Synthèse algébrique : mise en application

Formation Systèmes à Evénements Discrets

1ère édition
Janvier 2024

Société d'Automatique,
de Génie Industriel & de Productique

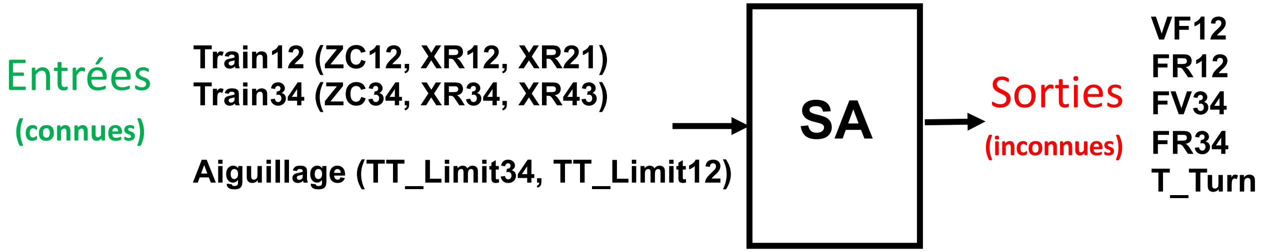
Plan

Plan

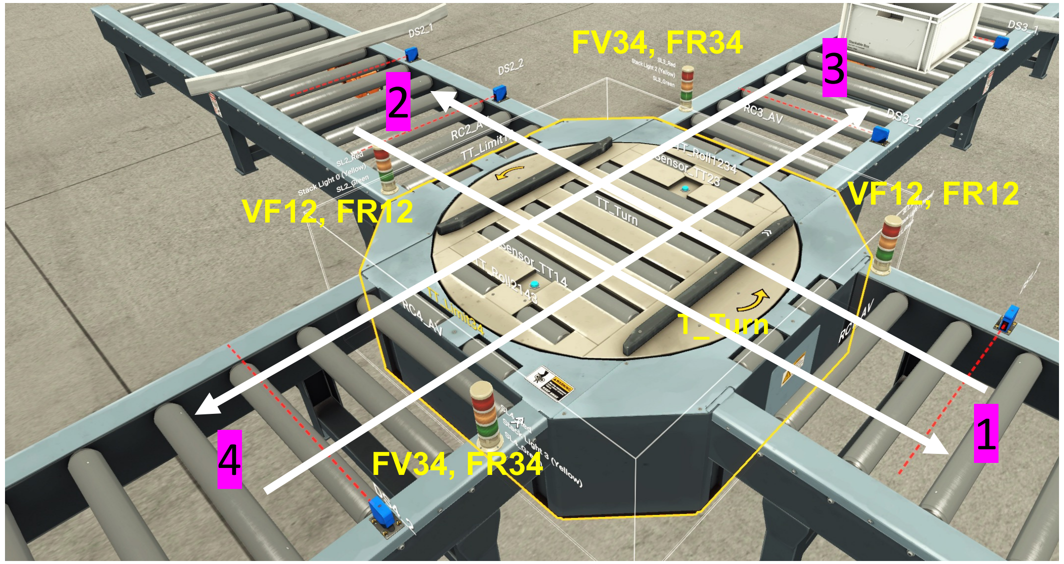
Introduction

Introduction

- Gestion de croisement (feux + « aiguillage ») de 2 « trains » (i.e. « caisses »)
  - La commande des feux et de l'aiguillage (« plateau tournant ») sont à réaliser par synthèse algébrique (SA)



FACTORY I/O



Introduction : synthèse algébrique

- 1) Formalisation du cahier des charges
  - Formalisation de chaque fragment donné de manière informelle
- 2) Vérification de la cohérence
  - Par calcul symbolique
- 3) Calcul des solutions
  - En résolvant analytiquement une équation entre des fonctions booléennes
- 4) Choix de la solution unique
  - En ajoutant des fragments de spécification (retour à l'étape 1)
  - En fixant une valeur spécifique à chaque paramètre
  - Utilisation de critères d'optimisation

Activité manuelle | Activité automatique

1 – Formalisation des exigences

Ensemble de relations entre des fonctions booléennes (fragments de spécification)

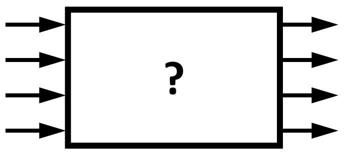
2 – Vérification de la cohérence

Oui	**Non**
Équation résoluble entre des fonctions booléennes	**Condition d'incohérence**

3 – Calcul des solutions possibles

Forme paramétrique des solutions

4 – Choix de la solution



Plan

Formalisation des exigences

2. Synthèse algébrique : formalisation des exigences

1 – Formalisation des exigences

Ensemble de relations entre des fonctions booléennes (fragments de spécification)

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

$$\boldsymbol{A \leq B \Leftrightarrow A \Rightarrow B \Leftrightarrow A./B = 0}$$

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

$$\boldsymbol{B \leq A \Leftrightarrow B \Rightarrow A \Leftrightarrow B./A = 0}$$

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

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

- A et B ne peuvent pas avoir lieu simultanément.

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

Assertions

R1: The two pumps never operate at the same time.
R2: The request order is necessary for the supply.
R3: Pump 1 cannot operate if it is out of order.
R4: Pump 2 cannot operate if it is out of order.
R5: The pumps can not operate if there is a global failure.

