Calcul des conditions de transition d'un Réseau de Petri par synthèse algébrique

Yann Hietter, Jean-Marc Roussel, Jean-Jacques Lesage

Calcul des conditions de transition d'un Réseau de Petri par synthèse algébrique

Yann Hietter, Jean-Marc Roussel, Jean-Jacques Lesage IEEE member

LURPA, ENS Cachan, UniverSud
61 Av. du President Wilson, 94235 Cachan, France

{yann.hietter, jean-marc.roussel, jean-jacques.lesage}@lurpa.ens-cachan.fr
http://www.lurpa.ens-cachan.fr/

Résumé— **La méthode de synthèse présentée dans cette communication a été développée pour concevoir automatiquement des contrôleurs logiques. Nous montrons comment utiliser cette approche dans le cas particulier où un concepteur cherche à obtenir un contrôleur à partir d'un modèle générique. L'instanciation du modèle est obtenue par synthèse algébrique. Pour illustrer cette approche, l'exemple d'une chaîne de production, dont le modèle générique du contrôleur est donné par un RdPI, est utilisé.**

Mots-clés— **Systèmes à évènements discrets, synthèse algébrique, Réseau de Petri Interprété.**

I. Introduction

Les contrôleurs logiques sont utilisés dans un grand nombre de systèmes critiques, comme les systèmes embarqués, les systèmes de transport ou les systèmes de production d'énergie. Ceci explique pourquoi la sûreté des contrôleurs logiques est une préoccupation majeure des concepteurs de commande. Pour améliorer la sûreté de ces contrôleurs, plusieurs méthodes formelles ont été proposées durant ces vingt dernières années (langages formels, synthèse automatique, vérification par model-checking, génération automatique de test, diagnostic, ...). En terme de sûreté, ces approches ne sont pas concurrentes mais complémentaires.

La méthode présentée dans cette communication appartient à la classe des approches de synthèse, c.-à-d. dont l'objectif est de déduire les lois de commande à partir des spécifications et des propriétés de sûreté attendues, sans intervention du concepteur (ou en la réduisant au maximum). Les plus grands progrès dans le domaine de la synthèse des Systèmes à Évènements Discrets (SED) ont été obtenus grâce à la « Supervisory Control Theory »(SCT) définie par Ramadge et Wonham [1]. Depuis la publication de cet innovant paradigme théorique, de nombreux chercheurs ont étendu les premiers résultats sur le plan méthodologique comme sur le plan théorique. Leurs innombrables publications ont permis d'étendre les possibilités de la SCT pour concevoir un superviseur optimal mais également de préciser les limites de cette approche pour la conception et l'implémentation de contrôleurs opérationnels [2]. La méthode de synthèse que nous présentons dans cette communication est radicalement différente de la SCT pour deux raisons principales. Tout d'abord, elle est limitée à une sous-classe des SED (les systèmes logiques) et vise la conception de lois de commande (et non d'un superviseur), et surtout, elle est purement algébrique.

Dans une communication précédente [3], nous avons utilisé notre approche pour synthétiser la loi de commande complète d'un système séquentiel. Nous avons montré que différents formalismes (tels que les machines à états finis, les expressions logiques, les diagrammes temporels, ...) peuvent être utilisés conjointement pour exprimer les besoins fonctionnels, les règles de sécurité et de sûreté, ... La traduction de ces fragments de spécification en équations algébriques conduit à l'écriture d'un système d'équations dont la résolution donne l'espace des lois de commande solution. Le choix d'une loi de commande dans cet espace peut ensuite être fait en utilisant un critère d'optimisation. Dans [3], nous avons montré par exemple comment choisir la loi de commande la plus sûre.

Dans cette communication, nous allons traiter le cas spécifique (mais très fréquent) où le savoir-faire du concepteur lui permet de construire *a priori* la forme globale de la solution (ou d'une partie de la solution complète). Dans ce cas, le concepteur exprime souvent la loi de commande en utilisant un modèle générique. La solution spécifique d'un problème de commande donné est alors obtenue en instanciant le modèle générique. Nous allons maintenant montrer comment notre méthode de synthèse peut être utilisée pour trouver et optimiser ces instances de modèles génériques.

