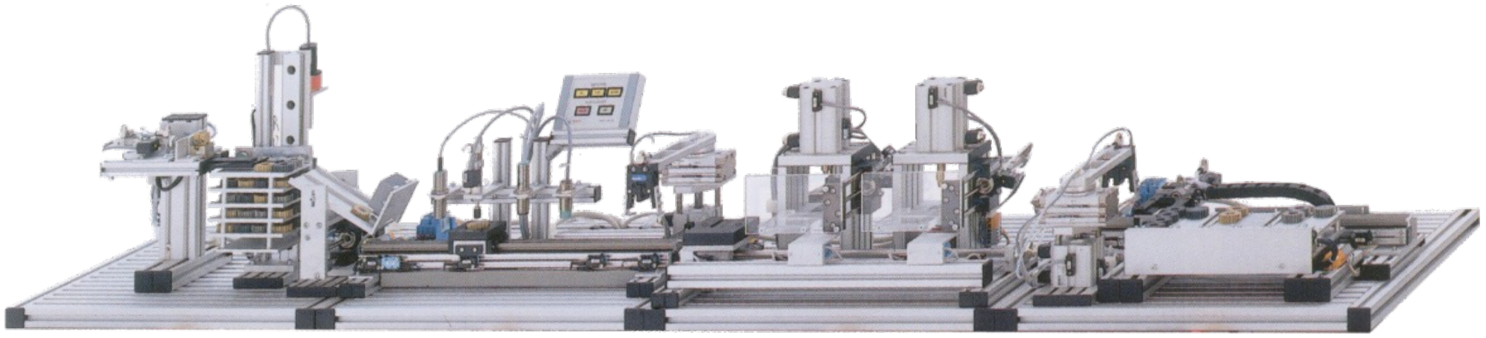
$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ \vdots \\ e_i \end{bmatrix} \qquad \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ \vdots \\ s_j \end{bmatrix}$$

$$\boldsymbol{s_2 = F(e_1, e_2, \dots, e_i)}$$

Equation algébrique

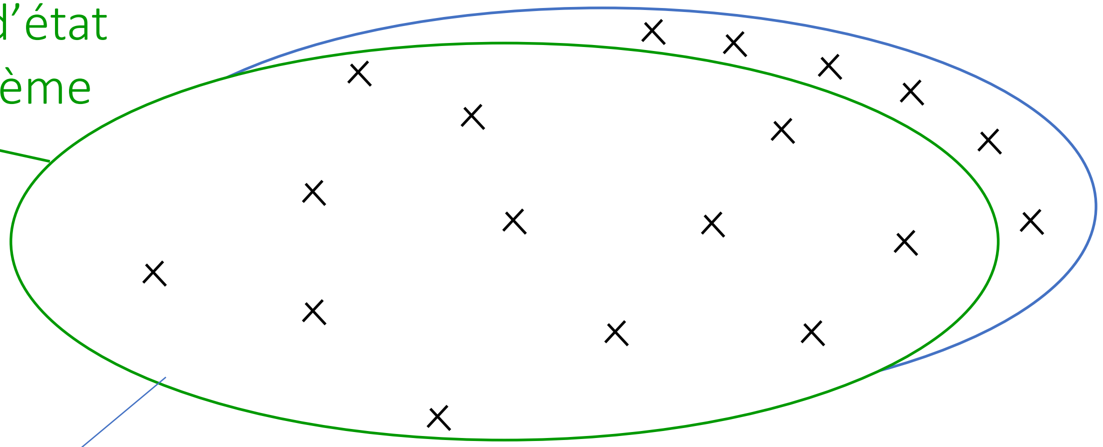
Synthèse Algébrique

Considérons une chaine manufacturière



$\Rightarrow$ 43 entrées / 30 sorties

Espace d’état du système



2^{43} états

Besoin d’augmenter l’espace d’état avec des variables internes

Vecteur Entrées

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ \vdots \\ e_i \end{bmatrix}$$

Vecteur Sorties

$$\begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ \vdots \\ s_j \end{bmatrix}$$

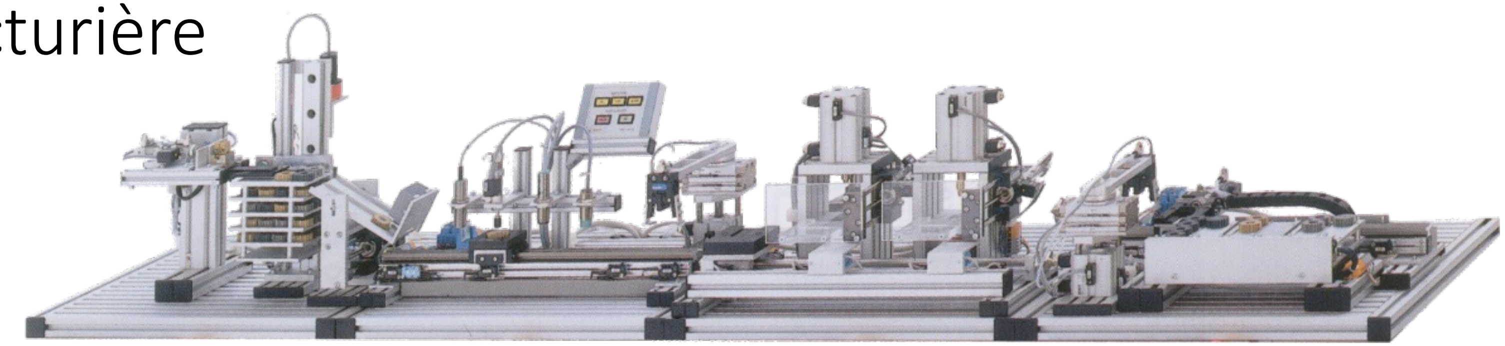
$$\boldsymbol{s_2 = F(e_1, e_2, \dots, e_i)}$$

Equation algébrique

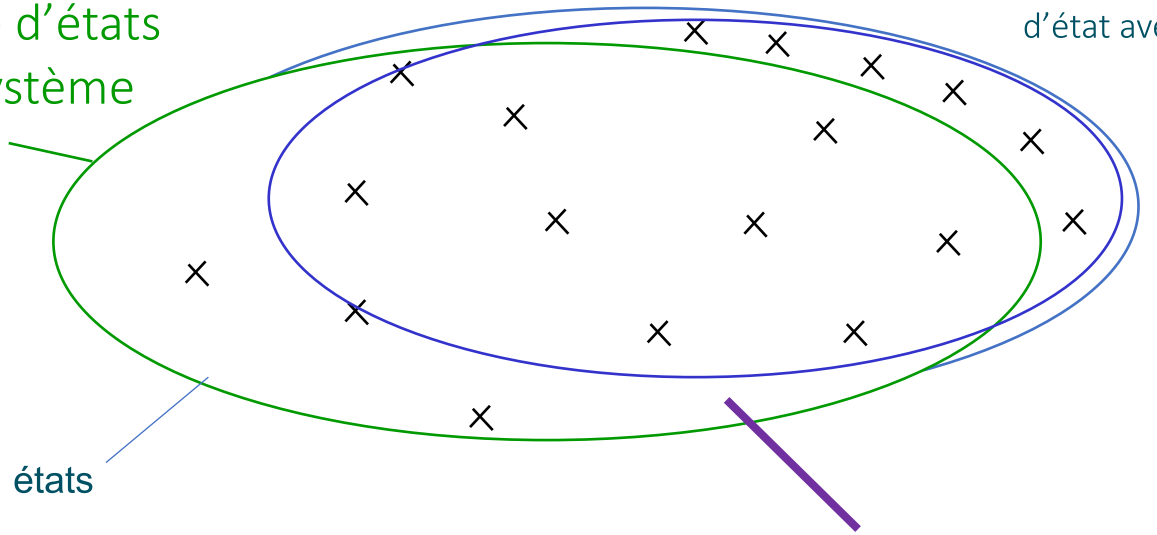
L’évolution d’un système n’est qu’une suite d’équations algébriques