Fragments de spécification

Synthèse algébrique

Equations paramétriques représentant l'espace de solution

2. Synthèse algébrique : formalisation des exigences

- Rappel algèbre de Boole $\left(\mathbf{B},+,\cdot,\overline{},0,1\right)$

$$\begin{cases} C_1(b_1,\dots,b_n) \leq C_2(b_1,\dots,b_n) \\ C_3(b_1,\dots,b_n) = 0 \\ C_4(b_1,\dots,b_n) = 1 \\ C_5(b_1,\dots,b_n) = C_6(b_1,\dots,b_n) \end{cases} \Leftrightarrow \begin{cases} C_1(b_1,\dots,b_n) \cdot \overline{C_2(b_1,\dots,b_n)} = 0 \\ C_3(b_1,\dots,b_n) = 0 \\ \overline{C_4(b_1,\dots,b_n)} = 0 \\ C_5(b_1,\dots,b_n) \cdot \overline{C_6(b_1,\dots,b_n)} + \overline{C_5(b_1,\dots,b_n)} \cdot C_6(b_1,\dots,b_n) = 0 \end{cases}$$

$$\begin{cases} C_1(b_1,\dots,b_n) = 0 \\ C_2(b_1,\dots,b_n) = 0 \end{cases} \Leftrightarrow C_1(b_1,\dots,b_n) + C_2(b_1,\dots,b_n) = 0$$

Vérification de la cohérence

2. Synthèse algébrique : vérification de la cohérence

Variables inconnues Variables connues

Ordre n : $C(x_1, \dots, x_n, \alpha_1, \dots, \alpha_m) = 0$

2 – Vérification de la cohérence	
Oui **Équation résoluble entre des fonctions booléennes**	**Non** **Condition d'incohérence**

- Vérification de la cohérence :

$$\prod_{\omega_j \in \Omega_n} C\left(\omega_j^1, \dots, \omega_j^n, \alpha_1, \dots, \alpha_m\right) = 0$$

- ω_j : n-uplet Ω_n : ensemble des ω_j
- $\Omega_n = \left\{\omega_j = \left(\omega_j^1, \dots, \omega_j^k, \dots, \omega_j^n\right), \omega_j^k \in \{0,1\}\right\}$

- Exemple 2e ordre :

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

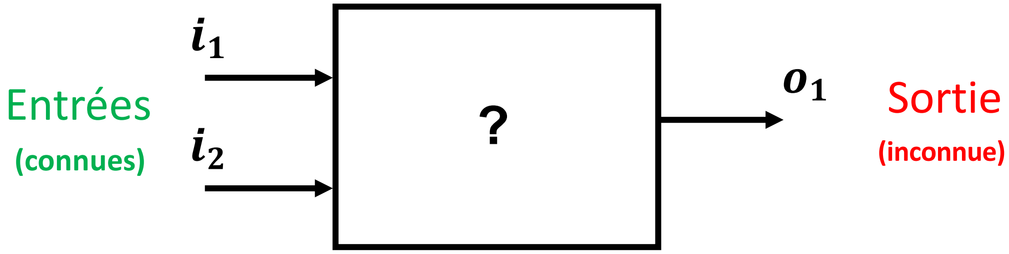
Équation résoluble *ssi* $a \cdot b \cdot c \cdot d = 0$

2. Synthèse algébrique : vérification de la cohérence

- Fragments de spécification
  - R1 : Il suffit d'avoir 'i1' pour avoir 'O1'

    $i_1 \Rightarrow O_1 \Leftrightarrow i_1 \le O_1 \Leftrightarrow i_1./O_1 = 0 \Leftrightarrow i_1.\overline{O_1} = 0$
  - R2 : Il suffit d'avoir 'i2' pour ne pas avoir 'O1'

    $i_2 \Rightarrow /O_1 \Leftrightarrow i_2 \le /O_1 \Leftrightarrow i_2 \le \overline{O_1} <=> i_2.O_1 = 0$



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

- Equation à résoudre

$$i_1.\overline{O_1} + i_2.O_1 = 0$$

- Condition de cohérence (hypothèse)

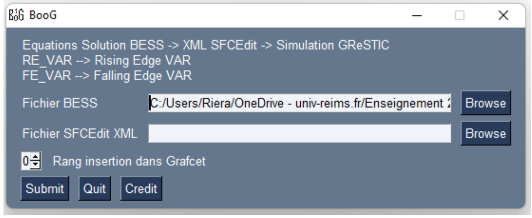
$$i_1.i_2 = 0$$

Outil logiciel BooG (basé sur le solveur BESS)

- Application

$$\begin{cases} i_1 \le O_1 \\ i_2 \le \overline{O_1} \end{cases}$$

Fichier exemple_coherence_NOK.txt



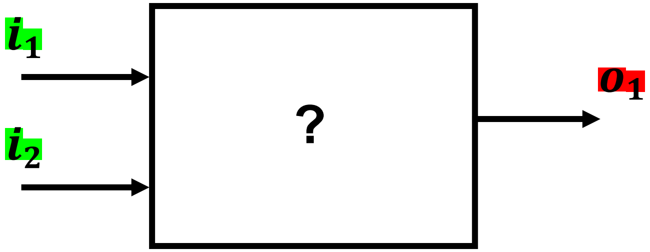
Fichier exemple_coherence_NOK_sol.txt

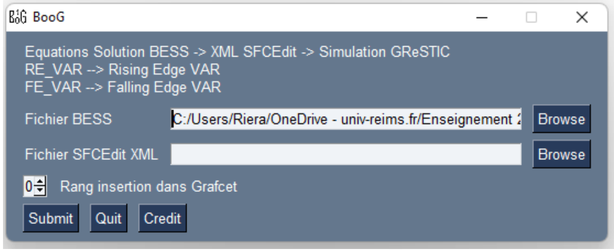
Répertoire : …\BooG 1.8.1\Exemple\Exemples_cours

2. Synthèse algébrique : vérification de la cohérence

```
<PROBLEM>
<SYMBOLS>
# Name : [Unknown|Known|Alias] (* Optional comment *) ;
  i1 : Known (**);
  i2 : Known (**);
  O1 : Unknown (* *) ;

</SYMBOLS>
<ALIASES>
# Name = BooleanFormula (* Optional comment *) ;
</ALIASES>
<REQUIREMENTS>
# Name : BooleanFormula [=<|=] BooleanFormula (* Optional comment *) ;
RR10 : (**)
i1./O1=0;

RR11 : (**)
i2.O1=0;

</REQUIREMENTS>
<ASSUMPTIONS>
# Name : (* Optional comment *)
# BooleanFormula = BooleanFormula ;
</ASSUMPTIONS>
<PRIORITIES>
# [Name|{Name, Name}] >> Name (* Optional comment *) ;
</PRIORITIES>
<OPTIMUM CRITERIA>
# Name : [Minimal|Maximal] (* Comment *)
#         BooleanFormula ;
</OPTIMUM CRITERIA>
</PROBLEM>
```