Cette communication est organisée de la manière suivante. La partie 2 donne les grandes lignes de notre méthode. Dans la partie 3, nous décrivons un exemple de contrôleur logique à concevoir. La partie 4 présente le cadre formel utilisé et les résultats mathématiques sur lesquels se base cette méthode. Notre méthode de synthèse algébrique est finalement utilisée dans la partie 5 en traitant l'exemple présenté dans la partie 3.

II. Méthode de synthèse

Une vue globale de notre méthode de synthèse est donnée figure 1. Tout d'abord, il est important de décrire les types de données d'entrée nécessaires pour la synthèse d'un contrôleur. Il s'agit de toutes les connaissances sur le système que peut exprimer le concepteur. Ces fragments de spécification peuvent être des besoins fonctionnels, des situations dangereuses qui doivent être interdites, des spécifications partielles de lois de commande, ... Différents formalismes peuvent être employés pour exprimer ces fragments de spécification : des modèles à états

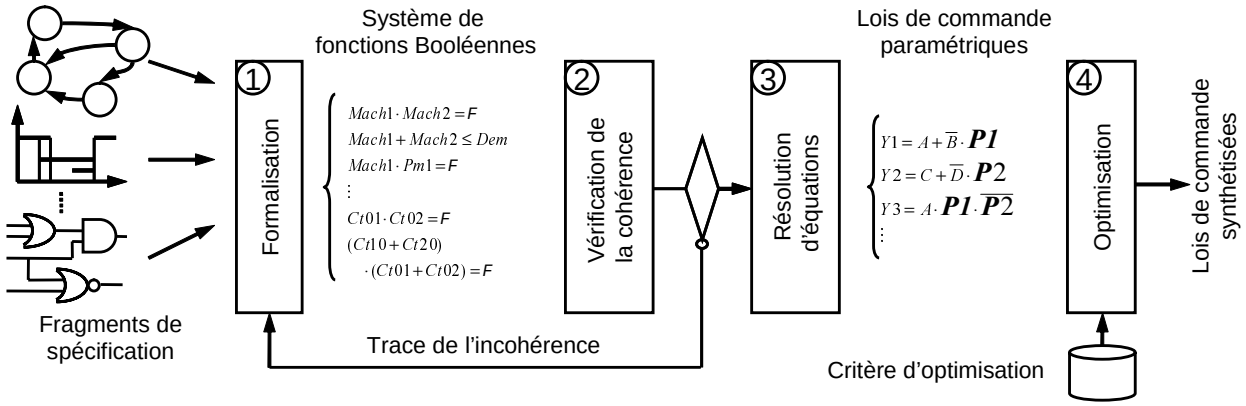

Fig. 1. Une vue globale de notre méthode de synthèse

(automates, réseaux de Petri, statecharts, ...), des diagrammes temporels, des logigrammes, ... D'une manière générale, peut être utilisé tout formalisme pouvant être traduit en équations booléennes.

La première étape de la méthode est la formalisation des fragments hétérogènes de spécification à l'aide de fonctions booléennes. Cette étape est en grande partie automatisée (voir par exemple, dans la partie 5, la traduction d'un RdPI en fonctions booléennes). À la fin de cette étape, l'ensemble des spécifications prend la forme d'un système d'équations booléennes. La deuxième étape, totalement automatique, consiste à vérifier la cohérence de ce système d'équations. Lorsqu'une incohérence est détectée, les spécifications concernées doivent être modifiées. La troisième étape, également automatique, est la résolution du système d'équations. Elle sera détaillée dans la partie 4 de cette communication. Le résultat de cette étape est une définition paramétrique de l'ensemble des lois de commande qui satisfont tous les fragments de spécification. La dernière étape de la méthode vise à choisir, parmi l'ensemble des solutions, la meilleure loi de commande pour un problème donné. Actuellement, cette étape n'est pas implémentée.

III. Description du cas d'étude

Dans cette communication, nous traitons le cas spécifique où le savoir-faire du concepteur lui permet de construire la forme globale de la solution. Nous illustrons le traitement de cette classe de problèmes avec l'exemple d'une chaîne de production (figure 2). Le système se compose de deux machines qui permettent de réaliser une même production. En mode nominal, seule une machine est utilisée. Le contrôleur logique qui est chargé de l'orientation de la production sur l'une ou l'autre des machines a 4 entrées et 2 sorties :

- **dem** : demande de pièce par l'aval de la chaîne,
- **pm1** : détection d'une panne de la machine 1,
- **pm2** : détection d'une panne de la machine 2,
- **pc** : détection d'une panne sur le convoyeur amont aux machines.

mach1 et **mach2** sont les deux signaux de sortie qui permettent de diriger le flux de pièces vers les machines.