Synthèse Algébrique

Considérons une chaine manufacturière



$\Rightarrow$ 43 entrées / 30 sorties



Besoin d'augmenter l'espace d'état avec des variables internes

$$s_2 = F(e_1, e_2, \dots, e_i, x_1, x_2, \dots, x_n)$$

L'évolution d'un système n'est qu'une suite d'équation algébrique

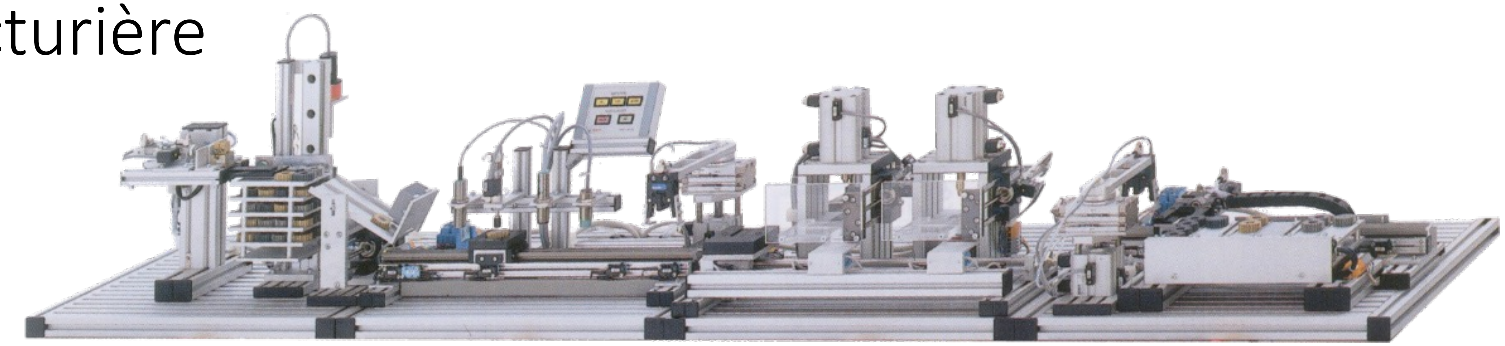
Vecteur Entrées	Vecteur Sorties
$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ \vdots \\ e_i \end{bmatrix}$	$\begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ \vdots \\ s_j \end{bmatrix}$

$$s_2 = F(e_1, e_2, \dots, e_i)$$

Equation algébrique

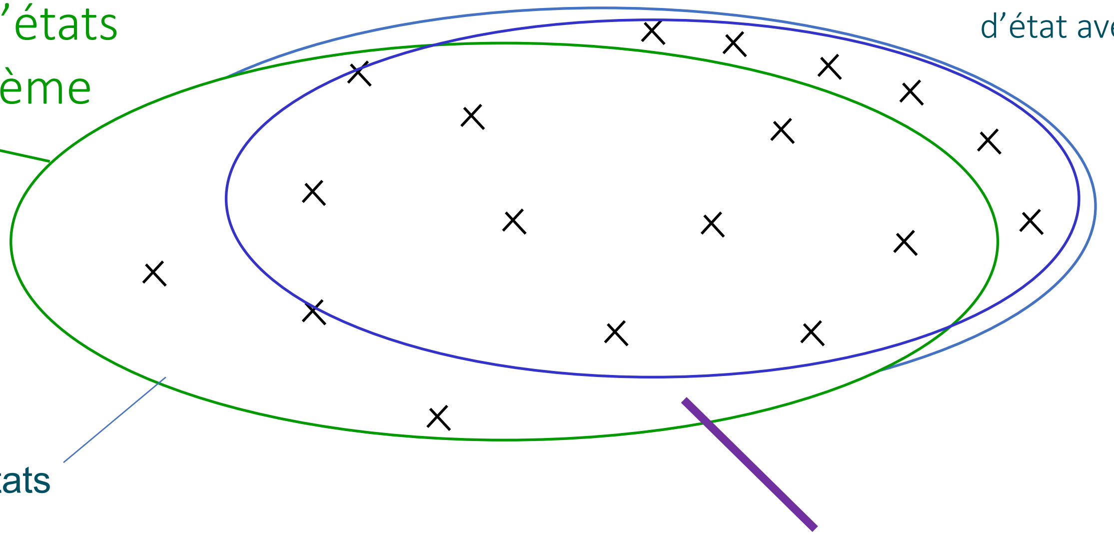
Synthèse Algébrique

Considérons une chaine manufacturière



⇒ 43 entrées / 30 sorties

Espace d'états
du système



2^{43} états

Espace représentant ce
que je souhaite faire du système

L'évolution d'un système n'est
qu'une suite d'équation algébrique

Besoin d'augmenter l'espace
d'état avec des variables internes

$$\boldsymbol{s_2 = F(e_1, e_2, \dots, e_i, x_1, x_2, \dots, x_n)}$$

Vecteur Entrées + variables

Vecteur Sorties
+ variables

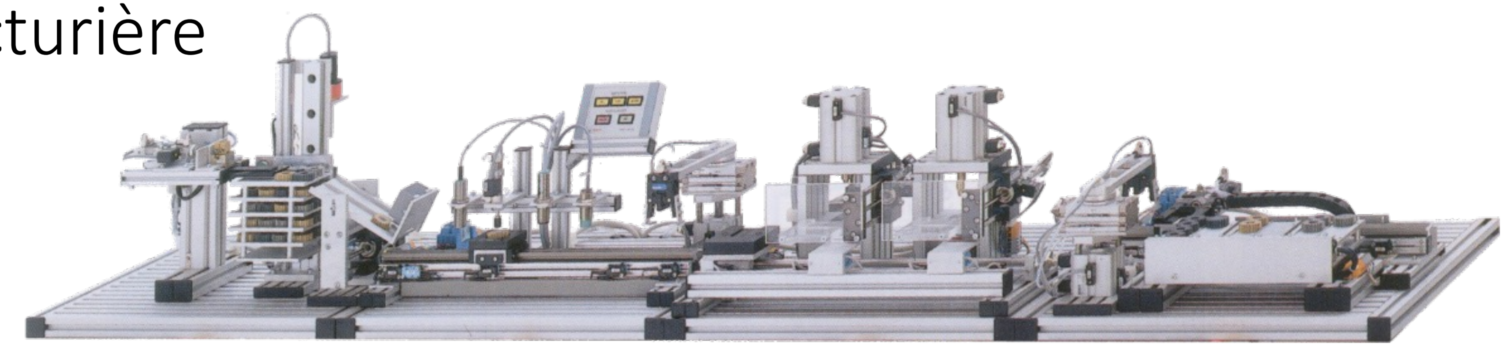
$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ \vdots \\ e_i \\ x_1 \\ x_2 \end{bmatrix} \qquad \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ \vdots \\ s_j \\ x_1 \\ x_2 \end{bmatrix}$$