$$\begin{cases} i_1 \leq o_1 \\ i_2 \leq \overline{o_1} \end{cases}$$






```
Global Boolean equations system building:
* SubProblem 1:
  - Number of unknowns: 1
  - Number of optimum criteria: 0
  - Size of the BDD which represents the Boolean equation: 5
  - Time spent: 0 ms
- Time spent for all systems building: 0 ms
Results:
* SubProblem 1:
- The following consistent condition must be equal to 0.
    Consistent condition = i1.i2

Generation of the code :

Global Boolean equation solving:
* SubProblem 1:
  - The Boolean equations system has no solution.
  - Time spent for resolution : 0 ms
- Time spent for all resolutions : 0 ms


Details of BDD operations:
- BDD nodes stored     :   16
- Operations stored    :    4
- Operations not made  :    0
```

Fichier exemple_coherence_NOK.txt

Fichier exemple_coherence_NOK_sol.txt

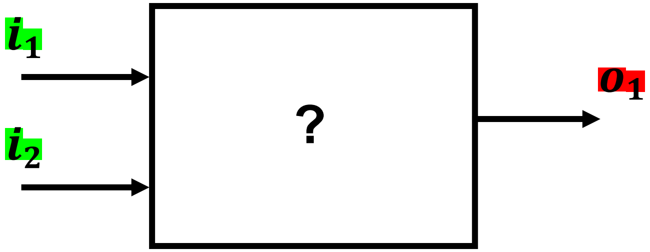
2. Synthèse algébrique : vérification de la cohérence

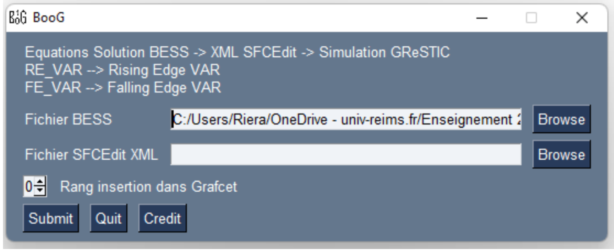
```
<PROBLEM>
<SYMBOLS>
# Name : [Unknown|Known|Alias] (* Optional comment *) ;
  i1 : Known (**);
  i2 : Known (**);
  O1 : Unknown (* *);

</SYMBOLS>
<ALIASES>
# Name = BooleanFormula (* Optional comment *) ;
</ALIASES>
<REQUIREMENTS>
# Name : BooleanFormula [=<|=] BooleanFormula (* Optional comment *) ;
RR10 : (**)
i1./O1=0;

RR11 : (**)
i2.O1=0;
</REQUIREMENTS>
<ASSUMPTIONS>
# Name :  (* Optional comment *)
# BooleanFormula = BooleanFormula ;
A0: (**)
i1.i2=0;
</ASSUMPTIONS>
<PRIORITIES>
# [Name|{Name, Name}] >> Name (* Optional comment *) ;
</PRIORITIES>
<OPTIMUM CRITERIA>
# Name : [Minimal|Maximal] (* Comment *)
#         BooleanFormula ;
</OPTIMUM CRITERIA>
</PROBLEM>
```

$$\begin{cases} i_1 \leq O_1 \\ i_2 \leq \overline{O_1} \end{cases}$$

$$i_1 . i_2 = 0$$

Fichier exemple_coherence_OK.txt






```
Global Boolean equations system building:
- Size of the BDD which represents the global assumption: 4
* SubProblem 1:
  - Number of unknowns: 1
  - Number of optimum criteria: 0
  - Size of the BDD which represents the Boolean equation: 5
  - Time spent: 0 ms
- Time spent for all systems building: 0 ms
Results:
* SubProblem 1:
  - Solutions:
[Sev.]          O1 = i1+p_O1.(/i2)
```

Fichier exemple_coherence_OK_sol.txt

$$i_1 . \overline{O_1} + i_2 . O_1 = 0$$

$$O_1 = i_1 + \overline{i_2} \cdot p \text{ est une solution } ssi \; i_1 . i_2 = 0$$

Calcul des solutions possibles

2. Synthèse algébrique : calcul des solutions possibles

3 – Calcul des solutions possibles

Forme paramétrique des solutions

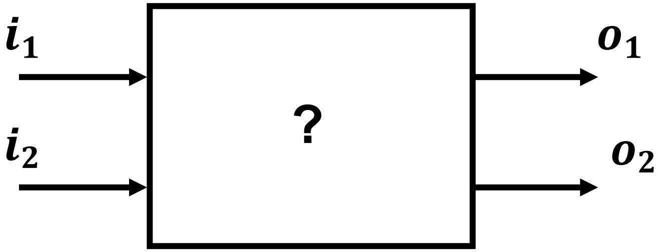
On veut résoudre $C_0(x_1, \dots, x_n, \alpha_1, \dots, \alpha_m) = 0$

Variables inconnues ($x_1, \dots, x_n$) — Variables connues ($\alpha_1, \dots, \alpha_m$)

$$C_i(x_{i+1}, \dots, x_n, \alpha_1, \dots, \alpha_m) = C_{i-1}\,(x_i, x_{i+1}, \dots, x_n, \alpha_1, \dots, \alpha_m)\Big|_{x_i \leftarrow S(x_i),\, i>0}$$

$$S(x_i) = \prod_{\substack{\omega_j \in \Omega_{n+1-i} \\ \omega_j^1 = 0}} C_{i-1}(\omega_j^1, \dots, \omega_j^{n+1-i}, \alpha_1, \dots, \alpha_m) + \overline{\prod_{\substack{\omega_j \in \Omega_{n+1-i} \\ \omega_j^1 = 1}} C_{i-1}(\omega_j^1, \dots, \omega_j^{n+1-i}, \alpha_1, \dots, \alpha_m)} \cdot p_i$$

2. Synthèse algébrique : calcul des solutions possibles



- Comportement souhaité
  - R1 : Il suffit d'avoir 'i1' pour avoir 'O1'

