Synthèse algébrique de contrôleurs logiques

Comment passer d'un cahier des charges informel donné en langage naturel à un modèle formel de commande ?

Jean-Marc Roussel

jean-marc.roussel@ens-paris-saclay.fr

LURPA, ENS Paris-Saclay

Synthèse algébrique de contrôleurs logiques

Origine de cette méthode

- Résultat d'une trajectoire de recherche

Introduction

- Définition et modèle mathématique
- Exemple introductif

Cadre mathématique

- Structure d'algèbre de Boole
- Principales caractéristiques
- Résolution d'équations dans une algèbre de Boole

Synthèse d'un système logique

- Méthode de synthèse proposée
- Formalisation du cahier des charges
- Exemple de résolution

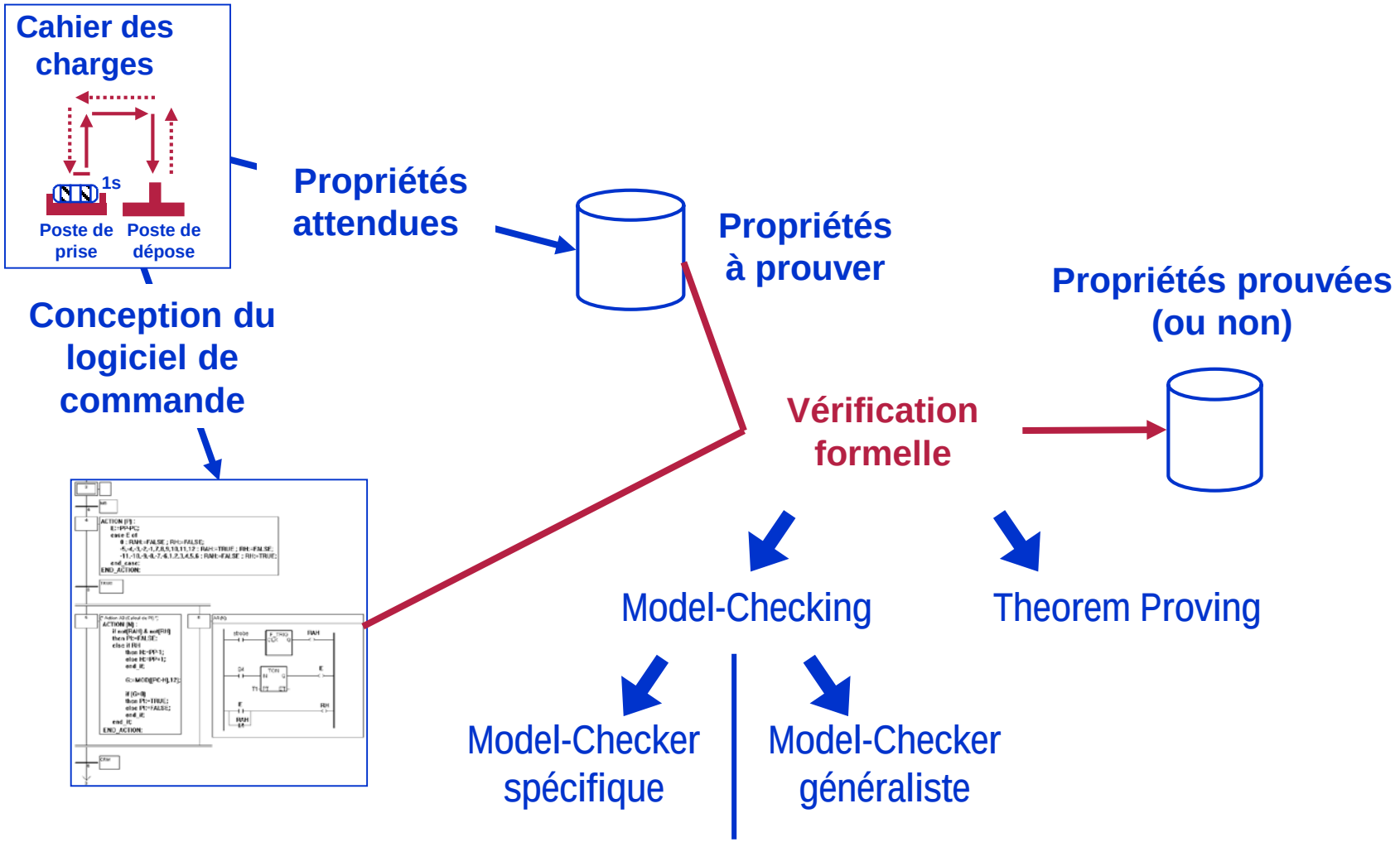
Traitement d'une étude de cas

- Commande d'un portail automatique

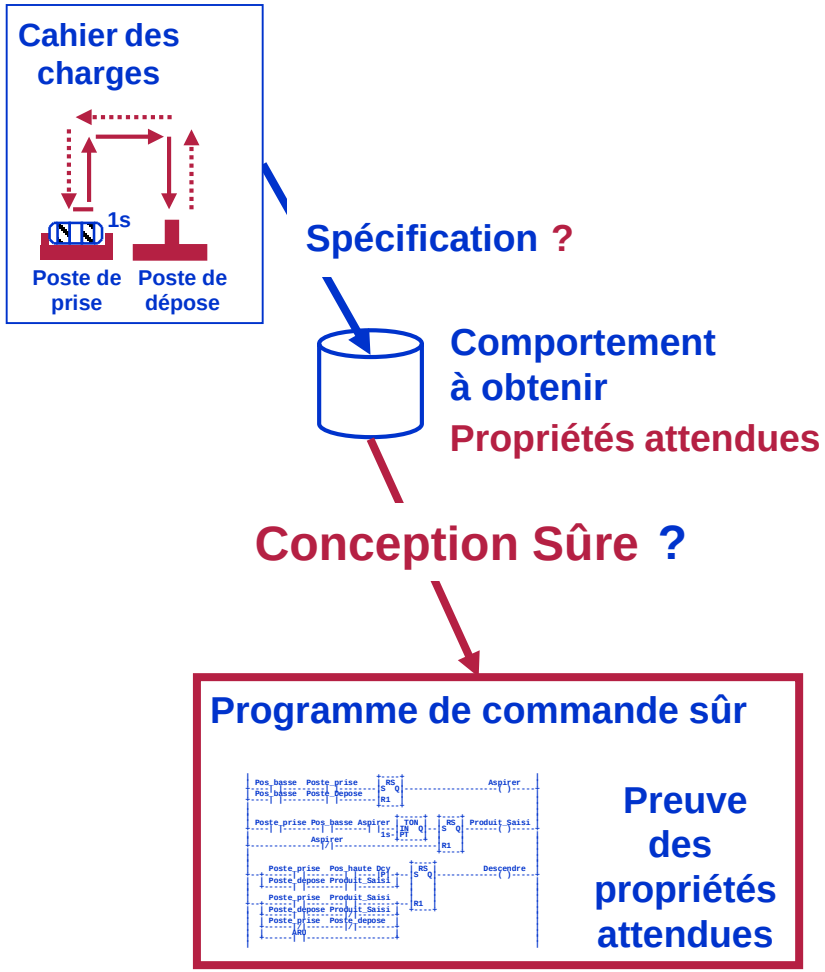
Logiciel support : BESS

- Interface opérateur
- Structure de données :
  - Shared Reduced Ordered Binary Decision Diagrams

Vérification formelle d'un modèle de commande



Synthèse d'un programme de commande sûr



Comment spécifier le comportement du système de contrôle-commande ?

- Par la donnée des différentes propriétés qu'il doit respecter :
  - propriétés de sûreté (ce qu'il ne doit pas faire)
  - propriétés décrivant le fonctionnement (ce qu'il doit faire)

Comment concevoir un programme qui garantisse le respect de la spécification ?

- Par extraction du programme de la spécification
  - en détectant ses incohérences,
  - en détectant ses incomplétudes,
  - …

Synthèse algébrique de contrôleurs logiques

Origine de cette méthode

- Résultat d'une trajectoire de recherche

Introduction

- Définition et modèle mathématique
- Exemple introductif

Cadre mathématique

- Structure d'algèbre de Boole
- Principales caractéristiques
- Résolution d'équations dans une algèbre de Boole

Synthèse d'un système logique

- Méthode de synthèse proposée
- Formalisation du cahier des charges
- Exemple de résolution

Traitement d'une étude de cas

- Commande d'un portail automatique

Logiciel support : BESS

- Interface opérateur
- Structure de données :
  - Shared Reduced Ordered Binary Decision Diagrams

Système logique combinatoire

Système logique

- Système dont les entrées et sorties ont une valeur logique (0 ou 1).

e_i : Entrée

s_i : Sortie

Système logique **combinatoire** (SLC)

- 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.**

$$\begin{cases} s_1[k] = F_1(e_1[k], \cdots, e_m[k]) \\ \vdots \\ s_p[k] = F_p(e_1[k], \cdots, e_m[k]) \end{cases}$$

Caractéristiques

- F_i : Fonctions combinatoires
- m entrées, p sorties :

$\left(2^{2^m}\right)^p$ **systèmes différents**

Système logique séquentiel

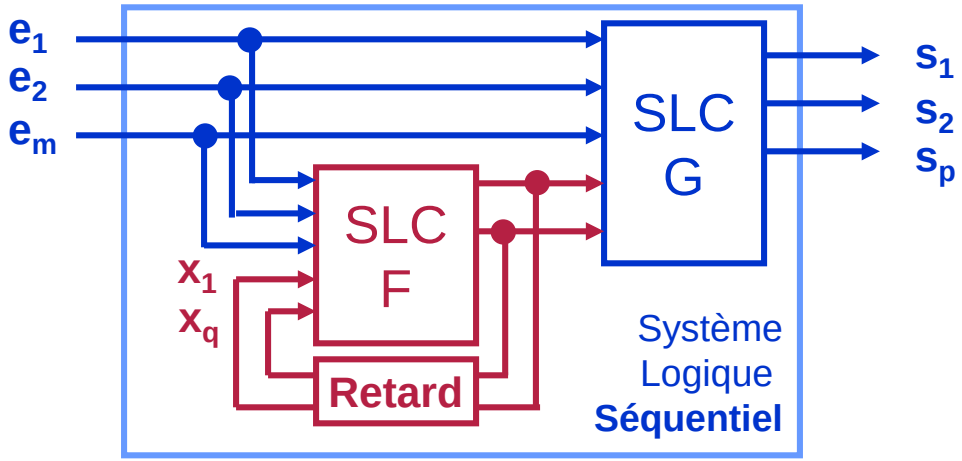
Système logique séquentiel (SLS)

- Système pour lequel la valeur de chaque sortie à tout instant ti est déterminée à partir
  - de la valeur des entrées à cet instant ti,
  - **et** de la **valeur antérieure des entrées**.