$$\boldsymbol{s_2 = F(e_1, e_2, \dots, e_i, x_1, x_2, \dots, x_n)}$$

Equation algébrique

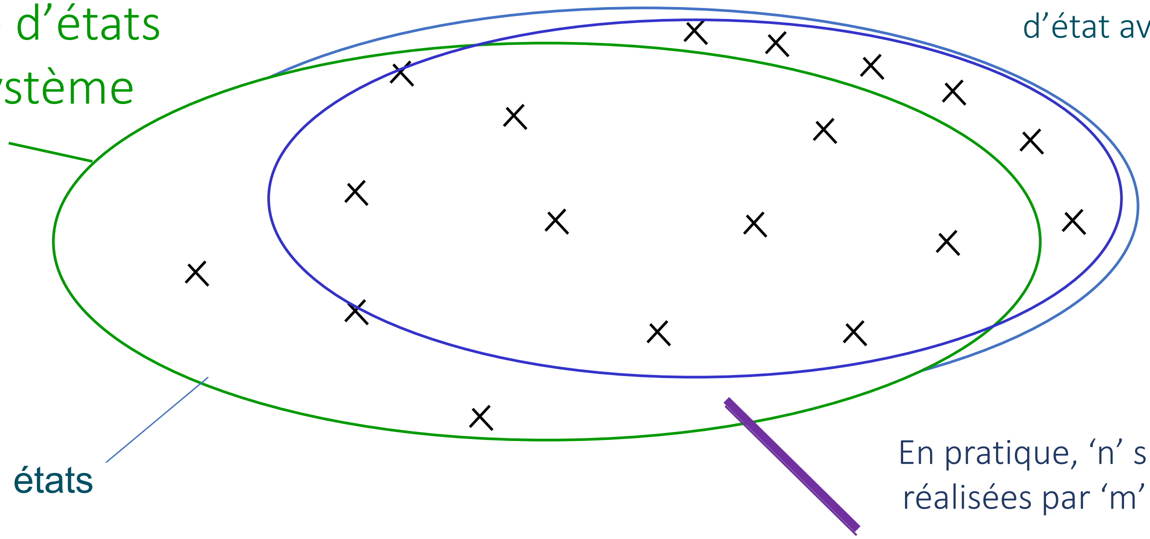
Synthèse Algébrique

Considérons une chaine manufacturière



⇒ 43 entrées / 30 sorties

Espace d'états
du système



2^{43} états

Besoin d'augmenter l'espace
d'état avec des variables internes

En pratique, 'n' spécifications
réalisées par 'm' concepteurs

~~Espace représentant ce~~
~~que je souhaite faire du système~~

L'évolution d'un système n'est
qu'une suite d'équation algébrique

Vecteur Entrées + variables

Vecteur Sorties
+ variables

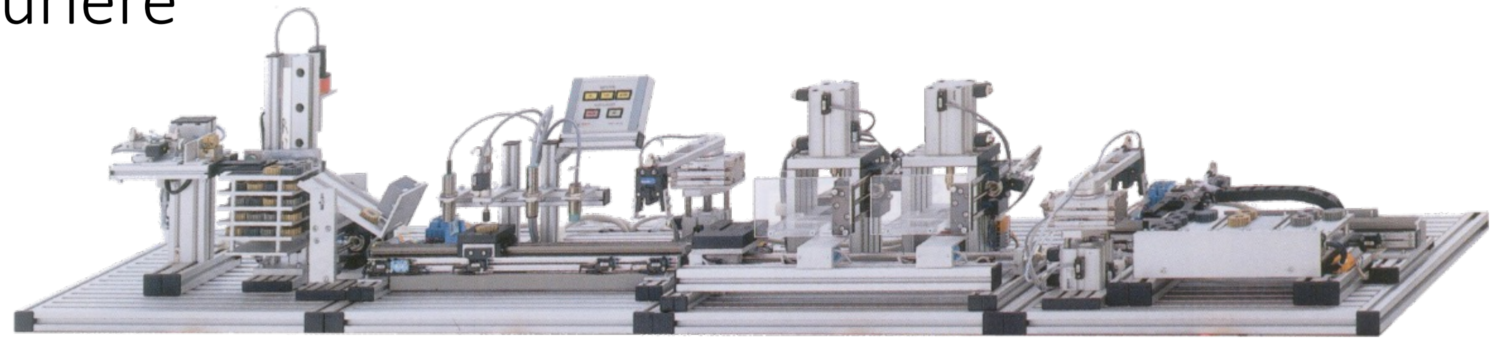
$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ \vdots \\ e_i \\ x_1 \\ x_2 \end{bmatrix} \qquad \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ \vdots \\ s_j \\ x_1 \\ x_2 \end{bmatrix}$$

$$\boldsymbol{s_2 = F(e_1, e_2, \dots, e_i, x_1, x_2, \dots, x_n)}$$

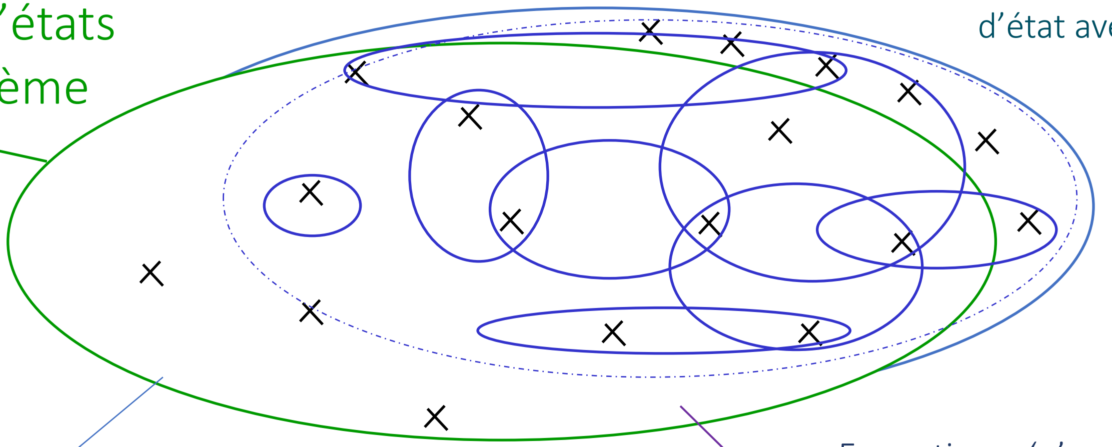
Equation algébrique

Synthèse Algébrique

Considérons une chaine manufacturière



$\Rightarrow$ 43 entrées / 30 sorties



Espace d'états du système

2^{43} états

Besoin d'augmenter l'espace d'état avec des variables internes

En pratique, 'n' spécifications réalisées par 'm' concepteurs

~~Espace représentant ce que je souhaite faire du système~~

L'évolution d'un système n'est qu'une suite d'équation algébrique

Vecteur Entrées + variables

Vecteur Sorties + variables

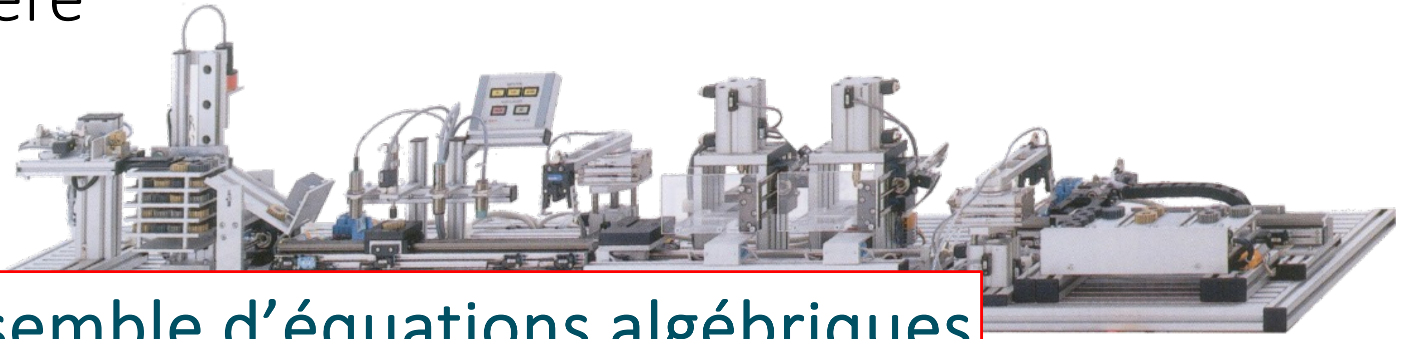
$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ \vdots \\ e_i \\ x_1 \\ x_2 \end{bmatrix} \qquad \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ \vdots \\ s_j \\ x_1 \\ x_2 \end{bmatrix}$$