    $i_1 \Rightarrow O_1$
  - R2 : Il faut avoir 'i2' pour avoir 'O2'

    $O_2 \Rightarrow i_2$
  - R3 : 'O1' et 'O2' ne sont jamais simultanément vrais

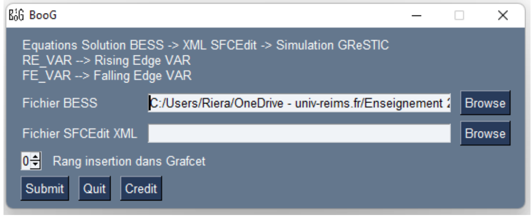
    $O_1 . O_2 = 0$

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
  - Résolution d'un système d'équations de fonctions booléennes
- Solution paramétrique

$$\begin{cases} i_1 \leq O_1 \\ O_2 \leq i_2 \\ O_1 \cdot O_2 = 0 \end{cases}$$

Outil logiciel BooG (BESS)

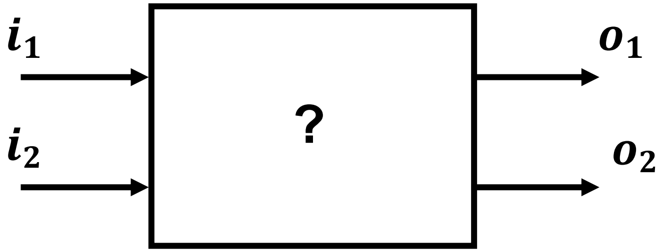


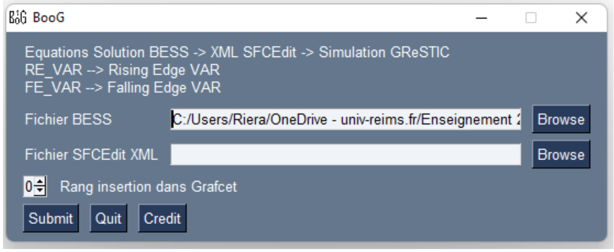
$$\begin{cases} O_1 = i_1 + \mathrm{P}_1 \\ O_2 = \overline{i_1} \cdot i_2 \cdot \overline{\mathrm{P}_1} \cdot \mathrm{P}_2 \end{cases}$$

2. Synthèse algébrique : calcul des solutions possibles

```
<PROBLEM>
<SYMBOLS>
# Name : [Unknown|Known|Alias] (* Optional comment *) ;
  i1 : Known (**);
  i2 : Known (**);
  O1 : Unknown (* *);
  O2 : Unknown (* *);
</SYMBOLS>
<ALIASES>
# Name = BooleanFormula (* Optional comment *) ;
</ALIASES>
<REQUIREMENTS>
# Name : BooleanFormula [=<|=] BooleanFormula (* Optional comment *) ;
RR10 : (**)
i1./O1=0;

RR11 : (**)
O2./i2=0;

RR12 : (**)
O1.O2=0;
</REQUIREMENTS>
<ASSUMPTIONS>
# Name :  (* Optional comment *)
# BooleanFormula = BooleanFormula ;
</ASSUMPTIONS>
<PRIORITIES>
# [Name|{Name, Name}] >> Name (* Optional comment *) ;
</PRIORITIES>
<OPTIMUM CRITERIA>
# Name : [Minimal|Maximal] (* Comment *)
#          BooleanFormula ;
</OPTIMUM CRITERIA>
</PROBLEM>
```

$$\begin{cases} i_1 \le o_1 \\ o_2 \le i_2 \\ o_1 \cdot o_2 = 0 \end{cases}$$






```
Global Boolean equations system building:
* SubProblem 1:
  - Number of unknowns: 2
  - Number of optimum criteria: 0
  - Size of the BDD which represents the Boolean equation: 8
  - Time spent: 0 ms
- Time spent for all systems building: 0 ms
Results:
* SubProblem 1:
  - Solutions:
[Sev.]            O1 = i1+p_O1
[Sev.]            O2 = p_O2.(/p_O1./i1.i2)

Generation of the code :
O1 := i1 OR p_O1;
O2 := p_O2 AND (NOT p_O1 AND NOT i1 AND i2);

Global Boolean equation solving:
* SubProblem 1:
  - The Boolean equations system has solutions.
  - Time spent for resolution : 0 ms
- Time spent for all resolutions : 0 ms


Details of BDD operations:
- BDD nodes stored     :   42
- Operations stored    :   19
- Operations not made  :    0
```

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

Fichier exemple_cours.txt

Fichier exemple_cours_sol.txt

Choix de la solution

2. Synthèse algébrique : choix de la solution

4 – Choix de la solution

$$S(x_i) = \prod_{\substack{\omega_j \in \Omega_{n+1-i} \\ \omega_j^1 = 0}} C_{i-1}\left(\omega_j^1, \dots, \omega_j^{n+1-i}, \alpha_1, \dots, \alpha_m\right) + \overline{\prod_{\substack{\omega_j \in \Omega_{n+1-i} \\ \omega_j^1 = 1}} C_{i-1}\left(\omega_j^1, \dots, \omega_j^{n+1-i}, \alpha_1, \dots, \alpha_m\right)} \cdot p_i$$

Comment obtenir une solution unique (déterministe, implémentable)?