État interne d'un système

- Ensemble des informations prenant en compte la mémorisation des évolutions passées et permettant de déterminer les évolutions futures.
- Représentation
  - par automates à états
  - par équations récurrentes

Représentation d'un Système Logique Séquentiel par équations récurrentes



e_i : Entrée

s_i : Sortie

x_i : Variable interne

$$\begin{cases} \mathbf{x_1}[k] = \mathbf{F_1}(\mathbf{e_1}[k], \cdots, \mathbf{e_m}[k], \mathbf{x_1}[k-1], \cdots, \mathbf{x_q}[k-1]) \\ \vdots \\ \mathbf{x_q}[k] = \mathbf{F_q}(\mathbf{e_1}[k], \cdots, \mathbf{e_m}[k], \mathbf{x_1}[k-1], \cdots, \mathbf{x_q}[k-1]) \\ \mathbf{s_1}[k] = \mathbf{G_1}(\mathbf{e_1}[k], \cdots, \mathbf{e_m}[k], \mathbf{x_1}[k], \cdots, \mathbf{x_q}[k]) \\ \vdots \\ \mathbf{s_p}[k] = \mathbf{G_p}(\mathbf{e_1}[k], \cdots, \mathbf{e_m}[k], \mathbf{x_1}[k], \cdots, \mathbf{x_q}[k]) \end{cases}$$

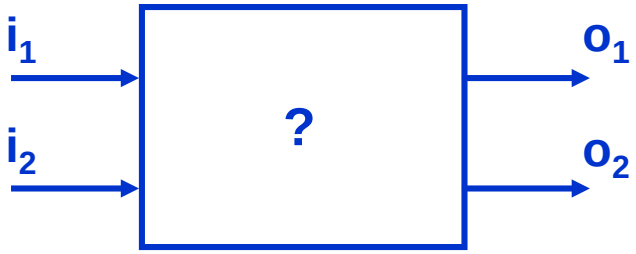
Caractéristiques

- F_i , G_i : Fonctions combinatoires
- X_i définie par récurrence
  - Valeurs initiales
- q variables internes :
  au plus 2^q états différents

Synthèse d'un système logique combinatoire

Comportement attendu

- **R1** : Il suffit d'avoir 'i_1' pour avoir 'o_1'.
- **R2** : Il faut avoir 'i_2' pour avoir 'o_2'.
- **R3** : 'o_1' et 'o_2' ne sont jamais simultanément Vrai.



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

Approches classiques

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

Approche proposée

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

$$\begin{cases} I_1 \leq O_1 \\ O_2 \leq I_2 \\ O_1 \cdot O_2 = 0 \end{cases} \Rightarrow \begin{cases} O_1 = I_1 + P_1 \\ O_2 = \overline{I_1} \cdot I_2 \cdot \overline{P_1} \cdot P_2 \end{cases}$$

Synthèse algébrique de contrôleurs logiques

Origine de cette méthode

- Résultat d'une trajectoire de recherche

Introduction

- Définition et modèle mathématique
- Exemple introductif

Cadre mathématique

- Structure d'algèbre de Boole
- Principales caractéristiques
- Résolution d'équations dans une algèbre de Boole

Synthèse d'un système logique

- Méthode de synthèse proposée
- Formalisation du cahier des charges
- Exemple de résolution

Traitement d'une étude de cas

- Commande d'un portail automatique

Logiciel support : BESS

- Interface opérateur
- Structure de données :
  - Shared Reduced Ordered Binary Decision Diagrams

Structure d'algèbre de Boole $\left(S,+,\cdot,\overline{\ },0,1\right)$

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
  - $\overline{\ }$: 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

$$0 \neq 1$$

Exemples 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 U

- Ensemble : $\mathrm{P}(\mathrm{U}) = \{ \emptyset , \cdots , \mathrm{U} \}$
- Éléments particuliers : $\emptyset , \mathrm{U}$
- Opérations : $\cup, \cap, \mathrm{C}_{\mathrm{U}}$

Algèbre des fonctions booléennes de dimension n

Algèbre de Boole des fonctions Booléennes

Eléments de l'algèbre : Fonctions booléennes de dimension n

$$F_n(B) = \{f : B^n \to B\} \qquad B = \{0, 1\} \qquad |F_n(B)| = 2^{2^n}$$

- 2 Fonctions constantes

0 : $B^n \to B$, $(b_1, \cdots, b_n) \mapsto 0$ **1** : $B^n \to B$, $(b_1, \cdots, b_n) \mapsto 1$

- n Fonctions « Projection »

$$f^i_{\text{Proj}} : B^n \to B, \quad (b_1, \cdots, b_n) \mapsto b_i$$

Opérations

OU : $F_n(B) \times F_n(B) \to F_n(B)$, $(f, g) \mapsto f + g$

ET : $F_n(B) \times F_n(B) \to F_n(B)$, $(f, g) \mapsto f \cdot g$

NON : $F_n(B) \to F_n(B)$, $f \mapsto \bar{f}$

$$\forall (b_1, \cdots, b_n) \in B^n,$$
$$(f + g)(b_1, \cdots, b_n) = f(b_1, \cdots, b_n) \vee g(b_1, \cdots, b_n)$$
$$(f \cdot g)(b_1, \cdots, b_n) = f(b_1, \cdots, b_n) \wedge g(b_1, \cdots, b_n)$$
$$\bar{f}(b_1, \cdots, b_n) = \neg f(b_1, \cdots, b_n)$$

Expressions booléennes dans $\left(S,+,\cdot,\overline{\ },0,1\right)$

Définition

- Expression : Composition des éléments de S par : $+,\cdot,\overline{\ }$

Caractéristiques

- À toute expression correspond un unique élément de S.
- 2 expressions sont équivalentes si elles correspondent au même élément de S.

Toute expression a une forme canonique

$$\forall\, \alpha_1 \in S \setminus \{0,1\},\quad E(\alpha_1,\cdots,\alpha_k) = E_0(\alpha_2,\cdots,\alpha_k)\cdot\overline{\alpha_1} + E_1(\alpha_2,\cdots,\alpha_k)\cdot\alpha_1$$

$$\begin{cases} E_0(\alpha_2,\cdots,\alpha_k) = E(\alpha_1,\cdots,\alpha_k)\Big|_{\alpha_1\leftarrow 0} = E(0,\alpha_2,\cdots,\alpha_k) \\ E_1(\alpha_2,\cdots,\alpha_k) = E(\alpha_1,\cdots,\alpha_k)\Big|_{\alpha_1\leftarrow 1} = E(1,\alpha_2,\cdots,\alpha_k) \end{cases}$$

Relation d'ordre partiel dans une algèbre de Boole

Relation **Inclusion**

$$x \leq y \quad \Leftrightarrow \quad x \cdot y = x$$

Ordre partiel : Relation réflexive, antisymétrique et transitive

- Réflexivité : $x \leq x$
- Antisymétrie : Si $x \leq y$ et $y \leq x$ alors $x = y$
- Transitivité : Si $x \leq y$ et $y \leq z$ alors $x \leq z$

Cas des ensembles

- $\mathbf{a} \subset \mathbf{b} \quad \Leftrightarrow \quad \mathbf{a} \cap \mathbf{b} = \mathbf{a}$

Cas des fonctions booléennes

- $\mathbf{f} \leq \mathbf{g} \quad \Leftrightarrow \quad \mathbf{f} \cdot \mathbf{g} = \mathbf{f}$

$\forall (b_1, \cdots, b_n) \in B^n$,
$f(b_1, \cdots, b_n) \wedge g(b_1, \cdots, b_n) = f(b_1, \cdots, b_n)$

b_1	b_2	$f(b_1,b_2)$	$g(b_1,b_2)$
0	0	0	1
0	1	0	0
1	0	1	1
1	1	1	1

Caractéristiques des relations dans $\left(S,+,\cdot,\overline{\ },0,1\right)$

Relations entre éléments de S ou des expressions booléennes de S

- $x \leq y$
- $E_1(\alpha_1,\cdots,\alpha_k) \leq E_2(\alpha_1,\cdots,\alpha_k)$

Une même relation admet des écritures équivalentes :

- $x = y \quad \Leftrightarrow \quad (x \cdot \bar{y}) + (\bar{x} \cdot y) = 0$
- $x \leq y \quad \Leftrightarrow \quad (x \cdot \bar{y}) = 0$
- $\begin{cases} x = 0 \\ y = 0 \end{cases} \quad \Leftrightarrow \quad x + y = 0$

Les relations peuvent être composées entre-elles :

$$\begin{cases} w \leq x \\ y = z \end{cases} \quad \Leftrightarrow \quad \begin{cases} w \cdot \bar{x} = 0 \\ (y \cdot \bar{z}) + (\bar{y} \cdot z) = 0 \end{cases} \quad \Leftrightarrow \quad (w \cdot \bar{x}) + (y \cdot \bar{z}) + (\bar{y} \cdot z) = 0$$

Tout ensemble de relations peut être exprimé à l'aide d'une égalité à 0.