$$\boldsymbol{s_2 = F(e_1, e_2, \dots, e_i, x_1, x_2, \dots, x_n)}$$

Equation algébrique

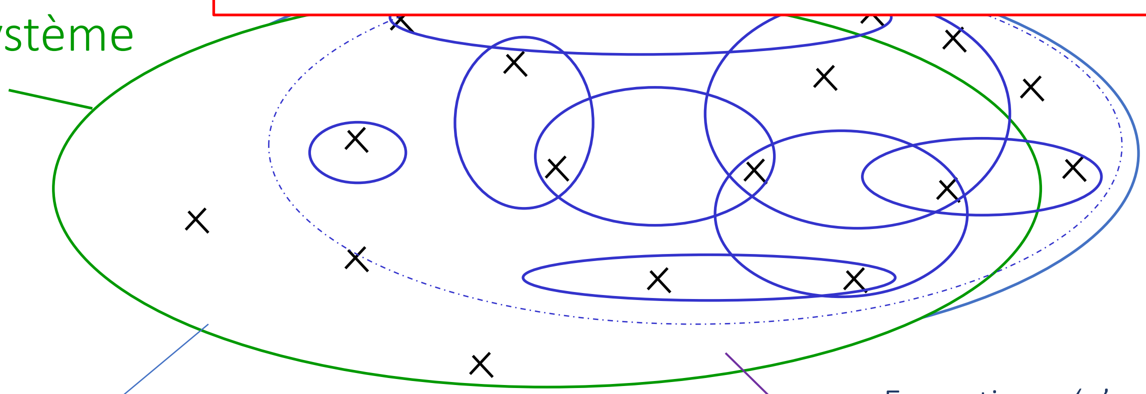
Synthèse Algébrique

Considérons une chaine manufacturière



⇒ 43 entrées / 30 sorties

Comment trouver un ensemble d’équations algébriques garantissant le respect des spécifications ?



L’évolution d’un système n’est qu’une suite d’équation algébrique

Vecteur Entrées + variables

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ \vdots \\ e_i \\ x_1 \\ x_2 \end{bmatrix}$$

Vecteur Sorties + variables

$$\begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ \vdots \\ s_j \\ x_1 \\ x_2 \end{bmatrix}$$

$$\boldsymbol{s_2 = F(e_1, e_2, \dots, e_i, x_1, x_2, \dots, x_n)}$$

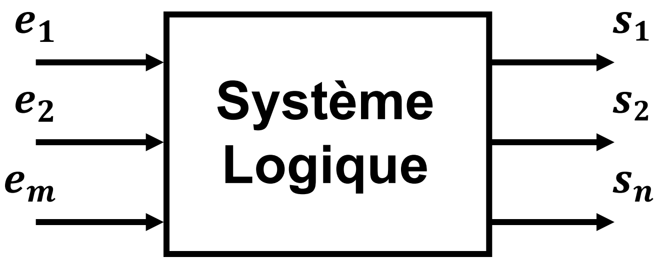
Equation algébrique

Rappels

Système combinatoire / séquentiel

- Système logique **combinatoire**
  - Système dont les entrées et sorties ont une valeur logique (0 ou 1)
  - Système pour lequel la valeur de chaque **sortie à tout instant k** est déterminée à partir de la valeur des **entrées à cet instant k**.

$$\begin{cases} \mathbf{s_1}[\boldsymbol{k}] = \mathbf{F_1}(\mathbf{e_1}[\boldsymbol{k}], \cdots, \mathbf{e}_{\boldsymbol{m}}[\boldsymbol{k}]) \\ \vdots \\ \mathbf{s}_{\boldsymbol{n}}[\boldsymbol{k}] = \mathbf{F}_{\boldsymbol{n}}(\mathbf{e_1}[\boldsymbol{k}], \cdots, \mathbf{e}_{\boldsymbol{m}}[\boldsymbol{k}]) \end{cases}$$

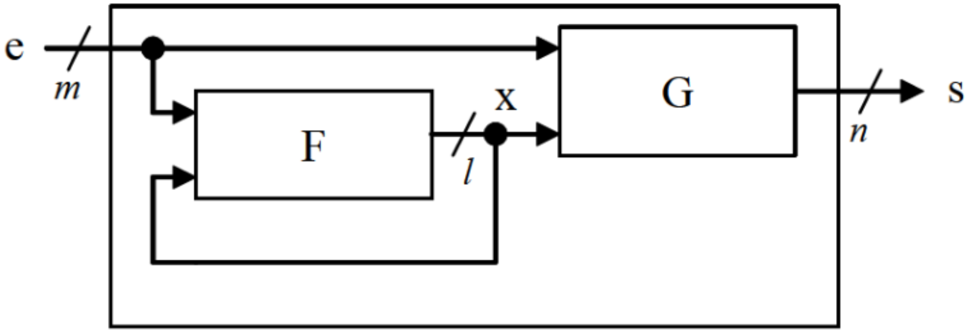


e_i : Entrée

s_i : Sortie

- Système logique **séquentiel**
  - Système dont la valeur $s_i[k]$ d'au moins une de ses sorties ne peut être déterminée uniquement à partir des seules valeurs de $e_j[k]$ à ce même instant k.
  - Ces sorties $s_i[k]$ dépendent également de la valeur antérieure de certaines entrées.
  - L'évolution des valeurs antérieures est vue par variables internes du système

$$\mathbf{s}_{\boldsymbol{n}}[\boldsymbol{k}] = \mathbf{F}_{\boldsymbol{n}}(\mathbf{e_1}[\boldsymbol{k}], \cdots, \mathbf{e}_{\boldsymbol{m}}[\boldsymbol{k}], \boldsymbol{x_1}[\boldsymbol{k}], \boldsymbol{x_2}[\boldsymbol{k}], \dots, \boldsymbol{x_l}[\boldsymbol{K}])$$



x_l : Variables internes

Structure d'algèbre de Boole

- Caractéristique
  - Une algèbre de Boole est une structure algébrique basée sur un ensemble d'éléments et comportant 3 opérations internes satisfaisant 9 axiomes.
    - **S** : Ensemble support de l'algèbre
    - **+,·** : Opérateurs binaires internes
    - : Opérateur unaire interne
    - 0,1 : Éléments particuliers de **S**

- Axiomes

$$x \cdot y = y \cdot x \quad x + y = y + x$$ Commutativité

$$x \cdot (y + z) = (x \cdot y) + (x \cdot z)$$
$$x + (y \cdot z) = (x + y) \cdot (x + z)$$ Distributivité

$$x \cdot 1 = x \quad x + 0 = x$$ Eléments neutres

$$x \cdot \overline{x} = 0 \quad x + \overline{x} = 1$$ Eléments complémentaires

$$0 \neq 1$$

Structure d'algèbre de Boole

- Algèbre des variables booléennes
  - Ensemble : $S = \{\,0\,,1\,\}$
  - Éléments particuliers : $0\,,1$
  - Opérations : $\vee,\ \wedge,\ \neg$