Un spécialiste reconnaît dans cet énoncé informel et incomplet un problème de gestion de redondance. Il connaît

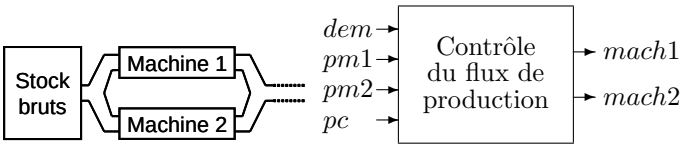

Fig. 2. (a) Chaîne de production (b) Entrées et sorties du contrôleur

différentes structures séquentielles paramétriques solutions de ce problème classique (l'une d'entre elles est donnée figure 3 sous la forme d'un Réseau de Petri Interprété (RdPI)). Le vrai problème à résoudre est en fait de trouver les paramètres de cette solution, c.-à-d. les conditions d'activation d'une seule machine ($Ct01$ et $Ct02$), les conditions de changement d'une machine pour l'autre ($Ct12$ et $Ct21$) et les conditions d'arrêter de la production ($Ct10$ et $Ct20$). *A priori*, les $Ctij$ sont des fonctions binaires des entrées et des variables d'état du RdPI.

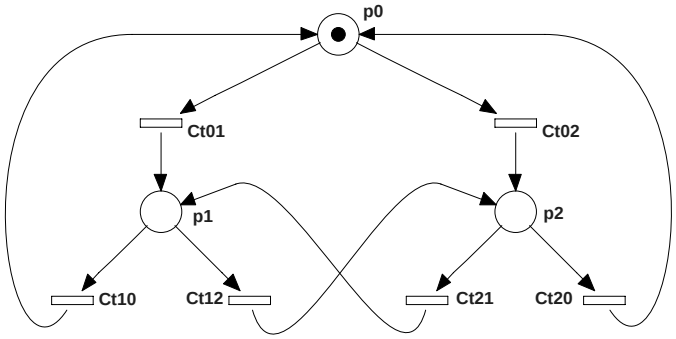

Fig. 3. Un RdPI partiel, solution experte paramétrique du contrôle de flux de production

Nous allons maintenant préciser le cadre mathématique sur lequel notre méthode de synthèse est établie.

IV. Cadre mathématique

A. Fonctions booléennes

Sur le plan mathématique, une Fonction Booléenne (FB) est une fonction G_i de $\mathbb{B}^m \rightarrow \mathbb{B}$, où $\mathbb{B} = \{0,1\}$ est le domaine booléen. Pour chaque m-uplet de variables booléennes $(b_1, ..., b_m)$ peuvent être définies $2^{(2^m)}$ FBs différentes. Soit Γ_m l'ensemble de ces $2^{(2^m)}$ FBs possibles.

Γ_m contient deux FBs particulières :

$$\begin{array}{rccc|rccc} \mathcal{F}: & \mathbb{B}^m & \rightarrow & \mathbb{B} & \mathcal{T}: & \mathbb{B}^m & \rightarrow & \mathbb{B} \\ & (b_1,...,b_m) & \mapsto & 0 & & (b_1,...,b_m) & \mapsto & 1 \end{array}$$

$\mathcal{F}$ est la FB « toujours fausse » et $\mathcal{T}$ est la FB « toujours vraie ».

Γ_m contient également m FBs spécifiques en raison de leur image qui est l'une des variables booléennes de $(b_1,...,b_m)$. Ces fonctions sont :

$$\begin{array}{rccc} B_i: & \mathbb{B}^m & \rightarrow & \mathbb{B} \\ & (b_1,...,b_m) & \mapsto & b_i \end{array}$$

Remarque : Dans cette communication, pour éviter toute confusion, les variables booléennes sont notées avec des lettres minuscules (b_i) tandis que les fonctions booléennes sont notées avec des lettres majuscules (B_i).