- En enrichissant la spécification
- En définissant les p_i, mais comment ?
  - De façon intuitive…
  - **En utilisant des critères d'optimisation**
    - **Minimiser une fonction booléenne**
    - **Maximiser une fonction booléenne**

2. Synthèse algébrique : choix de la solution

4 – Choix de la solution

Variables inconnues Variables connues

$\mathrm{F}(p_1, \dots, p_k, \alpha_1, \dots, \alpha_m)$: Critère d'optimisation

- Minimum d'une fonction booléenne

$$Min\big(\mathrm{F}(p_1, \dots, p_k, \alpha_1, \dots, \alpha_m)\big)$$

$$= \prod_{p_k \in \{0,1\}} \mathrm{F}(p_1, \dots, p_k, \alpha_1, \dots, \alpha_m)$$

- Maximum d'une fonction booléenne

$$Max\big(\mathrm{F}(p_1, \dots, p_k, \alpha_1, \dots, \alpha_m)\big)$$

$$= \sum_{p_k \in \{0,1\}} \mathrm{F}(p_1, \dots, p_k, \alpha_1, \dots, \alpha_m)$$

2. Synthèse algébrique : choix de la solution

Choix d'une solution : pour chaque critère d'optimisation (Min, Max)

Méthode :

- ① : Calcul des solutions paramétriques $S_{x_i}(p_i)$ sans appliquer le critère d'optimisation
- ② : Calcul du critère d'optimisation en remplaçant les variables inconnues x_i par les solutions précédemment calculées
- ③ : Ajout du critère d'optimisation dans l'équation et résolution de celle-ci afin d'obtenir $S_{x_i}(p_i')$
- ④: Retour à l'étape ② pour appliquer un nouveau critère d'optimisation avec les nouveaux paramètres p_i'

2. Synthèse algébrique : choix de la solution

Solution paramétrique

$$\begin{cases} o_{1p} = i_1 + P_1 \\ o_{2p} = \overline{i_1} \cdot i_2 \cdot \overline{P_1} \cdot P_2 \end{cases}$$

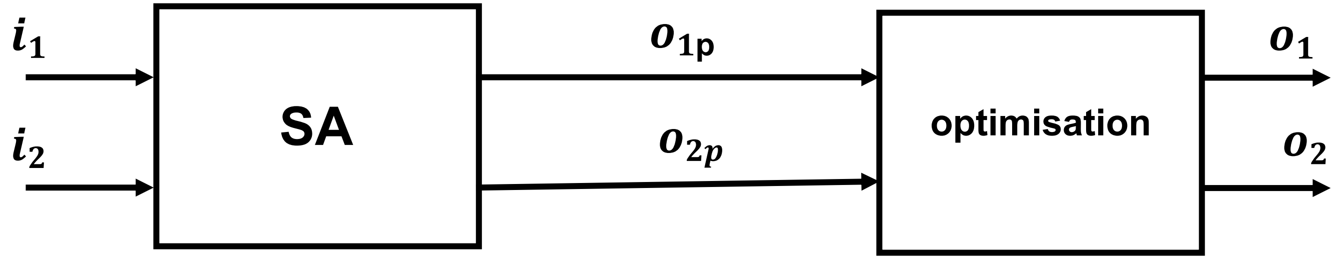
P_1	P_2	o_{1p}	o_{2p}
0	0	i_1	0
0	1	i_1	$/i_1.i_2$
1	0	1	0
1	1	1	0

Min O_1
Max O_2

Solution unique

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

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



2. Synthèse algébrique : choix de la solution

Choix d'une solution : pour chaque critère d'optimisation

Min $\boldsymbol{O_1}$
Max $\boldsymbol{O_2}$

Méthode :

- ① : Calcul des solutions paramétriques $S_{x_i}(p_i)$ sans appliquer le critère d'optimisation

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

- ② : Calcul du critère d'optimisation en remplaçant les variables inconnues x_i par les solutions précédemment calculées

Min $\boldsymbol{O_1} = \boldsymbol{(i_1 + 0).(i_1 + 1) = i_1}$

- ③ : Ajout du critère d'optimisation dans l'équation et résolution de celle-ci afin d'obtenir $S_{x_i}(p_i')$

$$\boldsymbol{O_1 = i_1 + \mathrm{P}_1 = i_1 \Leftrightarrow \mathrm{P}_1 \cdot \overline{i_1} = 0 \Rightarrow \mathrm{P}_1 = i_1 \cdot \mathrm{P}_1'}$$

Rappel : $a = b \Leftrightarrow a \cdot \bar{b} + \bar{a} \cdot b = 0$

Rappel ordre 1 : $a \cdot \bar{x} + b \cdot x = 0$
$ssi\ a.b = 0,\ x = a + \bar{b} \cdot p$

$$\begin{cases} \boldsymbol{O_1 = i_1 + \mathrm{P}_1 = i_1 + i_1 \cdot \mathrm{P}_1' = i_1} \\ \boldsymbol{O_2 = \overline{i_1} \cdot i_2 \cdot \mathrm{P}_2 \cdot \overline{\mathrm{P}_1} = \overline{i_1} \cdot i_2 \cdot \mathrm{P}_2 \cdot \overline{(i_1 \cdot \mathrm{P}_1')} = \overline{i_1} \cdot i_2 \cdot \mathrm{P}_2} \end{cases}$$

- ④: Retour à l'étape ② pour appliquer un nouveau critère d'optimisation avec les nouveaux paramètres p_i'

$$\begin{cases} \boldsymbol{O_1 = i_1} \\ \boldsymbol{O_2 = \overline{i_1} \cdot i_2 \cdot \mathrm{P}_2} \end{cases}$$

2. Synthèse algébrique : choix de la solution

Choix d'une solution : pour chaque critère d'optimisation

Min $\boldsymbol{O_1}$
Max $\boldsymbol{O_2}$

Méthode :

- ① : Calcul des solutions paramétriques $S_{x_i}(p_i)$ sans appliquer le critère d'optimisation $\begin{cases} \boldsymbol{O_1 = i_1} \\ \boldsymbol{O_2 = \overline{i_1} \cdot i_2 \cdot \mathrm{P}_2} \end{cases}$

- ② : Calcul du critère d'optimisation en remplaçant les variables inconnues x_i par les solutions précédemment calculées

$$\text{Max } \boldsymbol{O_2} = (\boldsymbol{\overline{i_1} \cdot i_2 \cdot 0}) + (\boldsymbol{\overline{i_1} \cdot i_2 \cdot 1}) = \boldsymbol{\overline{i_1} \cdot i_2}$$