- Algèbre des sous-ensembles d'un ensemble E
  - Ensemble : $P(U) = \{\,\emptyset\,,\cdots,E\,\}$
  - Éléments particuliers : $\emptyset\,,E$
  - Opérations : $\cup,\ \cap, C_E$

Synthèse algébrique

Synthèse Algébrique

Données d'entrée

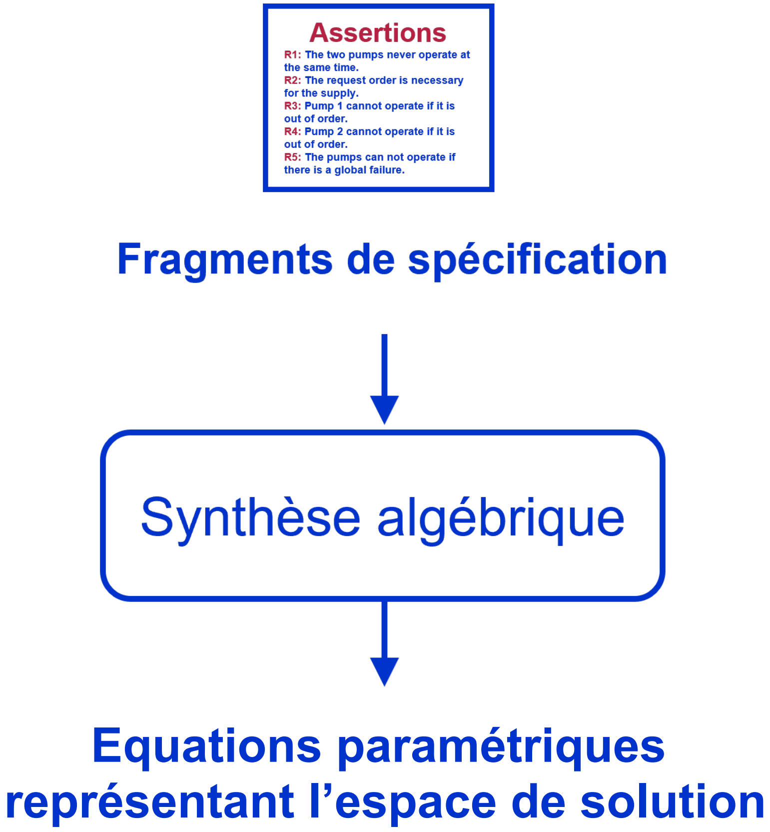
- Fragments de spécification donnés dans des formalismes différents (informel ou formel):
- Règles de sécurité,
- Fonctionnalités, …

Pourquoi une synthèse algébrique ?

- Limite l'explosion combinatoire
- Vérifie la cohérence des spécifications
- Pour exprimer toutes les conditions dans l'espace de solution

Résultat attendu

- Lois de commande à implanter



Synthèse Algébrique : formalisation des exigences

- Propositions simples
- Il **suffit** d'avoir 'A' pour avoir 'B'.

$$\boldsymbol{A} \leq \boldsymbol{B} \Leftrightarrow \boldsymbol{A} \Rightarrow \boldsymbol{B}$$

- Il **faut** avoir 'A' pour avoir 'B'.

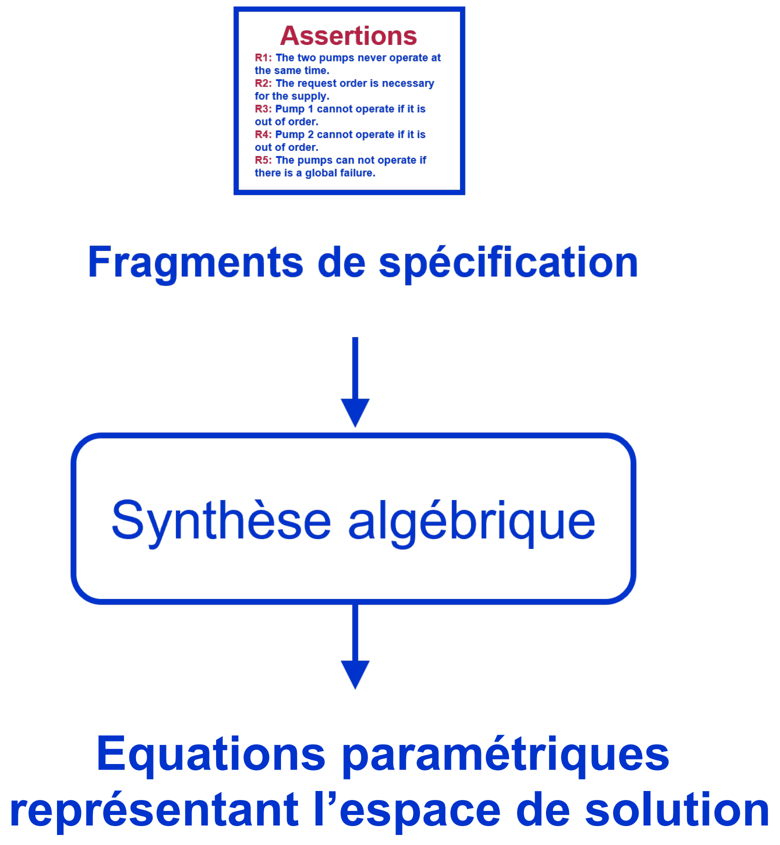
$$\boldsymbol{B} \leq \boldsymbol{A}$$

- Il est **nécessaire et suffisant** d'avoir 'A' pour avoir 'B'.

$$\begin{cases} \boldsymbol{A} \leq \boldsymbol{B} \\ \boldsymbol{B} \leq \boldsymbol{A} \end{cases} \Leftrightarrow \boldsymbol{A} = \boldsymbol{B}$$

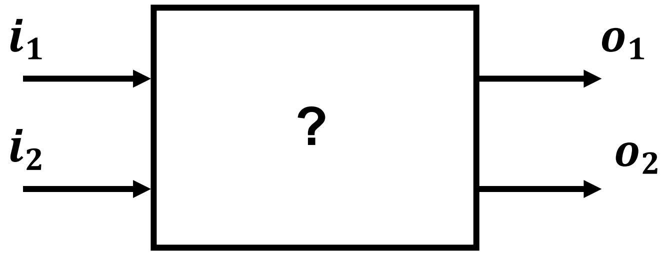
- 'A' et 'B' ne peuvent pas avoir lieu simultanément.

$$\boldsymbol{A} \cdot \boldsymbol{B} = 0$$



Synthèse Algébrique : intérêt

- **Comportement attendu**
  - R1 : Il suffit d'avoir 'i1' pour avoir 'o1'.
  - R2 : Il faut avoir 'i2' pour avoir 'o2'.
  - R3 : 'o1' et 'o2' ne sont jamais simultanément vrais.