Γ_m muni des deux opérations binaires OU (notée « $+$ ») et ET (notée « $\cdot$ »), de l'opération unaire NON (notée « $^-$ ») et des deux éléments neutres $\mathcal{T}$ et $\mathcal{F}$ est une algèbre de Boole [4]. Ces trois opérations sont ainsi définies :

$$\begin{array}{l} \forall G_1, G_2 \in \Gamma_m, \forall (b_1,...,b_m) \in \mathbb{B}^m \\ (G_1+G_2)(b_1,...,b_m) = (G_1(b_1,...,b_m)) \vee (G_2(b_1,...,b_m)) \\ (G_1 \cdot G_2)(b_1,...,b_m) = (G_1(b_1,...,b_m)) \wedge (G_2(b_1,...,b_m)) \\ \overline{G}(b_1,...,b_m) = \neg(G(b_1,...,b_m)) \end{array}$$

où $\vee$, $\wedge$, $\neg$ sont les opérations logiques sur les variables booléennes.

Remarque : Pour alléger les notations, la référence au m-uplet $(b_1,...,b_m)$ dans la notation des FBs sera dorénavant omise.

B. Composition de FBs

Puisque $(\Gamma_m, +, \cdot, ^-, \mathcal{F}, \mathcal{T})$ est une algèbre de Boole, chaque composition d'éléments de Γ_m par les opérations OU, ET et NON est également un élément de Γ_m. Deux compositions distinctes d'éléments de Γ_m sont dites « égales » si elles ont la même image dans Γ_m.

Soient $(G_1, \ldots, G_k)$ k éléments de Γ_m. Pour chaque composition C de $(G_1, \ldots, G_k)$, nous avons prouvé qu'il existe deux compositions C_1 et C_2 de $(G_2, \ldots, G_k)$ telles que :

$$C(G_1,\ldots,G_k) = C_1(G_2,\ldots,G_k)\cdot\overline{G_1} + C_2(G_2,\ldots,G_k)\cdot G_1$$

Ces compositions C_1 et C_2 s'obtiennent à partir de C par les substitutions suivantes :

$$\left\{ \begin{array}{l} C_1(G_2,\ldots,G_k) = C(\mathcal{F},G_2,\ldots,G_k) \\ C_2(G_2,\ldots,G_k) = C(\mathcal{T},G_2,\ldots,G_k) \end{array} \right.$$

Cette propriété est l'extension aux FBs de la décomposition proposée par Shannon [5] pour les variables booléennes. Grâce à cette propriété, toute composition de FBs peut être exprimée sous une forme générique.

C. Relations entre FBs

$(\Gamma_m, +, \cdot, ^-, \mathcal{F}, \mathcal{T})$ étant une algèbre de Boole, Γ_m peut être muni d'un ordre partiel [4] défini par cette relation :

Définition 1. Relation Inclusion

$$\forall G_1, G_2 \in \Gamma_m, \quad G_1 \leq G_2 \Leftrightarrow G_1 \cdot G_2 = G_1$$

Les FBs G_1 et G_2 satisfont la relation $G_1 \leq G_2$ (lire « G_1 est inclue dans G_2 ») si et seulement si l'ensemble des m-uplets $(b_1,...,b_m)$ dont l'image par G_1 est 1 est inclus dans l'ensemble des m-uplets $(b_1,...,b_m)$ dont l'image par G_2 est 1.

Cette relation est la pierre angulaire de notre approche car elle permet d'exprimer simplement de nombreuses exigences d'un cahier des charges. Par exemple, la relation $C \leq D$ peut être la formalisation des trois assertions suivantes :

- Quand C est vraie, D est aussi vraie.
- Il suffit d'avoir C pour avoir D.
- Il est nécessaire d'avoir D pour avoir C.

Nous utilisons par ailleurs la relation « égalité » pour formaliser des assertions comme :

- C et D ne sont jamais simultanément vraies : $C \cdot D = \mathcal{F}$
- L'une des FBs C et D est toujours vraie : $C + D = \mathcal{T}$

Il est important de noter que les relations exprimées en employant « l'inclusion » et « l'égalité » peuvent être réécrites [4] :

$$G_1 \leq G_2 \quad \Leftrightarrow \quad G_1 \cdot G_2 = G_1 \tag{1}$$

$$G_1 \leq G_2 \quad \Leftrightarrow \quad G_1 \cdot \overline{G_2} = \mathcal{F} \tag{2}$$

$$G_1 = G_2 \quad \Leftrightarrow \quad G_1 \cdot \overline{G_2} + \overline{G_1} \cdot G_2 = \mathcal{F} \tag{3}$$

$$\left\{ \begin{array}{l} G_1 = \mathcal{F} \\ G_2 = \mathcal{F} \end{array} \right. \quad \Leftrightarrow \quad G_1 + G_2 = \mathcal{F} \tag{4}$$