Résolution d'une équation au sein de l'algèbre de Boole $\left(F_n(B),+,\cdot,\overline{\ },0,1\right)$

Notations

- $(x_1,\cdots,x_k)$: k éléments de $F_n(B)$ considérés comme inconnues
- X_k : Vecteur $(x_1,\cdots,x_k)$
- $Proj_n$: Vecteur $(f^1_{Proj},\cdots,f^n_{Proj})$ des n fonctions « Projection » f^i_{Proj}
- Équation à résoudre :

$$Eq(X_k, Proj_n) = 0$$

Objectif de la résolution

$$\text{Trouver} \begin{cases} x_1 = E_1(Proj_n) \\ \vdots \\ x_i = E_i(Proj_n) \\ \vdots \\ x_k = E_k(Proj_n) \end{cases} \quad \text{tel que} \quad Eq(X_k, Proj_n) = 0$$

Forme canonique d'une équation de $\left(F_n(B),+,\cdot,\overline{\ },0,1\right)$

Toute équation admet une forme canonique

- 1 inconnue : $\mathbf{a \cdot \bar{x} + b \cdot x = 0}$
- 2 inconnues : $\mathbf{a \cdot \overline{x_1} \cdot \overline{x_2} + b \cdot \overline{x_1} \cdot x_2 + c \cdot x_1 \cdot \overline{x_2} + d \cdot x_1 \cdot x_2 = 0}$

Notations $a \in \{0,1\}$

- x_i^a : $x_i^0 = \bar{x_i}$ $x_i^1 = x_i$
- A_k : Vecteur $(a_1, \cdots, a_k) \in \{0,1\}^k$
- $X_k^{A_k}$: $X_k^{A_k} = \prod_{i=1}^{i=k} x_i^{a_i} = x_1^{a_1} \cdot \ldots \cdot x_k^{a_k}$

Forme canonique d'une équation

$$\mathbf{Eq}(X_k, Proj_n) = \sum_{A_k \in \{0,1\}^k} \mathbf{Eq}(A_k, Proj_n) \cdot X_k^{A_k} = 0$$

- $\mathrm{Eq}(A_k, Proj_n)$: Un des 2^k discriminants de $\mathrm{Eq}(X_k, Proj_n)$ suivant X_k

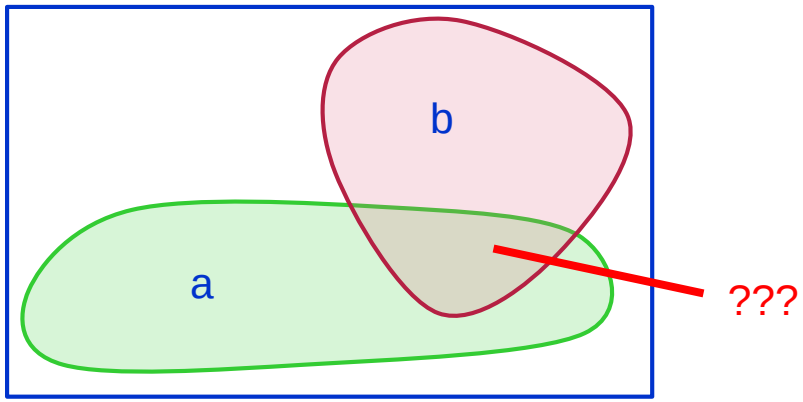
Principe de résolution

Équation à résoudre

- a, b et x sont des ensembles

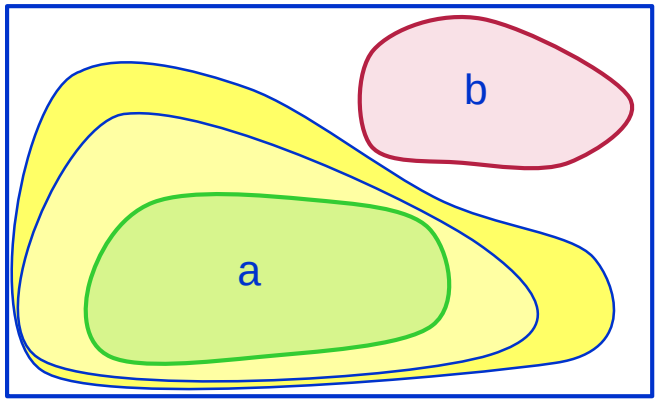
$$(\mathbf{a} \cap \bar{\mathbf{x}}) \cup (\mathbf{b} \cap \mathbf{x}) = \emptyset$$

$$\Leftrightarrow \begin{cases} (\mathbf{a} \cap \bar{\mathbf{x}}) = \emptyset \\ (\mathbf{b} \cap \mathbf{x}) = \emptyset \end{cases} \Leftrightarrow \begin{cases} \mathbf{a} \subset \mathbf{x} \\ (\mathbf{b} \cap \mathbf{x}) = \emptyset \end{cases}$$



Existence de solutions

$$\mathbf{a} \cap \mathbf{b} = \emptyset$$



Forme paramétrique de la solution

$$\mathbf{x} = \mathbf{a} \cup (\bar{\mathbf{b}} \cap \mathbf{p}) \text{ avec } \mathbf{p} \in \mathrm{P}(\mathrm{U})$$

Résolution d'une équation à une inconnue au sein de l'algèbre de Boole $\left(F_n(B),+,\cdot,\overline{\ },0,1\right)$

Équation à résoudre

$$a \cdot \bar{x} + b \cdot x = 0$$

Existence de solutions

$$a \cdot b = 0$$

Forme paramétrique de la solution

$$x = a + \bar{b} \cdot p \quad \text{avec} \quad p \in F_n(B)$$

- Autres formulations possibles

$$x = a + \bar{b} \cdot p = \bar{b} \cdot (a + p) = a \cdot \bar{p} + \bar{b} \cdot p$$

Résolution d'une équation à k inconnues au sein de l'algèbre de Boole $\left(F_n(B),+,\cdot,\overline{\ },0,1\right)$

$$\mathbf{Eq_0}(X_k, Proj_n) = \sum_{A_k \in \{0,1\}^k} \mathbf{Eq_0}(A_k, Proj_n) \cdot X_k^{A_k} = \mathbf{0}$$

Existence de solutions

$$\prod_{A_k \in \{0,1\}^k} \mathbf{Eq_0}(A_k, Proj_n) = \mathbf{0}$$

Forme paramétrique de la solution : k-uplet $(S(x_1), \cdots, S(x_k))$

$$S(x_i) = \left(\prod_{A_{k-i} \in \{0,1\}^{k-i}} \mathbf{Eq}_{i-1}(\mathbf{0}, A_{k-i}, Proj_n) \right) + \mathbf{p}_i \cdot \overline{\left(\prod_{A_{k-i} \in \{0,1\}^{k-i}} \mathbf{Eq}_{i-1}(\mathbf{1}, A_{k-i}, Proj_n) \right)}$$

- $\mathbf{p}_i \in \mathbf{F}_n(\mathbf{B})$
- $\mathbf{Eq}_i(x_{i+1}, \cdots, x_k, Proj_n) = \mathbf{Eq}_{i-1}(x_i, x_{i+1}, \cdots, x_k, Proj_n)|_{x_i \leftarrow S(x_i)}$

Synthèse algébrique de contrôleurs logiques

Origine de cette méthode

- Résultat d'une trajectoire de recherche

Introduction

- Définition et modèle mathématique
- Exemple introductif

Cadre mathématique

- Structure d'algèbre de Boole
- Principales caractéristiques
- Résolution d'équations dans une algèbre de Boole

Synthèse d'un système logique

- Méthode de synthèse proposée
- Formalisation du cahier des charges
- Exemple de résolution

Traitement d'une étude de cas

- Commande d'un portail automatique

Logiciel support : BESS

- Interface opérateur
- Structure de données :
  - Shared Reduced Ordered Binary Decision Diagrams

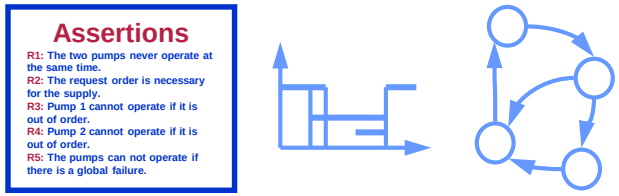
Synthèse algébrique de contrôleurs logiques

Résultat attendu

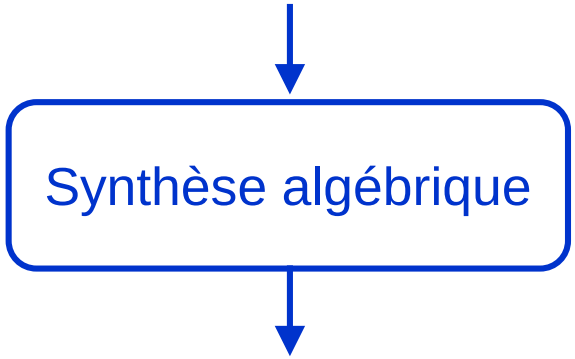
- Lois de commande à implanter

Données d'entrée

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



Fragments de spécification



Lois de commande à implanter

Pourquoi une synthèse algébrique ?

- **Synthèse automatique :**
  - Pour supprimer des erreurs humaines
- **Synthèse algébrique :**
  - Pour limiter l'explosion combinatoire

Étapes de la méthode

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.