- ③ : Ajout du critère d'optimisation dans l'équation et résolution de celle-ci afin d'obtenir $S_{x_i}(p_i')$

$$\boldsymbol{O_2 = \overline{i_1} \cdot i_2 \cdot \mathrm{P}_2 = \overline{i_1} \cdot i_2 \cdot \overline{i_1} \cdot i_2 = \overline{i_1} \cdot i_2}$$
$$\Leftrightarrow \boldsymbol{\overline{i_1} \cdot i_2 . \overline{\mathrm{P}_2} = 0} \Rightarrow \mathrm{P}_2 = \boldsymbol{\overline{i_1} \cdot i_2}$$

$$\begin{cases} \boldsymbol{O_1 = i_1} \\ \boldsymbol{O_2 = \overline{i_1} \cdot i_2 \cdot \mathrm{P}_2 = \overline{i_1} \cdot i_2 \cdot \overline{i_1} \cdot i_2 = \overline{i_1} \cdot i_2} \end{cases} \qquad \begin{cases} \boldsymbol{O_1 = i_1} \\ \boldsymbol{O_2 = \overline{i_1} \cdot i_2} \end{cases}$$

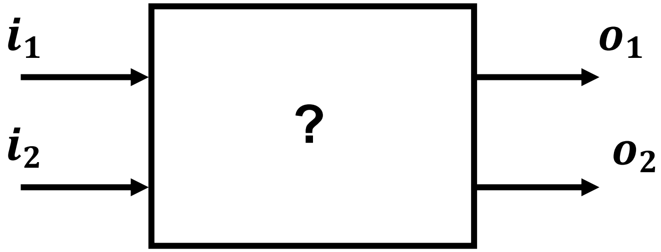
- ④: Retour à l'étape ② pour appliquer un nouveau critère d'optimisation avec les nouveaux paramètres p_i'

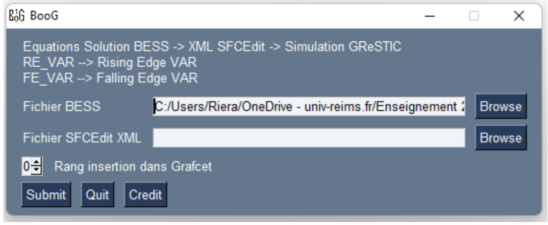
2. Synthèse algébrique : choix de la solution

```
<PROBLEM>
<SYMBOLS>
# Name : [Unknown|Known|Alias] (* Optional comment *) ;
  i1 : Known (**);
  i2 : Known (**);
  O1 : Unknown (* *);
  O2 : Unknown (* *);
</SYMBOLS>
<ALIASES>
# Name = BooleanFormula (* Optional comment *) ;
</ALIASES>
<REQUIREMENTS>
# Name : BooleanFormula [=<|=] BooleanFormula (* Optional comment *) ;
RR10 : (**)
i1./O1=0;

RR11 : (**)
O2./i2=0;

RR12 : (**)
O1.O2=0;
</REQUIREMENTS>
<ASSUMPTIONS>
# Name :  (* Optional comment *)
# BooleanFormula = BooleanFormula ;
</ASSUMPTIONS>
<PRIORITIES>
# [Name|{Name, Name}] >> Name (* Optional comment *) ;
</PRIORITIES>
<OPTIMUM CRITERIA>
# Name : [Minimal|Maximal] (* Comment *)
#        BooleanFormula ;
OC1 : Minimal (**)
O1;

OC2 : Maximal (**)
O2;
</OPTIMUM CRITERIA>
</PROBLEM>
```

$$\begin{cases} i_1 \leq O_1 \\ O_2 \leq i_2 \\ O_1 \cdot O_2 = 0 \end{cases}$$

Fichier exemple_cours_opt.txt





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

```
Global Boolean equations system building:
* SubProblem 1:
  - Number of unknowns: 2
  - Number of optimum criteria: 2
  - Size of the BDD which represents the Boolean equation: 14
  - Time spent: 0 ms
- Time spent for all systems building: 0 ms
Results:
* SubProblem 1:
  - Solutions:
[One]                O1 = i1
[One]                O2 = /i1.i2
  - Optimum criteria:
[Min]                OC1 = i1
[Max]                OC2 = /i1.i2

Generation of the code :
O1 := i1;
O2 := NOT i1 AND i2;

Global Boolean equation solving:
* SubProblem 1:
  - The Boolean equations system has solutions.
  - Time spent for resolution : 0 ms
- Time spent for all resolutions : 0 ms


Details of BDD operations:
- BDD nodes stored     :  102
- Operations stored    :   55
- Operations not made  :    7
```

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

Fichier exemple_cours_opt_sol.txt

Plan

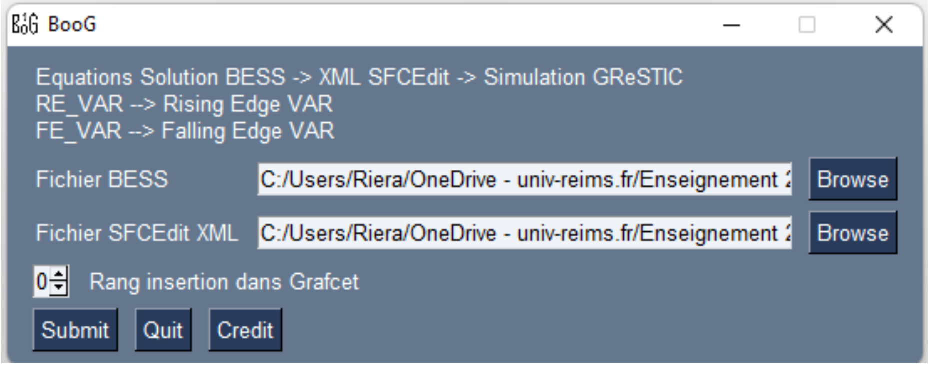
Méthodologie

3. Gestion de feux et d'aiguillage : méthodologie

3 logiciels fournis

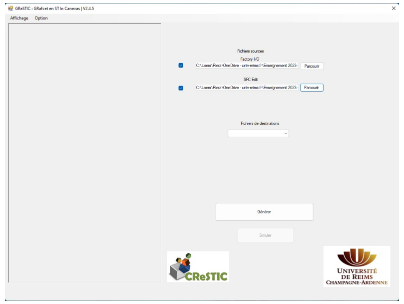
- BooG : solveur BESS

...\BooG 1.8.1 BooG_V1.8.1.exe



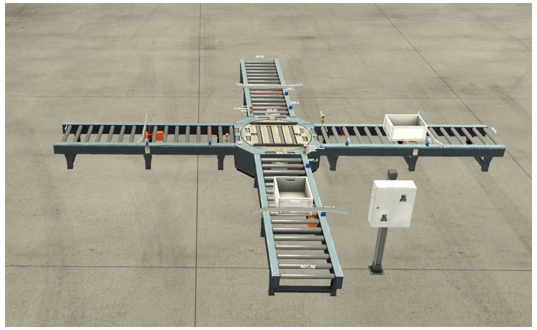
- GReSTIC : simulateur Partie Commande (PC)