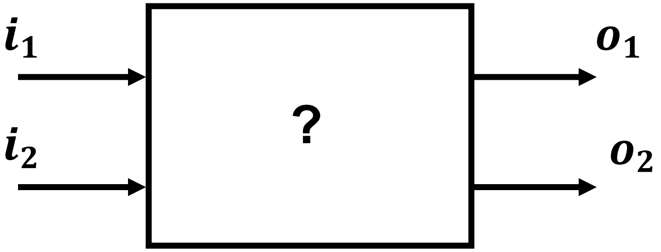
- **Approches classiques**
  - Identification d'une solution
    ⇒ Êtes-vous chanceux ?
  - Renseignement d'une table de vérité
    ⇒ Laborieux quand la taille augmente

i_1	i_2	o_1	o_2
0	0	x	0
0	1	x	x
1	0	1	0
1	1	1	0

- **Synthèse Algébrique**
  - Résolution d'un système d'équations entre des fonctions booléennes

Synthèse Algébrique : intérêt

- Comportement attendu
  - R1 : Il suffit d'avoir 'i1' pour avoir 'o1'.
  - R2 : Il faut avoir 'i2' pour avoir 'o2'.
  - R3 : 'o1' et 'o2' ne sont jamais simultanément vrais.



- Approches classiques
  - Identification d'une solution
    ⇒ Êtes-vous chanceux ?
  - Renseignement d'une table de vérité
    ⇒ Laborieux quand la taille augmente

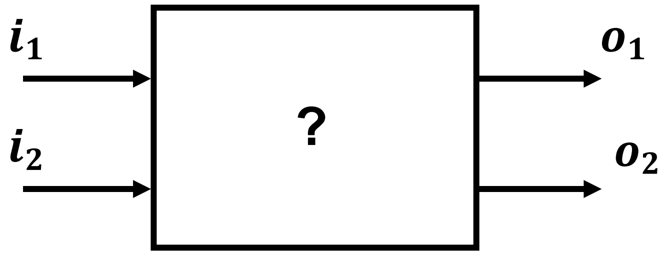
i_1	i_2	o_1	o_2
0	0	0	0
0	1	1	x
1	0	1	0
1	1	1	0

- Synthèse Algébrique
  - Résolution d'un système d'équations entre des fonctions booléennes

$$O_1 = i_1 + i_2$$
$$O_2 =$$

Synthèse Algébrique : intérêt

- Comportement attendu
  - R1 : Il suffit d'avoir 'i1' pour avoir 'o1'.
  - R2 : Il faut avoir 'i2' pour avoir 'o2'.
  - R3 : 'o1' et 'o2' ne sont jamais simultanément vrais.



- Approches classiques
  - Identification d'une solution
    ⇒ Êtes-vous chanceux ?
  - Renseignement d'une table de vérité
    ⇒ Laborieux quand la taille augmente

i_1	i_2	o_1	o_2
0	0	0	0
0	1	1	1
1	0	1	0
1	1	1	0

- Synthèse Algébrique
  - Résolution d'un système d'équations entre des fonctions booléennes

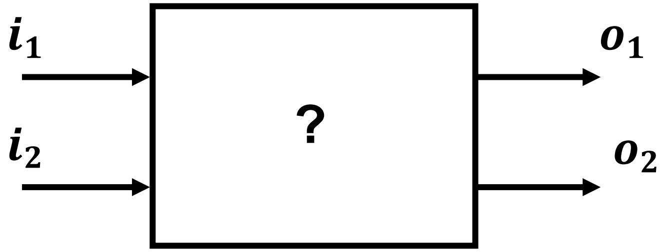
$$O_1 = i_1 + i_2$$
$$O_2 = \overline{i_1} \cdot i_2$$

Synthèse Algébrique : intérêt

- Comportement attendu
  - R1 : Il suffit d'avoir 'i1' pour avoir 'o1'.
  - R2 : Il faut avoir 'i2' pour avoir 'o2'.
  - R3 : 'o1' et 'o2' ne sont jamais simultanément vrais.

- Approches classiques
  - Identification d'une solution
    - ⇒ Êtes-vous chanceux ?   Nope… (6/256)
  - Renseignement d'une table de vérité
    - ⇒ Laborieux quand la taille augmente

- Synthèse Algébrique
  - Résolution d'un système d'équations entre des fonctions booléennes



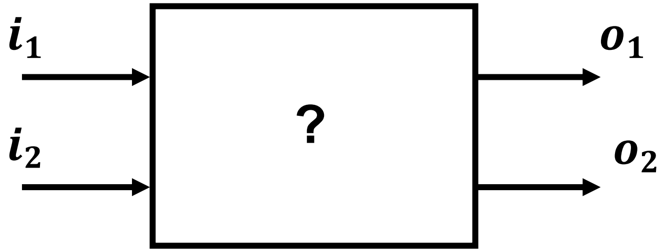
i_1	i_2	o_1	o_2
0	0	0	0
0	1	1	1
1	0	1	0
1	1	1	0

$$O_1 = i_1 + i_2$$
$$O_2 = \overline{i_1} \cdot i_2$$

Erreur : La valeur dépend de celles fixées précédemment !

Synthèse Algébrique : intérêt

- Comportement attendu
  - R1 : Il suffit d'avoir 'i1' pour avoir 'o1'.
  - R2 : Il faut avoir 'i2' pour avoir 'o2'.
  - R3 : 'o1' et 'o2' ne sont jamais simultanément vrais.
- Approches classiques
  - Identification d'une solution
    ⇒ Êtes-vous chanceux ? Nope… (6/256)
  - Renseignement d'une table de vérité
    ⇒ Laborieux quand la taille augmente
- Synthèse Algébrique
  - Résolution d'un système d'équations entre des fonctions booléennes



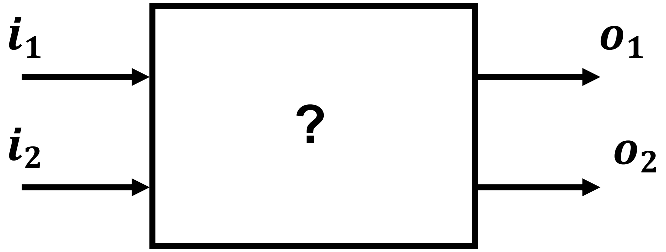
i_1	i_2	o_1	o_2
0	0	0	0
0	1	0	1
1	0	1	0
1	1	1	0

$$\cancel{o_1 = i_1 + i_2}$$
$$\cancel{o_2 = \bar{i_1} \cdot i_2}$$

$$o_1 = i_1$$
$$o_2 = \bar{i_1} \cdot i_2$$

Erreur : La valeur dépend de celles fixées précédemment !

Synthèse Algébrique : intérêt



- Comportement attendu
  - R1 : Il suffit d'avoir 'i1' pour avoir 'o1'.
  - R2 : Il faut avoir 'i2' pour avoir 'o2'.
  - R3 : 'o1' et 'o2' ne sont jamais simultanément vrais.

i_1	i_2	o_1	o_2
0	0	x	0
0	1	x	x
1	0	1	0
1	1	1	0

- Approches classiques
  - Identification d'une solution
    ⇒ Êtes-vous chanceux ?
  - Renseignement d'une table de vérité
    ⇒ Laborieux quand la taille augmente

- Synthèse Algébrique
  - Résolution d'un système d'équations entre des fonctions booléennes

$$\begin{cases} i_1 \leq O_1 \\ O_2 \leq i_2 \\ O_1 \cdot O_2 = 0 \end{cases} \Longrightarrow \begin{cases} O_1 = i_1 + \mathrm{P}_1 \\ O_2 = \overline{i_1} \cdot i_2 \cdot \overline{\mathrm{P}_1} \cdot \mathrm{P}_2 \end{cases}$$

La synthèse algébrique donne les équations paramétriques permettant d'exprimer toutes les solutions possibles

Synthèse algébrique : Principe de résolution

Tout part des équations booléennes :

➔Ordre 1 : $a \cdot \bar{x} + b \cdot x = 0$

➔Ordre 2 : $a \cdot \overline{x_1} \cdot \overline{x_2} + b \cdot \overline{x_1} \cdot x_2 + c \cdot x_1 \cdot \overline{x_2} + d \cdot x_1 \cdot x_2 = 0$