1) Formalisation du CdC

- Formalisation de chaque fragment donné de manière informelle

2) Vérification de la cohérence

- Par calcul symbolique

3) Calcul des solutions

- En résolvant une équation entre des fonctions booléennes

Résolution analytique

4) Choix de la solution

- En fixant une valeur spécifique à chaque paramètre

1 - Formalisation

Ensemble de relations entre des fonctions booléennes

2 – Vérification de la cohérence

Oui

Équation solvable entre des fonctions booléennes

Non

Condition d'incohérence

3 – Calcul des solutions possibles

Forme paramétrique des solutions

4 – Choix de la solution

Activité manuelle Activité automatique

Étapes de la formalisation du cahier des charges

1. Choix de l'algèbre de Boole support des calculs : $\left(F_n(B),+,\cdot,\overline{\ },0,1\right)$

- Choix des n fonctions « Projection » : f^i_{Proj}
  - Une fonction « Projection » par variable booléenne libre du problème

2. Référencement des autres éléments du discours

- Fonctions booléennes de $\left(F_n(B),+,\cdot,\overline{\ },0,1\right)$
  - Une fonction par éléments à synthétiser (sorties, variables internes, réceptivités, ...)
  - Une fonction pour chaque mot-clé du cahier des charges
    - Nom de fonction
    - Règle d'obtention

3. Formalisation du cahier des charges

- Représentation de chaque assertion du cahier des charges par une ou plusieurs relations entre des éléments de $F_n(B)$

Synthèse d'un système combinatoire

(m entrées, p sorties)

Éléments à déterminer

- p fonctions booléennes $\mathbf{F_i}$

Algèbre : $\left(F_m(B),+,\cdot,\overline{\ },0,1\right)$

- Fonctions « Projection » : f^i_{Proj}
  - Une pour chaque entrée : $\mathbf{U_i}$

$$B^m \rightarrow B$$
$$(u_1[k], \cdots, u_m[k]) \mapsto u_i[k]$$

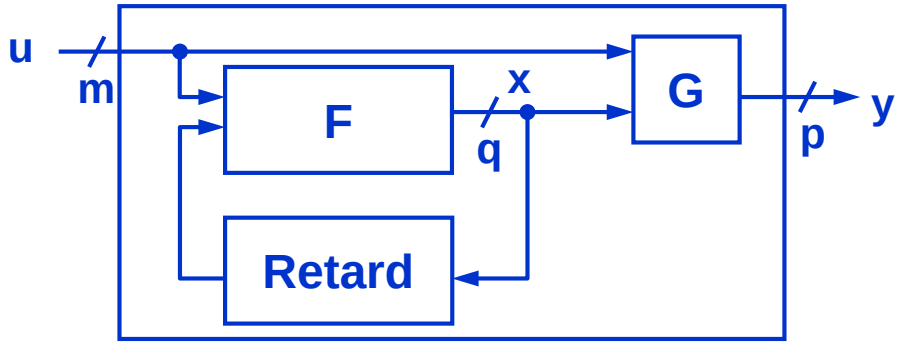
- Fonctions « Inconnue » :
  - Une pour chaque sortie : $\mathbf{Y_i}$

$$B^m \rightarrow B$$
$$(u_1[k], \cdots, u_m[k]) \mapsto y_i[k]$$

Représentations récursives d'un système séquentiel

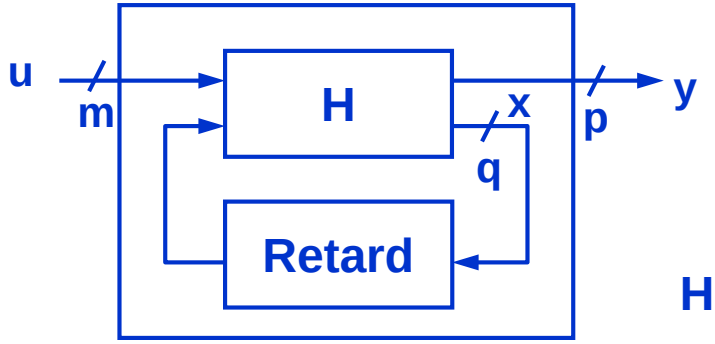
(m entrées, p sorties, q variables internes)

Représentation traditionnelle



$$\begin{cases} \mathbf{x}_i[k] = \mathbf{F}_i(\mathbf{u}[k], \mathbf{x}[k-1]) \\ \mathbf{y}_j[k] = \mathbf{G}_j(\mathbf{u}[k], \mathbf{x}[k]) \end{cases}$$

Représentation retenue



$$\begin{cases} \mathbf{x}_i[k] = \mathbf{H}_i(\mathbf{u}[k], \mathbf{x}[k-1]) \\ \mathbf{y}_j[k] = \mathbf{H}_{q+j}(\mathbf{u}[k], \mathbf{x}[k-1]) \end{cases}$$

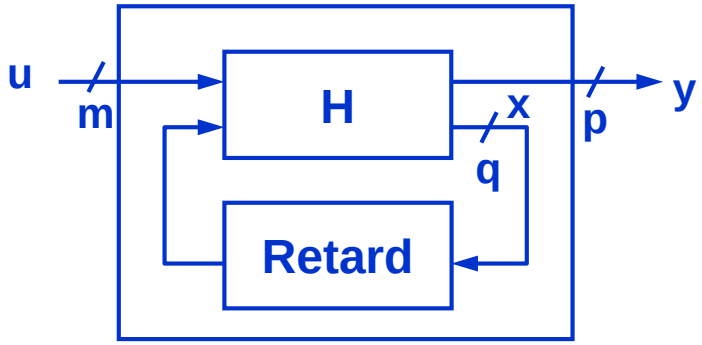
$$\mathbf{H}_{q+j}(\mathbf{u}[k], \mathbf{x}[k-1]) = \mathbf{G}_j\big(\mathbf{u}[k], \mathbf{F}_i(\mathbf{u}[k], \mathbf{x}[k-1])\big)$$

Synthèse d'un système séquentiel (1)

(m entrées, p sorties, q variables internes)

Éléments à déterminer

- $p+q$ fonctions booléennes $\mathbf{H}_i$



Algèbre : $\left(F_{m+q}(B),+,\cdot,\overline{\ },0,1\right)$

- Fonctions « Projection » : f^{i}_{Proj}
  - Une pour chaque entrée : $\mathbf{U}_i$

$$B^{m+q} \rightarrow B$$
$$\left(u_1[k], \cdots, u_m[k], x_1[k-1], \cdots, x_q[k-1]\right) \mapsto u_i[k]$$

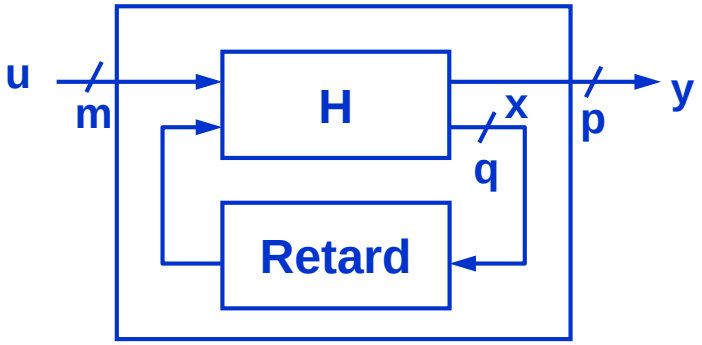
  - Une pour chaque variable interne : $\mathbf{Xp}_i$
    - $\mathbf{Xp}_i$: Fonction booléenne définissant la valeur précédente $x_i[k-1]$

$$B^{m+q} \rightarrow B$$
$$\left(u_1[k], \cdots, u_m[k], x_1[k-1], \cdots, x_q[k-1]\right) \mapsto x_i[k-1]$$

Synthèse d'un système séquentiel (2)

(m entrées, p sorties, q variables internes)

Algèbre : $\left(F_{m+q}(B),+,\cdot,\overline{\ },0,1\right)$



- Fonctions « Projection » : f^{i}_{Proj}
  - Une pour chaque entrée : $\mathbf{U}_i$
  - Une pour chaque variable interne : $\mathbf{Xp}_i$
- Fonctions « Inconnue » :
  - Une pour chaque variable interne : $\mathbf{X}_i$
    - $\mathbf{X}_i$: Fonction booléenne définissant la valeur courante $x_i[k]$

$$B^{m+q} \rightarrow B$$

$$\left(u_1[k], \cdots, u_m[k], x_1[k-1], \cdots, x_q[k-1]\right) \mapsto x_i[k]$$

  - Une pour chaque sortie : $\mathbf{Y}_i$

$$B^{m+q} \rightarrow B$$

$$\left(u_1[k], \cdots, u_m[k], x_1[k-1], \cdots, x_q[k-1]\right) \mapsto y_i[k]$$

À chaque variable interne sont associées 2 fonctions booléennes : $\mathbf{Xp}_i$, $\mathbf{X}_i$

Apport de la relation Inclusion pour la formalisation du cahier des charges

Relation **Inclusion**

$$\mathbf{f \leq g \quad \Leftrightarrow \quad f \cdot g = f}$$

$\forall (b_1, \cdots, b_n) \in B^n$,
$f(b_1, \cdots, b_n) \wedge g(b_1, \cdots, b_n) = f(b_1, \cdots, b_n)$

Ou

$\forall (b_1, \cdots, b_n) \in B^n$,
Si $f(b_1, \cdots, b_n) = 1_b$
Alors $g(b_1, \cdots, b_n) = 1_b$

b_1	b_2	$f(b_1,b_2)$	$g(b_1,b_2)$
0	0	0	1
0	1	0	0
1	0	1	1
1	1	1	1

Intérêt pour notre méthode

- Permet d'exprimer différentes assertions
  - Si **f** est vrai, alors **g** est vrai.
  - Il **suffit** d'avoir **f** pour avoir **g**.
  - Il **faut** avoir **g** pour avoir **f**.