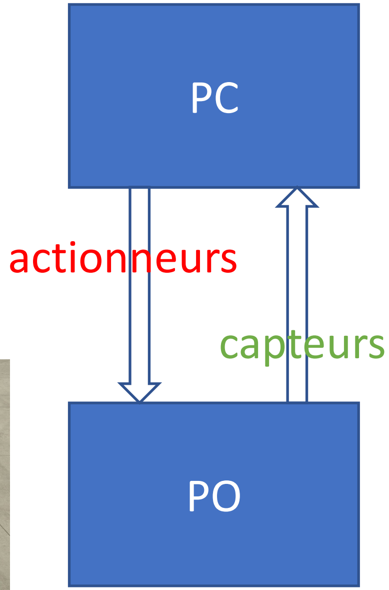
...\GReSTIC 2.4.5 GReSTIC.exe



- FACTORY I/O Simulateur Partie Opérative (PO)

  À installer
  Share code = KRC-1SM-HQH
  Validité : 2 mars 2024





3. Gestion de feux et d'aiguillage : méthodologie

Synthèse Algébrique
(aiguillage + feux)

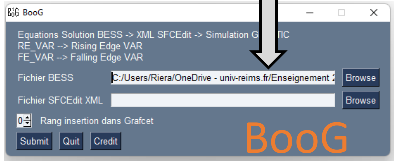
raffinement / optimisation

Assertions

R1: The two pumps never operate at the same time.
R2: The request order is necessary for the supply.
R3: Pump 1 cannot operate if it is out of order.
R4: Pump 2 cannot operate if it is out of order.
R5: The pumps can not operate if there is a global failure.

Fragments de spécification

2trains_SA_AR_RC_Vx.txt (BESS)



2trains_SA_AR_RC_Vx_sol.txt (BESS)

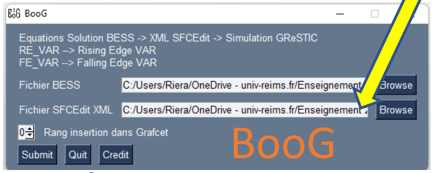
cohérence ?
[sev] ?

correction / raffinement / optimisation

[one] ?