- L'équivalence (2) permet de réécrire chaque inclusion comme une égalité dont le second membre est $\mathcal{F}$.
- L'équivalence (3) permet de réécrire chaque égalité comme une égalité dont le second membre est $\mathcal{F}$.
- L'équivalence (4) permet de réécrire un ensemble d'égalités comme une seule égalité dont le second membre est $\mathcal{F}$.

Ainsi, toute expression liant des FBs connues B_i et des FBs inconnues Y_j peut toujours être ramenée sous la forme suivante :

$$R(B_1,\ldots,B_m,Y_1,\ldots,Y_n) = \mathcal{F} \tag{5}$$

où $R(B_1,\ldots,B_m,Y_1,\ldots,Y_n)$ (comme « Requirements ») est une composition des FBs B_i et Y_j par les opérations OU, ET et NON.

Pour simplifier encore les notations, nous noterons dans la suite $R(Y_j)$ en omettant les B_i.

D. Résolution d'un système d'équations de FBs

Les résultats présentés dans cette partie ont été démontrés pour toute algèbre de Boole. Ils sont donc vrais pour l'algèbre des FBs. Les démonstrations proposées sont la généralisation de précédents résultats obtenus pour des équations entre des variables booléennes :

- La condition pour qu'une équation entre des variables booléennes ait une unique solution a été donnée dans [6].
- Le cas de systèmes d'équations entre des variables booléennes a été étudié dans [7].
- La forme générale de l'ensemble des solutions pour des équations avec plusieurs inconnues booléennes a été donnée dans [8].

Avant de présenter le cas général à n inconnues, nous allons tout d'abord nous intéresser au cas des équations entre des FBs avec une seule inconnue.

D.1 Cas d'une seule inconnue : $R(Y) = \mathcal{F}$

En appliquant la généralisation de la décomposition de Shannon à $R(Y)$, nous avons :

$$R(Y) = \mathcal{F} \quad \Leftrightarrow \quad R(\mathcal{F}) \cdot \overline{Y} + R(\mathcal{T}) \cdot Y = \mathcal{F} \qquad (6)$$

Théorème 1. Soit $R(Y) = \mathcal{F}$ une équation comportant une seule inconnue Y exprimée sur l'algèbre de Boole $(\Gamma_m, +, \cdot, \bar{}, \mathcal{F}, \mathcal{T})$ des FBs.
$R(Y) = \mathcal{F}$ admet des solutions Y si et seulement si :

$$R(\mathcal{F}) \cdot R(\mathcal{T}) = \mathcal{F}$$

Dans ce cas, la solution paramétrique de cette équation est :

$$Y = R(\mathcal{F}) + \overline{R(\mathcal{T})} \cdot P$$

où P est une FB de Γ_m dont l'ensemble des valeurs possibles permet de décrire l'espace entier des solutions.

Lorsque $R(\mathcal{F}) + R(\mathcal{T}) = \mathcal{T}$, $R(Y) = \mathcal{F}$ admet une seule solution : $Y = R(\mathcal{F})$.

Démonstration : La démonstration de ce résultat comporte trois temps permettant d'établir que :
- $R(\mathcal{F}) \cdot R(\mathcal{T}) = \mathcal{F}$ est une condition nécessaire et suffisante d'existence de solutions de (6).
- La forme proposée $Y = R(\mathcal{F}) + \overline{R(\mathcal{T})} \cdot P$ est bien une solution de (6).
- Toutes les solutions de (6) peuvent s'exprimer sous la forme précédente.