Exemples de formalisation (1)
Contraintes intemporelles

Propositions simples

- Il **suffit** d'avoir 'a' pour avoir 'b'.

$$\mathbf{A} \leq \mathbf{B}$$

- Il **faut** avoir 'a' pour avoir 'b'.

$$\mathbf{B} \leq \mathbf{A}$$

- Il est **nécessaire** et **suffisant** d'avoir 'a' pour avoir 'b'.

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

- 'a' et 'b' ne peuvent pas avoir lieu simultanément.

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

Propositions composées

- Lorsque 'c' est vrai, il **suffit** d'avoir 'a' pour avoir 'b'.

$$(\mathbf{C} \cdot \mathbf{A}) \leq \mathbf{B}$$

- Lorsque 'c' est vrai, il **faut** avoir 'a' pour avoir 'b'.

$$(\mathbf{C} \cdot \mathbf{B}) \leq \mathbf{A}$$

- Lorsque 'c' est vrai, il est **nécessaire** et **suffisant** d'avoir 'a' pour avoir 'b'.

$$\begin{cases} (\mathbf{C} \cdot \mathbf{A}) \leq \mathbf{B} \\ (\mathbf{C} \cdot \mathbf{B}) \leq \mathbf{A} \end{cases} \Leftrightarrow \mathbf{C} \leq \left(\mathbf{A} \cdot \mathbf{B} + \overline{\mathbf{A}} \cdot \overline{\mathbf{B}}\right)$$

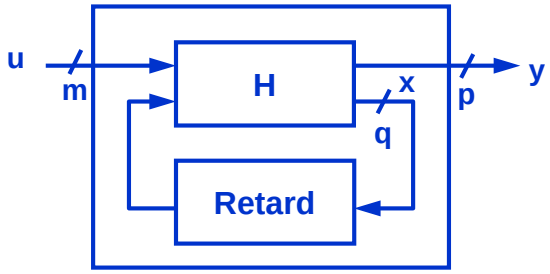
- Lorsque 'c' est vrai, 'a' et 'b' ne peuvent pas avoir lieu simultanément.

$$\mathbf{C} \cdot \mathbf{A} \cdot \mathbf{B} = \mathbf{0}$$

Exemples de formalisation (2)
Contraintes de changement d'état

Éléments du discours

- État : $\mathbf{X_i}$
- Mise à **1** d'un état : $\mathbf{X_i \cdot \overline{Xp_i}}$
- Mise à **0** d'un état : $\mathbf{\overline{X_i} \cdot Xp_i}$



Contraintes liées à l'état

- Il **suffit** d'avoir 'a' pour avoir 'x_i'.

$$\mathbf{A \leq X_i}$$

- Il **faut** avoir 'a' pour avoir 'x_i'.

$$\mathbf{X_i \leq A}$$

Contraintes liées à la mise à 1

- Il **faut** avoir 'a' pour mettre à 1 'x_i'.

$$\mathbf{X_i \cdot \overline{Xp_i} \leq A}$$

- Il **suffit** d'avoir 'a' pour mettre à 1 'x_i'.

$$\mathbf{\overline{Xp_i} \cdot A \leq X_i \cdot \overline{Xp_i}}$$

Contraintes liées à la mise à 0

- Il **faut** avoir 'a' pour mettre à 0 'x_i'.

$$\mathbf{\overline{X_i} \cdot Xp_i \leq A}$$

- Il **suffit** d'avoir 'a' pour mettre à 0 'x_i'.

$$\mathbf{Xp_i \cdot A \leq \overline{X_i} \cdot Xp_i}$$

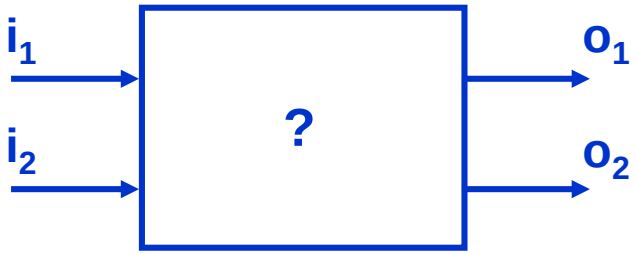
Modélisation du cahier des charges

Objet du discours

- Éléments de $F_2(B)$: $\{I_1, I_2, O_1, O_2\}$

Entrées du SLC : $\{I_1, I_2\}$
Fonctions Projection de $F_2(B)$

Sorties du SLC : $\{O_1, O_2\}$
Éléments de $F_2(B)$ à déterminer



Comportement attendu

- **R1** : Il suffit d'avoir 'i_1' pour avoir 'o_1'.
- **R2** : Il faut avoir 'i_2' pour avoir 'o_2'.
- **R3** : 'o_1' et 'o_2' ne sont jamais simultanément Vrai.

$$\begin{cases} \textbf{R1 :} & I_1 \leq O_1 \\ \textbf{R2 :} & O_2 \leq I_2 \\ \textbf{R3 :} & O_1 \cdot O_2 = 0 \end{cases}$$

Mise en équation

Reformulation des différentes relations

- Objectif : Une unique égalité à 0

$$\begin{cases} I_1 \leq O_1 \\ O_2 \leq I_2 \\ O_1 \cdot O_2 = 0 \end{cases} \Leftrightarrow \begin{cases} I_1 \cdot \overline{O_1} = 0 \\ O_2 \cdot \overline{I_2} = 0 \\ O_1 \cdot O_2 = 0 \end{cases} \Leftrightarrow I_1 \cdot \overline{O_1} + \overline{I_2} \cdot O_2 + O_1 \cdot O_2 = 0$$

Identification des inconnues : $\{O_1, O_2\}$

- Objectif : Expression uniquement des inconnues et des fonctions « projection »

$$Eq_0(X_1, X_2, I_1, I_2) = I_1 \cdot \overline{X_1} + \overline{I_2} \cdot X_2 + X_1 \cdot X_2 = 0$$

Mise sous canonique

$$I_1 \cdot \overline{X_1} \cdot (\overline{X_2} + X_2) + \overline{I_2} \cdot (\overline{X_1} + X_1) \cdot X_2 + X_1 \cdot X_2 = 0$$

$$I_1 \cdot \overline{X_1} \cdot \overline{X_2} + (I_1 + \overline{I_2}) \cdot \overline{X_1} \cdot X_2 + 0 \cdot X_1 \cdot \overline{X_2} + 1 \cdot X_1 \cdot X_2 = 0$$

Résolution de l'équation (1)

Équation à résoudre

$$Eq_0(X_1, X_2, I_1, I_2) = I_1 \cdot \overline{X_1} \cdot \overline{X_2} + (I_1 + \overline{I_2}) \cdot \overline{X_1} \cdot X_2 + 0 \cdot X_1 \cdot \overline{X_2} + 1 \cdot X_1 \cdot X_2 = 0$$

$$Eq_0(0,0, I_1, I_2) = I_1 \qquad Eq_0(1,0, I_1, I_2) = 0$$

$$Eq_0(0,1, I_1, I_2) = (I_1 + \overline{I_2}) \qquad Eq_0(1,1, I_1, I_2) = 1$$

Condition d'existence de solutions

$$\prod_{A_k \in \{0,1\}^k} Eq_0(A_k, Proj_n) = 0$$

$$\prod_{A_2 \in \{0,1\}^2} Eq_0(A_2, Proj_n) = I_1 \cdot (I_1 + \overline{I_2}) \cdot 0 \cdot 1 = 0$$

Solution pour X_1

$$S(x_i) = \left(\prod_{A_{k-i} \in \{0,1\}^{k-i}} Eq_{i-1}(0, A_{k-i}, Proj_n) \right) + p_i \cdot \overline{\left(\prod_{A_{k-i} \in \{0,1\}^{k-i}} Eq_{i-1}(1, A_{k-i}, Proj_n) \right)}$$

$$S(X_1) = \left(I_1 \cdot (I_1 + \overline{I_2}) \right) + P_1 \cdot \overline{(0 \cdot 1)} = I_1 + P_1 \quad \text{avec} \quad P_1 \in F_2(B)$$

Résolution de l'équation (2)

Définition de $Eq_1(X_2, I_1, I_2)$

$$Eq_i(x_{i+1}, \cdots, x_k, Proj_n) = Eq_{i-1}(x_i, x_{i+1}, \cdots, x_k, Proj_n)\Big|_{x_i \leftarrow S(x_i)}$$

$$Eq_1(X_2, I_1, I_2) = Eq_0(X_1, X_2, I_1, I_2)\Big|_{X_1 \leftarrow I_1 + P_1} = \left(I_1 \cdot \overline{X_1} + \overline{I_2} \cdot X_2 + X_1 \cdot X_2\right)\Big|_{X_1 \leftarrow I_1 + P_1}$$

$$Eq_1(X_2, I_1, I_2) = I_1 \cdot \overline{(I_1 + P_1)} + \overline{I_2} \cdot X_2 + (I_1 + P_1) \cdot X_2 = \left(I_1 + \overline{I_2} + P_1\right) \cdot X_2 = 0$$

Solution pour X_2

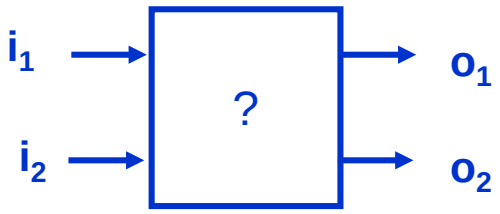
$$S(x_i) = \left(\prod_{A_{k-i} \in \{0,1\}^{k-i}} Eq_{i-1}(0, A_{k-i}, Proj_n)\right) + p_i \cdot \overline{\left(\prod_{A_{k-i} \in \{0,1\}^{k-i}} Eq_{i-1}(1, A_{k-i}, Proj_n)\right)}$$

$$S(X_2) = 0 + P_2 \cdot \overline{\left(I_1 + \overline{I_2} + P_1\right)} = \overline{I_1} \cdot I_2 \cdot \overline{P_1} \cdot P_2 \quad \text{avec} \quad P_2 \in F_2(B)$$

Résultat pour l'exemple introductif

Comportement attendu

- **R1** : Il suffit d'avoir 'i_1' pour avoir 'o_1'.
- **R2** : Il faut avoir 'i_2' pour avoir 'o_2'.
- **R3** : 'o_1' et 'o_2' ne sont jamais simultanément Vrai.