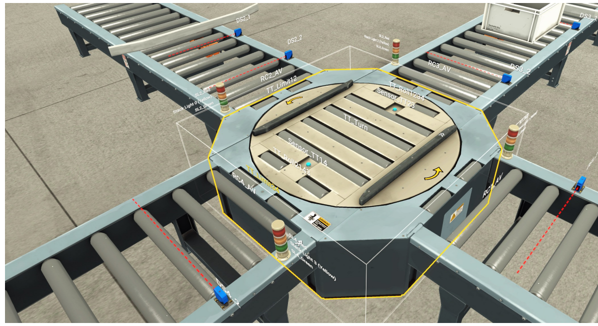
Commande des 2 trains

2trains_G7_AR_RC.xml

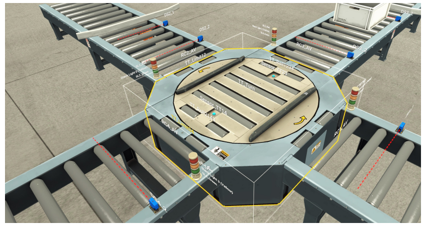


2trains_SA_AR_RC_Vx_sol_Export_complet.xml

FACTORY I/O

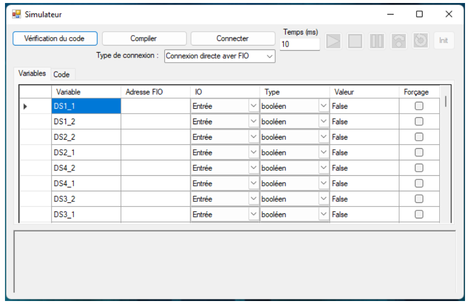


2trains_V1.factoryio



FACTORY I/O

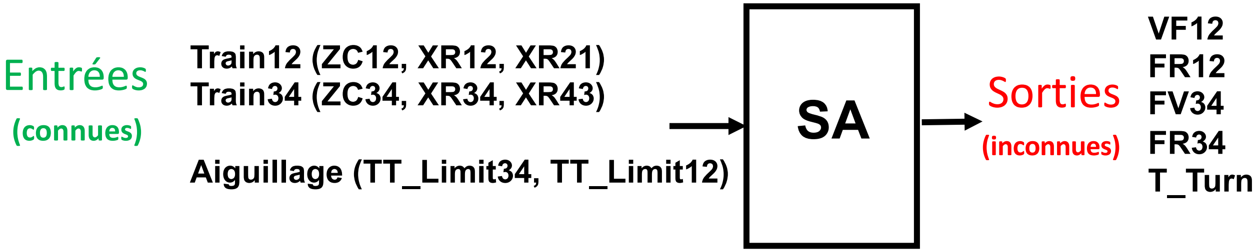
Simulation / test

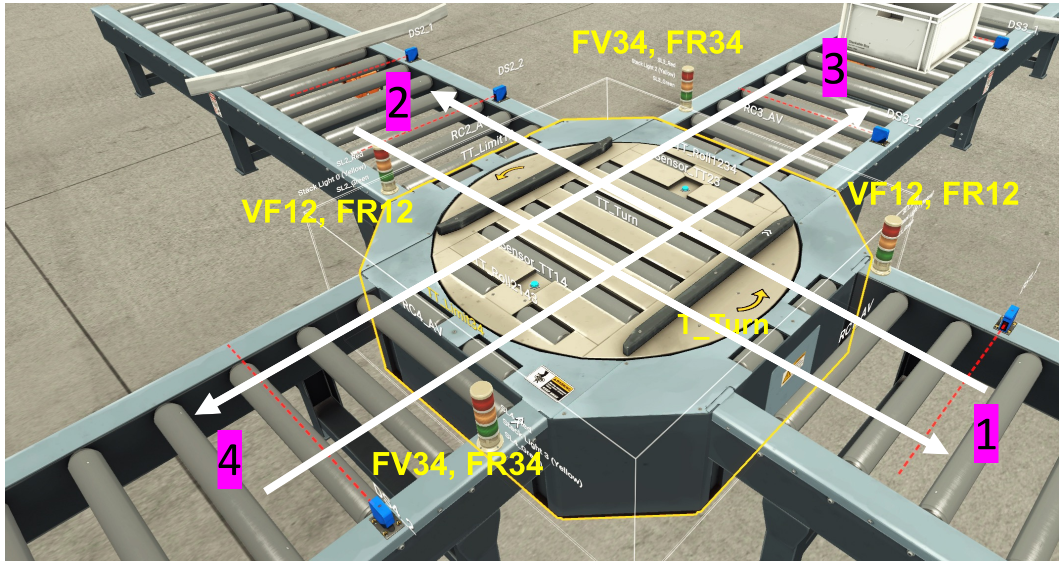


Travail personnel

3. Gestion de feux et d’aiguillage : travail personnel

- Gestion de croisement (feux + « aiguillage ») de 2 « trains » (i.e. « caisses »)
  - La commande des feux et de l’aiguillage (« plateau tournant ») sont à réaliser par synthèse algébrique (SA)





3. Gestion de feux et d'aiguillage : travail personnel

- Cahier des charges
  - fragments de spécification
    - Le feu est soit vert, soit rouge
    - Les 2 trains ne peuvent pas être autorisés simultanément à traverser la zone commune
    - L'autorisation est donnée si un train est détecté
    - Pour que le feu soit vert, il faut avoir l'autorisation de traverser et que l'aiguillage soit bien positionné
    - Le feu reste vert pendant que le train traverse
    - …

3. Gestion de feux et d’aiguillage : travail personnel

- Partie 1
  - Répertoire : BooG\Exemple\2trains_G7_SA
    - Ouvrir et analyser le fichier BESS : 2trains_SA_AR_RC_V0.txt
    - Générer avec BooG le fichier 2trains_SA_AR_RC_V0_sol.txt
      - Conclusion (cf. condition de cohérence) ?
    - Ouvrir et analyser le fichier BESS : 2trains_SA_AR_RC_V1.txt
      - Quelles sont les différences (cf. hypothèses) avec le fichier 2trains_SA_AR_RC_V2.txt
    - Générer avec BooG le fichier 2trains_SA_AR_RC_V1_sol.txt
      - Conclusion (cf. solutions) ?
    - Ouvrir et analyser le fichier BESS : 2trains_SA_AR_RC_V2.txt
      - Quelles sont les différences (cf. optimisation) avec le fichier 2trains_SA_AR_RC_V1.txt
    - Générer avec BooG le fichier 2trains_SA_AR_RC_V2_sol.txt
      - Conclusion ?
    - Ajouter le fichier 2trains_G7_AR_RC.xml et générer avec BooG le fichier 2trains_SA_AR_RC_V2_sol _Export_complet.txt
    - Ouvrir le fichier 2trains_V1.factoryio avec FACTORY I/O
    - Simuler et tester avec GReSTIC, en sélectionnant les fichiers 2trains_V1.factoryio et 2trains_SA_AR_RC_V2_sol_Export_complet.xml
      - Conclusion, quels sont les inconvénients de cette loi de commande ?

3. Gestion de feux et d’aiguillage : travail personnel

- Partie 2
  - Répertoire BooG\Exemple\2trains_G7_SA
    - Ouvrir et analyser le fichier BESS : 2trains_SA_AR_RC_V3.txt
      - Quelles sont les différences (variables, spécifications, hypothèses, optimisation) avec le fichier 2trains_SA_AR_RC_V2.txt
    - Ajouter le fichier 2trains_G7_AR_RC.xml et générer avec BooG le fichier 2trains_SA_AR_RC_V3_sol _Export_complet.txt
    - Ouvrir le fichier 2trains_V1.factoryio avec FACTORY I/O
    - Simuler et Tester avec GReSTIC, en sélectionnant les fichiers 2trains_V1.factoryio et 2trains_SA_AR_RC_V3_sol_Export_complet.xml
      - Conclusion, que pensez-vous cette loi de commande ?

    - Proposer et tester d’autres contrôleurs logiques pour ce système

Conclusion

Conclusion

- Introduction à la synthèse algébrique
  - Méthode formelle
  - Outils logiciels
- Boîte à outils de l'automaticien des SED
- Applications au contrôle/commande, mais pas seulement…
- Nécessité d'une étape de V&V
- A vous de tester et de jouer !

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 : Grégory Faraut, Dimitri Renard, Bernard Riera, Jean-Marc Roussel

Intervenants : Grégory Faraut, Bernard Riera

Synthèse algébrique : choix de la solution

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

P1=0 & P2=0

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

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

P1=0 & P2=1

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

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

P1=/i1 & P2=x

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

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

P1=/i2 & P2=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

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

P1=i2 & P2=x

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

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

P1=/i2 & P2=0

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

6 solutions possibles

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

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 : choix de la solution

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

P1=i2 & P2=0

Max $(\overline{O_1} \cdot \overline{i_2} + O_1 \cdot i_2)$ & Min O_2

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

P1=/i2 & P2=0

Max $(\overline{O_1} \cdot i_2 + O_1 \cdot \overline{i_2})$ & Min O_2

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

6 solutions possibles

$$\begin{cases} O_1 = i_1 + P_1 \\ O_2 = \overline{i_1} \cdot i_2 \cdot \overline{P_1} \cdot P_2 \end{cases}$$

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