➔Ordre n : $C(a, b, c, \dots, x_1, x_2, \dots, x_m) = 0$

Synthèse algébrique : Principe de résolution

Tout part des équations booléennes :

➔Ordre 1 : $a \cdot \bar{x} + b \cdot x = 0$

➔Ordre 2 : $a \cdot \overline{x_1} \cdot \overline{x_2} + b \cdot \overline{x_1} \cdot x_2 + c \cdot x_1 \cdot \overline{x_2} + d \cdot x_1 \cdot x_2 = 0$

➔Ordre n : $C(a, b, c, \dots, x_1, x_2, \dots, x_m) = 0$

- **Pour une équation d'ordre 1** :
  il a été démontré que $x = a + \bar{b} \cdot p$ est une solution ssi $a \cdot b = 0$,
  avec p un paramètre binaire

Synthèse algébrique : Principe de résolution

Tout part des équations booléennes :

➔ Ordre 1 : $a \cdot \bar{x} + b \cdot x = 0$

➔ Ordre 2 : $a \cdot \overline{x_1} \cdot \overline{x_2} + b \cdot \overline{x_1} \cdot x_2 + c \cdot x_1 \cdot \overline{x_2} + d \cdot x_1 \cdot x_2 = 0$

➔ Ordre n : $C(a, b, c, \dots, x_1, x_2, \dots, x_m) = 0$

- **Pour une équation d'ordre 1** :

il a été démontré que $x = a + \bar{b} \cdot p$ est une solution ssi $a \cdot b = 0$, avec p un paramètre binaire

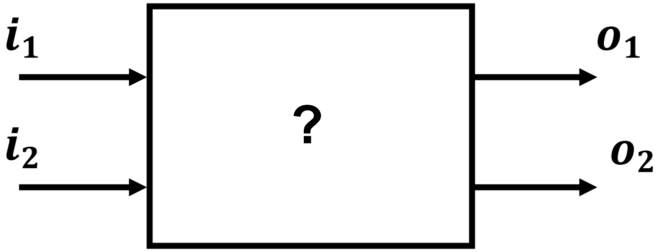
- **Pour une équation d'ordre 2 :**

Si nous posons $\begin{cases} a \cdot \overline{x_1} + c \cdot x_1 = a_2 \\ b \cdot \overline{x_1} + d \cdot x_1 = b_2 \end{cases}$ alors, nous devons résoudre $a_2 \cdot \overline{x_2} + b_2 \cdot x_2 \underset{?}{=} 0$ (éq. d'ordre 1)

Synthèse algébrique : Principe de résolution

Tout part des équations booléennes (contraintes)

- R1 : Il suffit d'avoir 'i1' pour avoir 'o1'. $\Rightarrow \; i1 \leq o1 \Rightarrow i1 \cdot \overline{o1} = 0$
- R2 : Il faut avoir 'i2' pour avoir 'o2'. $\Rightarrow \; o2 \leq i2 \Rightarrow o2 \cdot \overline{i2} = 0$
- R3 : 'o1' et 'o2' ne sont jamais simultanément Vrai. $\Rightarrow \; o1 \cdot o2 = 0$



Synthèse algébrique

- Nous voulons une unique équation à égalité 0

$\Rightarrow i1 \cdot \overline{o1} + o2 \cdot \overline{i2} + o1 \cdot o2 = 0$

- Identification des inconnues $\{x_1 = o1, x_2 = o2\}$

$\Rightarrow i1 \cdot \overline{x1} + \overline{i2} \cdot x2 + x1 \cdot x2 = 0$

- il faut le mettre sous la forme :

$a \cdot \overline{x_1} \cdot \overline{x_2} + b \cdot \overline{x_1} \cdot x_2 + c \cdot x_1 \cdot \overline{x_2} + d \cdot x_1 \cdot x_2 = 0$

Astuce :

$i1 \cdot \overline{x1} \cdot (\overline{x2} + x2) + \overline{i2} \cdot (\overline{x1} + x1) \cdot x2 + x1 \cdot x2 = 0$

$\Rightarrow i1 \cdot \overline{x1} \cdot \overline{x2} + (i1 + \overline{i2}) \cdot \overline{x1} \cdot x2 + 0 \cdot x1 \cdot \overline{x2} + 1 \cdot x1 \cdot x2 = 0$

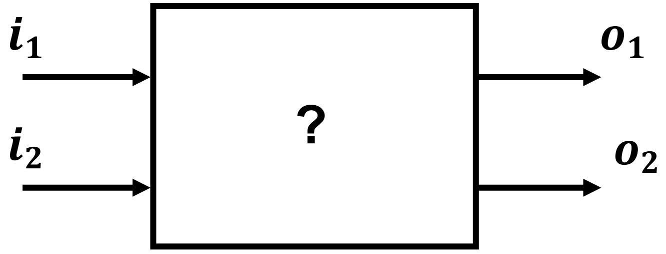
i_1	i_2	o_1	o_2
0	0	x	0
0	1	x	x
1	0	1	0
1	1	1	0

Synthèse algébrique : Principe de résolution

Synthèse algébrique

- il faut le mettre sous la forme :

$$a \cdot \overline{x_1} \cdot \overline{x_2} + b \cdot \overline{x_1} \cdot x_2 + c \cdot x_1 \cdot \overline{x_2} + d \cdot x_1 \cdot x_2 = 0$$



Astuce :

$$i1 \cdot \overline{x1} \cdot (\overline{x2} + x2) + \overline{i2} \cdot (\overline{x1} + x1) \cdot x2 + x1 \cdot x2 = 0$$

$$\Rightarrow i1 \cdot \overline{x1} \cdot \overline{x2} + (i1 + \overline{i2}) \cdot \overline{x1} \cdot x2 + 0 \cdot x1 \cdot \overline{x2} + 1 \cdot x1 \cdot x2 = 0$$

i_1	i_2	o_1	o_2
0	0	x	0
0	1	x	x
1	0	1	0
1	1	1	0

Solution paramétrique :

$$\begin{cases} o1 = i1 + \mathrm{P}_1 \\ o2 = \overline{i1} \cdot i2 \cdot \overline{\mathrm{P}_1} \cdot \mathrm{P}_2 \end{cases}$$

6 Solutions possibles

$$\begin{cases} \boldsymbol{O_1 = i_1} \\ \boldsymbol{O_2 = 0} \end{cases}$$

P1=0 & P2=0

$$\begin{cases} \boldsymbol{O_1 = i_1 + i_2} \\ \boldsymbol{O_2 = 0} \end{cases}$$

P1=i2 & P2=x

$$\begin{cases} \boldsymbol{O_1 = i_1} \\ \boldsymbol{O_2 = \overline{i_1} \cdot i_2} \end{cases}$$

P1=0 & P2=1

$$\begin{cases} \boldsymbol{O_1 = i_1 + \overline{i_2}} \\ \boldsymbol{O_2 = 0} \end{cases}$$

P1=/i2 & P2=0

$$\begin{cases} \boldsymbol{O_1 = 1} \\ \boldsymbol{O_2 = 0} \end{cases}$$

P1=/i1 & P2=x

$$\begin{cases} \boldsymbol{O_1 = i_1 + \overline{i_2}} \\ \boldsymbol{O_2 = \overline{i_1} \cdot i_2} \end{cases}$$