Solution paramétrée

$$\begin{cases} I_1 \leq O_1 \\ O_2 \leq I_2 \\ O_1 \cdot O_2 = 0 \end{cases} \Leftrightarrow \begin{cases} O_1 = I_1 + P_1 & P_1 \in F_2(B) \\ O_2 = \overline{I_1} \cdot I_2 \cdot \overline{P_1} \cdot P_2 & P_2 \in F_2(B) \end{cases}$$

6 Solutions possibles

$$\begin{cases} O_1 = I_1 \\ O_2 = 0 \end{cases} \quad \begin{cases} O_1 = I_1 \\ O_2 = \overline{I_1} \cdot I_2 \end{cases} \quad \begin{cases} O_1 = I_1 + \overline{I_2} \\ O_2 = 0 \end{cases}$$

$$\begin{cases} O_1 = I_1 + \overline{I_2} \\ O_2 = \overline{I_1} \cdot I_2 \end{cases} \quad \begin{cases} O_1 = I_1 + I_2 \\ O_2 = 0 \end{cases} \quad \begin{cases} O_1 = 1 \\ O_2 = 0 \end{cases}$$

Synthèse algébrique de contrôleurs logiques

Origine de cette méthode

- Résultat d'une trajectoire de recherche

Introduction

- Définition et modèle mathématique
- Exemple introductif

Cadre mathématique

- Structure d'algèbre de Boole
- Principales caractéristiques
- Résolution d'équations dans une algèbre de Boole

Synthèse d'un système logique

- Méthode de synthèse proposée
- Formalisation du cahier des charges
- Exemple de résolution

Traitement d'une étude de cas

- Commande d'un portail automatique

Logiciel support : BESS

- Interface opérateur
- Structure de données :
  - Shared Reduced Ordered Binary Decision Diagrams

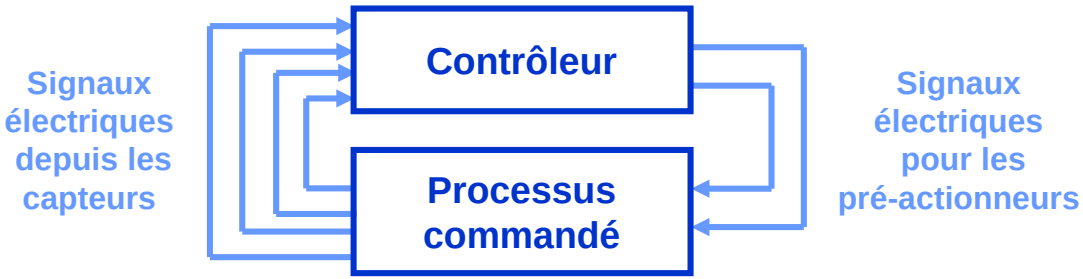
Commande d'un portail automatique de parking

Comportement attendu

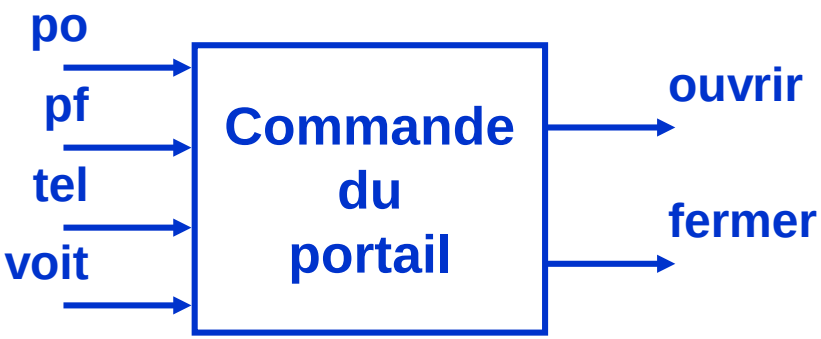
- Ouverture à distance (télécommande)
- Fermeture automatique après le passage de la voiture

Processus commandé

- Portail avec 2 capteurs de fin de course
- Moteur d'entraînement avec 2 contacteurs
- Récepteur de la télécommande
- Capteur de détection des voitures



Définition du système étudié : vue externe (entrées / sorties)



	Nature	Commentaire
ouvrir	Sortie	Commande d'ouverture du portail
fermer	Sortie	Commande de fermeture du portail
po	Entrée	Portail entièrement ouvert
pf	Entrée	Portail entièrement fermé
tel	Entrée	Demande d'ouverture du portail par l'intermédiaire de la télécommande
voit	Entrée	Détection d'une voiture

Cahier des charges du système

Sécurité des usagers

- S1 : La fermeture du portail est impossible lorsqu'une voiture est détectée.
- S2 : La fermeture du portail est impossible lorsque la télécommande est activée.

Sécurité des locaux

- S3 : Lorsque le portail est fermé, pour démarrer son ouverture, il faut une demande via la télécommande.

Aspect fonctionnel

- F1 : Il suffit que la télécommande soit activée pour que le portail s'ouvre.
- F2 : Pour arrêter l'ouverture du portail, il faut qu'il soit entièrement ouvert (toute ouverture est complète).
- F3 : L'arrêt complet du portail n'est possible qu’en fin de course.

Définition du système étudié : vue interne (variables d'état)

Recherche de la nature du système

- **Est-ce un système combinatoire ?**
  - Système pour lequel la valeur de chaque sortie à tout instant ti est déterminée à partir de la valeur des entrées à cet instant ti.

- **Est-ce un système séquentiel ?**
  - Système pour lequel la valeur de chaque sortie à tout instant ti est déterminée à partir
    - de la valeur des entrées à cet instant ti,
    - et de la valeur antérieure des entrées.
  - Comment caractériser l'état interne du système étudié ?
    - Ensemble des informations prenant en compte la mémorisation des évolutions passées et permettant de déterminer les évolutions futures.

La réponse est dans le cahier des charges...

Cahier des charges du système

Sécurité des usagers

- S1 : La fermeture du portail est impossible lorsqu'une voiture est détectée.
- S2 : La fermeture du portail est impossible lorsque la télécommande est activée.

Sécurité des locaux

- S3 : Lorsque le portail est fermé, pour démarrer son ouverture, il faut une demande via la télécommande.

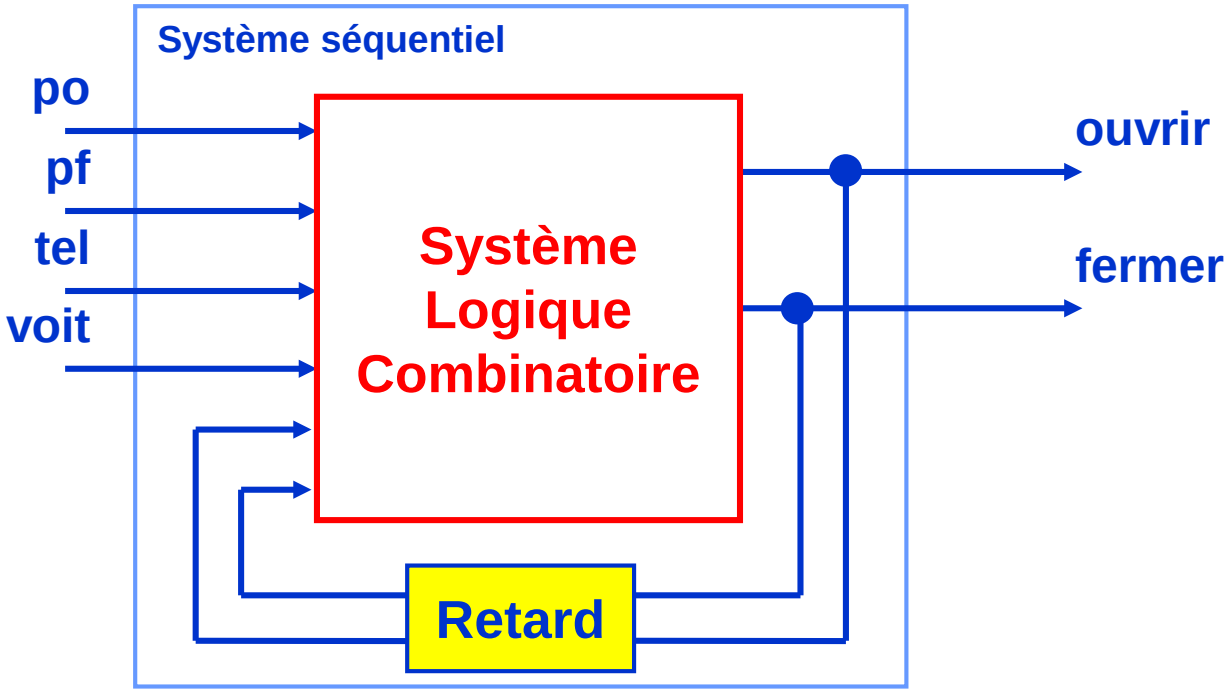
Aspect fonctionnel

- F1 : Il suffit que la télécommande soit activée pour que le portail s'ouvre.
- F2 : Pour arrêter l'ouverture du portail, il faut qu'il soit entièrement ouvert (toute ouverture est complète).
- F3 : L'arrêt complet du portail n'est possible qu'en fin de course.