P1=/i2 & P2=1

Synthèse Algébrique : la meilleure solution

Données d'entrée

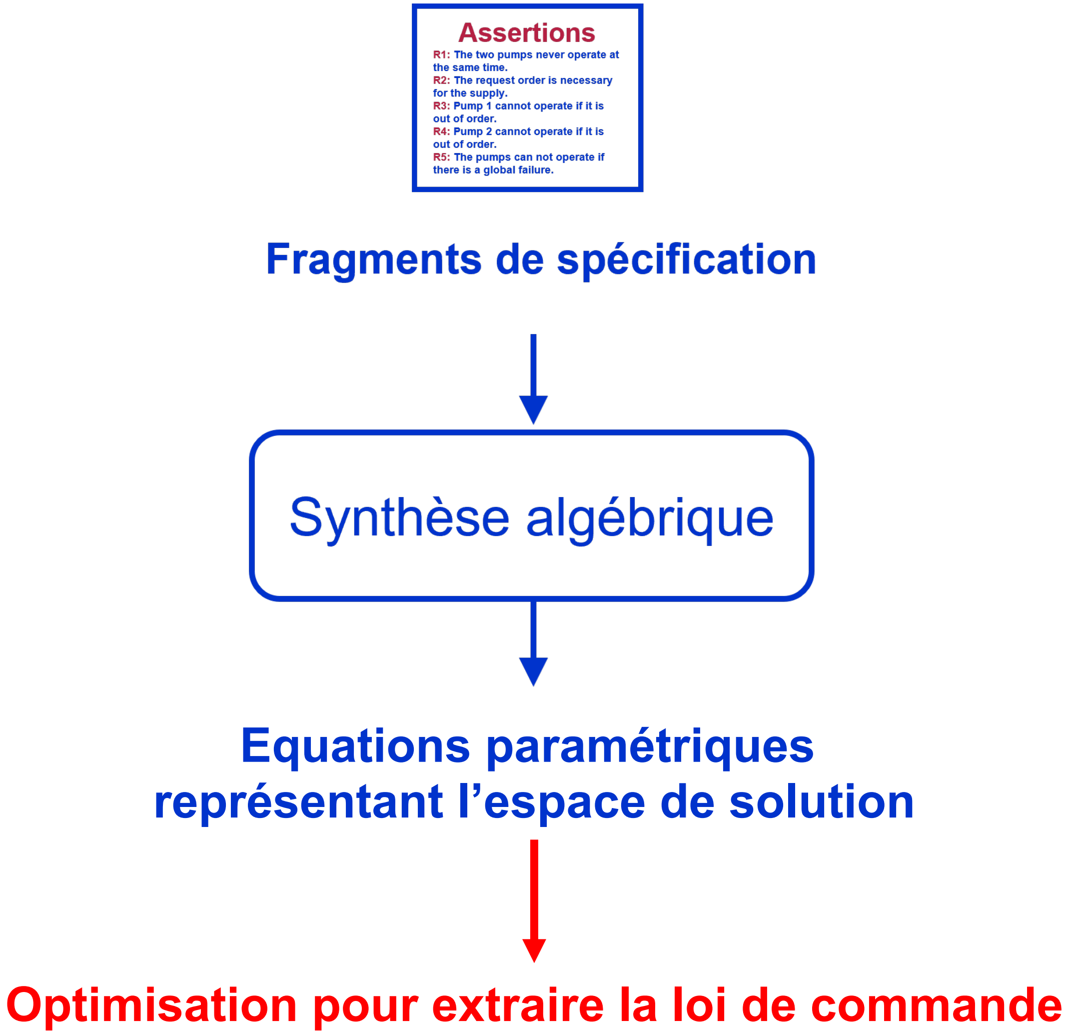
- Fragments de spécification donnés dans des formalismes différents (informel ou formel):
- Règles de sécurité,
- Fonctionnalités, …

Pourquoi une synthèse algébrique ?

- Limite l'explosion combinatoire
- Vérifie la cohérence des spécifications
- Pour exprimer toutes les conditions dans l'espace de solution

Résultat attendu

- Lois de commande à implanter

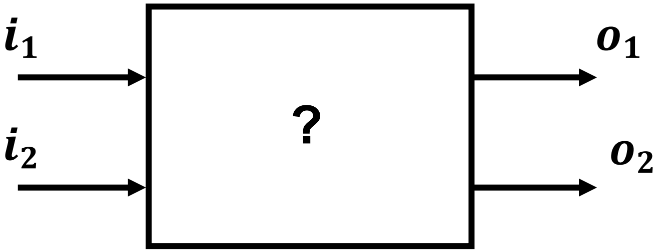


Synthèse Algébrique : la meilleure solution

Solution paramétrique

$$\begin{cases} o_1 = i_1 + P_1 \\ o_2 = \overline{i_1} \cdot i_2 \cdot \overline{P_1} \cdot P_2 \end{cases}$$

P_1	P_2	o_1	o_2
0	0	i_1	0
0	1	i_1	$\overline{i_1} \cdot i_2$
1	0	1	0
1	1	1	0



i_1	i_2	o_1	o_2
0	0	x	0
0	1	x	x
1	0	1	0
1	1	1	0

P1=0 & P2=0

Min o_1 & Min o_2

Min o_2 & Min o_1

i_1	i_2	o_1	o_2
0	0	0	0
0	1	0	0
1	0	1	0
1	1	1	0

P1=0 & P2=1

Max o_2 & Min o_1

Min o_1 & Max o_2

i_1	i_2	o_1	o_2
0	0	0	0
0	1	0	1
1	0	1	0
1	1	1	0

P1=/i1 & P2=1

Min o_2 & Max o_1

Max o_1 & Max o_2

Max o_1 & Min o_2

i_1	i_2	o_1	o_2
0	0	1	0
0	1	1	0
1	0	1	0
1	1	1	0

Importance de l'ordre !

P1=/i2 & P2=1

Max o_2 & Max o_1

i_1	i_2	o_1	o_2
0	0	1	0
0	1	0	1
1	0	1	0
1	1	1	0

Conclusion

Références bibliographiques

Références bibliographiques

- Hietter, Y. *Synthèse algébrique de lois de commande pour les systèmes à évènements discrets logiques.* Automatique / Robotique. École normale supérieure de Cachan - ENS Cachan, 2009. Français. ⟨NNT : ⟩. ⟨tel-00402699⟩
- Leroux, H. and Roussel, J.-M. *Algebraic synthesis of logical controllers with optimization criteria.* 6th International Workshop on Verification and Evaluation of Computer and Communication Systems VECOS 2012, Aug 2012, Paris, France. pp. 103-114. ⟨hal-00734840⟩
- Ranger, T. *Approche par synthèse algébrique et filtre logique pour la commande des systèmes manufacturiers cyber-physiques.* Automatique. Université de Reims Champagne Ardenne, 2023. Français. ⟨NNT : ⟩. ⟨tel-04255722⟩
- Ranger, T., Philippot, A. and Riera, B. *Manufacturing Tasks Synchronization by Algebraic Synthesis.* IFAC Conference on Embedded Systems, Computational Intelligence and Telematics in Control (CESCIT), 2021, Valenciennes, France. ⟨10.1016/j.ifacol.2021.10.038⟩. ⟨hal-03404408⟩
- Roussel, J.-M. and Lesage, J.-J. (2014). *Design of Logic Controllers Thanks to Symbolic Computation of Simultaneously Asserted Boolean Equations.* Mathematical Problems in Engineering, 2014. ⟨10.1155/2014/726246⟩. ⟨hal-01002261⟩

Equipe pédagogique

Equipe pédagogique

Auteurs : Gregory Faraut, Dimitri Renard, Bernard Riera, Jean-Marc Roussel

Intervenants : Gregory Faraut, Bernard Riera