Nature de la commande du système

- **Séquentiel** (d'après le cahier du charges)
  - Indices : « Pour démarrer l'ouverture », « Pour arrêter l'ouverture »

Modèle logique retenu pour le système



$$\begin{cases} \text{ouvrir}[k] = \textbf{Ouvrir}(\text{po}[k], \text{pf}[k], \text{tel}[k], \text{voit}[k], \text{ouvrir}[k-1], \text{fermer}[k-1]) \\ \text{fermer}[k] = \textbf{Fermer}(\text{po}[k], \text{pf}[k], \text{tel}[k], \text{voit}[k], \text{ouvrir}[k-1], \text{fermer}[k-1]) \end{cases}$$

Comment trouver ces 2 fonctions booléennes ?

Problème à résoudre

Trouver dans $F_6(B)$, les 2 fonctions booléennes recherchées

$$\begin{cases} \text{ouvrir[k]} = \text{Ouvrir(po[k], pf[k], tel[k], voit[k], ouvrir[k−1], fermer[k−1])} \\ \text{fermer[k]} = \text{Fermer(po[k], pf[k], tel[k], voit[k], ouvrir[k−1], fermer[k−1])} \end{cases}$$

$$|F_6(B)| = 2^{2^6} = 2^{64}$$

Approches classiques

- Identification d'une solution
  - Êtes-vous chanceux ?
    - 1 chance sur $(2^{64})^2 = 2^{128} \approx 3.4 \times 10^{38}$
- Renseignement de 2 tableaux de Karnaugh (64 cases) simultanément
  - 1 pour Ouvrir, 1 pour Fermer

Approche proposée

- Résolution (via l'outil Bess) d'un système d'équations à 2 inconnues

Il faut cependant poser le système d'équations...

Support des calculs : Algèbre de Boole des fonctions booléennes de dimension 6

Eléments de l'algèbre : Fonctions booléennes de dimension 6

$$F_6(B) = \{f : B^6 \rightarrow B\}$$

$$B = \{0, 1\}$$

$$|F_6(B)| = 2^{2^6} = 2^{64}$$

- 6 Fonctions « Projection »

$$f^i_{Proj}: \quad B^6 \rightarrow B$$

$$(b_1, \cdots, b_6) \mapsto b_i$$

PO, PF, Tel, Voit, OuvPrec, FerPrec

OuvPrec :

$$B^6 \rightarrow B$$

(po[k], pf[k], tel[k], voit[k], ouvrir[k−1], fermer[k−1])↦ ouvrir[k−1]

FerPrec :

$$B^6 \rightarrow B$$

(po[k], pf[k], tel[k], voit[k], ouvrir[k−1], fermer[k−1])↦ fermer[k−1]

Formalisation du cahier des charges (1/2)

Sécurité des usagers

- S1 : La fermeture du portail est impossible lorsqu'une voiture est détectée.

$$\textbf{Fermer} \cdot \textbf{Voit} = 0$$

- S2 : La fermeture du portail est impossible lorsque la télécommande est activée.

$$\textbf{Fermer} \cdot \textbf{Tel} = 0$$

Sécurité des locaux

- S3 : Lorsque le portail est fermé, pour démarrer son ouverture, il faut une demande via la télécommande.

$$\textbf{PF} \cdot \uparrow \textbf{Ouvrir} \leq \textbf{Tel}$$

où $\uparrow \text{Ouvrir} = \text{Ouvrir} \cdot \overline{\text{OuvPrec}}$

Formalisation du cahier des charges (2/2)

Aspect fonctionnel

- F1 : Il suffit que la télécommande soit activée pour que le portail s'ouvre.

$$\textbf{Tel} \leq \textbf{Ouvrir}$$

- F2 : Pour arrêter l'ouverture du portail, il faut qu'il soit entièrement ouvert.

$$\downarrow \textbf{Ouvrir} \leq \textbf{PO}$$

où $\downarrow \text{Ouvrir} = \overline{\text{Ouvrir}} \cdot \text{OuvPrec}$

- F3 : L'arrêt complet du portail n'est possible qu'en fin de course.

$$\overline{\textbf{Fermer}} \cdot \overline{\textbf{Ouvrir}} \leq \textbf{PO + PF}$$

Compléments au cahier des charges

Aspect technologique

- T1 : Le moteur ne doit pas être commandé simultanément dans les deux sens.

$$\textbf{Ouvrir} \cdot \textbf{Fermer} = 0$$

- T2 : L'ouverture du portail est impossible lorsqu'il est entièrement ouvert.

$$\textbf{Ouvrir} \cdot \textbf{PO} = 0$$

- T3 : La fermeture du portail est impossible lorsqu'il est entièrement fermé.

$$\textbf{Fermer} \cdot \textbf{PF} = 0$$

Résolution du problème (1/2)

Récapitulatif

- Inconnues du problème :
  - Ouvrir
  - Fermer
- Données du problème :
  - PO, PF, Tel, Voit, OuvPrec, FerPrec
- Alias utilisés : ↑ Ouvrir, ↓ Ouvrir
  - ↑ Ouvrir = Ouvrir · $\overline{\text{OuvPrec}}$
  - ↓ Ouvrir = $\overline{\text{Ouvrir}}$ · OuvPrec

Système d'équations

$$\begin{cases} \text{S1} : & \text{Fermer} \cdot \text{Voit} = 0 \\ \text{S2} : & \text{Fermer} \cdot \text{Tel} = 0 \\ \text{S3} : & \text{PF} \cdot \uparrow\text{Ouvrir} \leq \text{Tel} \\ \text{T1} : & \text{Ouvrir} \cdot \text{Fermer} = 0 \\ \text{T2} : & \text{Ouvrir} \cdot \text{PO} = 0 \\ \text{T3} : & \text{Fermer} \cdot \text{PF} = 0 \\ \text{F1} : & \text{Tel} \leq \text{Ouvrir} \\ \text{F2} : & \downarrow\text{Ouvrir} \leq \text{PO} \\ \text{F3} : & \overline{\text{Fermer}} \cdot \overline{\text{Ouvrir}} \leq \text{PO} + \text{PF} \end{cases}$$

Résolution

- **Le système d'équations n'admet pas de solutions.**

Condition d'existence de solutions : **PO · Tel = 0**

Les spécifications 'T2' et 'F1' sont incompatibles.

T2 : L'ouverture du portail est impossible lorsqu'il est entièrement ouvert.

F1 : Il suffit que la télécommande soit activée pour que le portail s'ouvre.

Résolution du problème (2/2)

Incompatibilité des spécifications ‘T2’ et ‘F1’

- Introduction d’une règle de priorité :
  - Les contraintes liés à la technologie sont prioritaires sur les attentes fonctionnelles.

$$T2 \gg F1$$

Nouvelle résolution

$$\begin{cases} S(Ouvrir) = \overline{PO}\cdot(Tel + OuvPrec + \overline{PF}\cdot Voit) + P_{Ouvrir}\cdot(\overline{PO}\cdot\overline{PF}\cdot\overline{Voit}\cdot\overline{Tel}\cdot\overline{OuvPrec}) \\ S(Fermer) = \overline{PF}\cdot\overline{Voit}\cdot\overline{Tel}\cdot(\overline{PO}\cdot\overline{OuvPrec}\cdot\overline{P_{Ouvrir}} + P_{Fermer}\cdot PO) \end{cases}$$

Le cahier des charges est incomplet.

Pas de spécification pour : $(\overline{PO}\cdot\overline{PF}\cdot\overline{Voit}\cdot\overline{Tel}\cdot\overline{OuvPrec}) + (PO\cdot\overline{PF}\cdot\overline{Voit}\cdot\overline{Tel})$

Incomplétude du cahier des charges

- Ajout d’une spécification supplémentaire non prioritaire
  - F4 : Il suffit que le portail ne soit pas entièrement fermé, pour que le portail se ferme.

Formalisation complète du problème

Problème complet

$$
\begin{cases}
S1 : & \text{Fermer} \cdot \text{Voit} = 0 \\
S2 : & \text{Fermer} \cdot \text{Tel} = 0 \\
S3 : & \text{PF} \cdot \uparrow\text{Ouvrir} \leq \text{Tel} \\
T1 : & \text{Ouvrir} \cdot \text{Fermer} = 0 \\
T2 : & \text{Ouvrir} \cdot \text{PO} = 0 \\
T3 : & \text{Fermer} \cdot \text{PF} = 0 \\
F1 : & \text{Tel} \leq \text{Ouvrir} \\
F2 : & \downarrow\text{Ouvrir} \leq \text{PO} \\
F3 : & \overline{\text{Fermer}} \cdot \overline{\text{Ouvrir}} \leq \text{PO} + \text{PF} \\
F4 : & \overline{\text{PF}} \leq \text{Fermer} \\
 & \text{T2} \gg \text{F1} \\
 & \{\text{T1,T2,S1,S2,F2}\} \gg \text{F4}
\end{cases}
$$

Solution du problème

$$
\begin{cases}
S(\text{Ouvrir}) = \overline{\text{PO}} \cdot (\text{Tel} + \text{OuvPrec} + \overline{\text{PF}} \cdot \text{Voit}) \\
S(\text{Fermer}) = \overline{\text{PF}} \cdot \overline{\text{Voit}} \cdot \overline{\text{Tel}} \cdot (\overline{\text{OuvPrec}} + \text{PO})
\end{cases}
$$

Synthèse algébrique de contrôleurs logiques

Origine de cette méthode

- Résultat d'une trajectoire de recherche

Introduction

- Définition et modèle mathématique
- Exemple introductif

Cadre mathématique

- Structure d'algèbre de Boole
- Principales caractéristiques
- Résolution d'équations dans une algèbre de Boole

Synthèse d'un système logique

- Méthode de synthèse proposée
- Formalisation du cahier des charges
- Exemple de résolution

Traitement d'une étude de cas

- Commande d'un portail automatique

Logiciel support : BESS

- Interface opérateur
- Structure de données :
  - Shared Reduced Ordered Binary Decision Diagrams

Logiciel BESS : Boolean Equations System Solver

Attente

- Réaliser automatiquement toutes les opérations symboliques

Données d'entrée :

- 'Symbols' :
  - 'Unknown' : les FB à déterminer
  - 'Known' : les m FB "Projection"
  - 'Alias' : Compositions des précédents types
- 'Requirements'
  - Relations décrivant les exigences du CdC
- 'Assumptions' (Optionnel)
  - Relations connues entre les FB "Projection"
- 'Priorities' (Optionnel)
  - Priorités déclarées entre les exigences
- 'Critères d'optimisation' (Optionnel)
  - Maximisation|Minimisation d'une expression

```
<PROBLEM>

<SYMBOLS>
# Name : [Unknown|Known|Alias] (* Comment *) ;
  O1 : Unknown ;
  O2 : Unknown ;
  I1 : Known ;
  I2 : Known ;
</SYMBOLS>

<REQUIREMENTS>
# Name : (* Comment *)
# BooleanFormula [<=|=] BooleanFormula ;
  R1 : I1 <= O1 ;
  R2 : O2 <= I2 ;
  R3 : O1.O2 = 0 ;
</REQUIREMENTS>

<OPTIMUM CRITERIA>
# Name : [Minimal|Maximal] (* Comment *)
#         BooleanFormula ;
</OPTIMUM CRITERIA>

</PROBLEM>
```

Logiciel BESS : Boolean Equations System Solver

Opérations réalisées par le logiciel

- **Lecture du problème**
  - Vérification de la syntaxe et de la cohérence des données du problème
- **Détermination de l'équation à résoudre**
  - Prise en compte des hypothèses
  - Prise en compte des priorités
- **Résolution de l'équation**
  - Résolution itérative suivant l'ordre déclaré pour les inconnues
- **Export**
  - Export au format texte de la forme paramétrée des solutions

```
<SOLUTIONS>
* O1 = I1+p_S1
* O2 = p_O2.(/p_O1./I1.I2)
</SOLUTIONS>
```

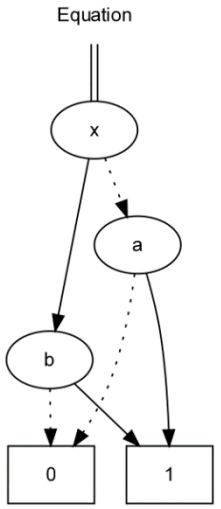
```
<SOLUTIONS>
* O2 = p_O2.(/I1.I2)
* O1 = I1+p_O1.(/I2+/p_O2)
</SOLUTIONS>
```

Module de calcul symbolique de BESS

Structure de données retenue : BDD (Binary Diagram Decision)

- ROBBD :
  - Toute expression booléenne est représentable à l'aide de ROBDD.

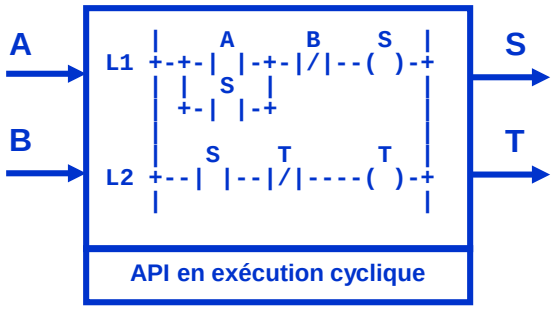
$$\underbrace{a \cdot \bar{x} + b \cdot x}_{\textbf{ROBDD}} = 0$$



- ROBBD partagé :
  - Représentation simultanée de plusieurs expressions booléennes
    - Celle décrivant l'équation
    - Celles décrivant les solutions paramétrées

- Bibliothèque ROBDD développée en Python au LURPA
  - Facilité de développement
  - Optimisation de nombreuses opérations de calcul
    - Identification de certains nœuds, parcours spécifique du ROBDD
  - Intégration de besoins spécifiques
    - Représentation graphique des BDD

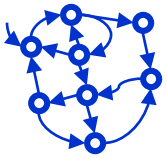
Vérification formelle par Model-checking



Propriétés à vérifier (extraites du CdC)

Formalisation

Automate à états finis



Formalisation

AG(APB→AF ~horn)

AG(~d1→AF ~lig)

Logique temporelle (LTL, CTL, …)

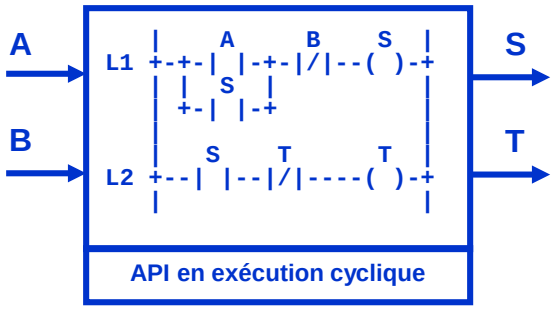
Model-checker

Propriétés vérifiées ou diagnostic en cas d'échec

Caractéristiques du Model-checking

- Approche largement utilisée en informatique
- Preuves réalisées automatiquement

Theorem-Proving par calcul symbolique



Propriétés à vérifier (extraites du CdC)

Formalisation partielle

Formalisation

$$\begin{cases} A \cdot C \Rightarrow D \\ \overline{A} \Rightarrow \overline{D} \end{cases}$$

$$Rentrer \cdot Sortir = 0^{*}$$

Algébre pour les signaux binaires

Module calcul symbolique

Propriétés vérifiées ou aide au diagnostic

Theorem Proving

- Peu développé pour nos modèles
- Limité aux propriétés de sûreté
- Applicable sur des programmes de taille importante lorsqu'ils sont bien structuré

Idées fortes

- Formaliser le modèle de commande en fonction de la propriété à démontrer
- Exploiter la structure du modèle pour simplifier la preuve de la propriétés

Qu'est ce qu'un modèle de commande ?

Rôle d'un modèle de commande

- Piloter les opérations à réaliser en fonction
  - de l'état du produit,
  - de l'état de la partie opérative,
  - des consignes données par l'opérateur.

Origine des différences entre deux modélisations

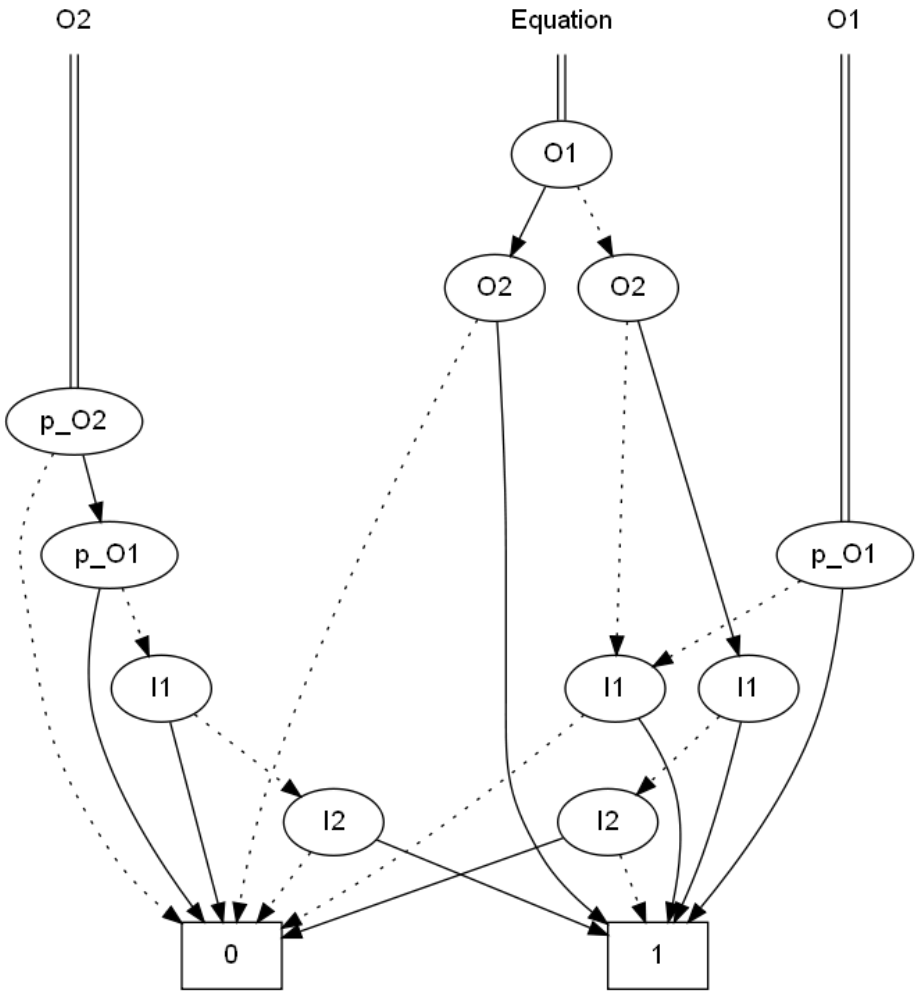
- Objectif à atteindre :
  - un modèle, une possible implantation, une réelle implantation
- Approche suivie pour sa conception
- Choix faits par le concepteur
  - choix volontaire ou involontaire ?

Méthode de synthèse algébrique

Un exemple de A à Z

Des spécifications en langage naturel aux équations récurrentes décrivant la commande

Structure BDD du problème
Équation à résoudre et Solutions du problème



Structure BDD du problème
Équation à résoudre et Solutions du problème