Pour faciliter la lecture de cette démonstration, nous noterons $R(\mathcal{F}) = A$ et $R(\mathcal{T}) = B$. Avec ces notations, l'équation étudiée est :

$$A \cdot \overline{Y} + B \cdot Y = \mathcal{F} \qquad (7)$$

La condition d'existence de solutions est $A \cdot B = \mathcal{F}$ et la forme des solutions est $Y = A + \overline{B} \cdot P$.
- $A \cdot B = \mathcal{F}$ est une condition nécessaire et suffisante d'existence de solutions de (7) :

$$\begin{aligned}
& A \cdot \overline{Y} + B \cdot Y = \mathcal{F} \\
\Leftrightarrow \quad & A \cdot \overline{Y} \cdot \mathcal{T} + B \cdot Y \cdot \mathcal{T} = \mathcal{F} \\
\Leftrightarrow \quad & A \cdot \overline{Y} \cdot (B + \mathcal{T}) + B \cdot Y \cdot (A + \mathcal{T}) = \mathcal{F} \\
\Leftrightarrow \quad & A \cdot B \cdot (\overline{Y} + Y) + A \cdot \overline{Y} \cdot \mathcal{T} + B \cdot Y \cdot \mathcal{T} = \mathcal{F} \\
\Leftrightarrow \quad & A \cdot B \cdot \mathcal{T} + A \cdot \overline{Y} + B \cdot Y = \mathcal{F} \\
\Leftrightarrow \quad & \begin{cases} A \cdot B = \mathcal{F} \\ A \cdot \overline{Y} + B \cdot Y = \mathcal{F} \end{cases}
\end{aligned}$$

La condition $A \cdot B = \mathcal{F}$ est donc nécessaire pour l'existence de solutions de (7). Cette condition est également suffisante, car lorsqu'elle est vérifiée, $Y = A$ est une solution.

$$\begin{cases} A \cdot B = \mathcal{F} \\ Y = A \end{cases} \Rightarrow$$
$$A \cdot \overline{Y} + B \cdot Y = A \cdot \overline{A} + B \cdot A = \mathcal{F} + A \cdot B = A \cdot B = \mathcal{F}$$

- La forme proposée $Y = A + \overline{B} \cdot P$ est bien une solution de (7) :

$$\begin{cases} A \cdot B = \mathcal{F} \\ Y = A + \overline{B} \cdot P \end{cases} \Rightarrow$$
$$\begin{aligned}
A \cdot \overline{Y} + B \cdot Y &= A \cdot \overline{(A + \overline{B} \cdot P)} + B \cdot (A + \overline{B} \cdot P) \\
&= A \cdot \overline{A} \cdot (\overline{B} \cdot P) + (A \cdot B + B \cdot \overline{B} \cdot P) \\
&= \mathcal{F} + A \cdot B + \mathcal{F} = A \cdot B = \mathcal{F}
\end{aligned}$$

- Toutes les solutions de (7) peuvent s'exprimer sous cette forme.

Pour démontrer ce point, il est suffisant de trouver, pour chaque solution de (7), un élément P de Γ_m qui satisfait $Y = A + \overline{B} \cdot P$

Considérons P défini à partir de Y de la façon suivante : $P = \overline{A} \cdot Y$ et montrons que Y est exprimable sous la forme proposée.

$$\begin{aligned}
A \cdot \overline{Y} + B \cdot Y = \mathcal{F} \quad \Leftrightarrow \quad & \begin{cases} A \cdot \overline{Y} = \mathcal{F} \\ B \cdot Y = \mathcal{F} \end{cases} \quad \Rightarrow \\
Y \quad &= (A + B + \overline{A} \cdot \overline{B}) \cdot Y \\
&= A \cdot Y + B \cdot Y + \overline{A} \cdot \overline{B} \cdot Y \\
&= A \cdot Y + \mathcal{F} + \overline{A} \cdot \overline{B} \cdot Y \\
&= A \cdot Y + A \cdot \overline{Y} + \overline{A} \cdot \overline{B} \cdot Y \\
&= A + \overline{B} \cdot (\overline{A} \cdot Y) = A + \overline{B} \cdot P \qquad \square
\end{aligned}$$

D.2 Cas de plusieurs inconnues : $R(Y_1, \ldots, Y_n) = \mathcal{F}$

En partant du cas à une seule inconnue, nous avons généralisé ce résultat au cas à une équation avec n inconnues. Pour 2 inconnues, la forme générale de (5) est :

$$\begin{aligned}
R(Y_1, Y_2) = {} & R(\mathcal{F}, \mathcal{F}) \cdot (\overline{Y_1} \cdot \overline{Y_2}) + R(\mathcal{F}, \mathcal{T}) \cdot (\overline{Y_1} \cdot Y_2) \\
& + R(\mathcal{T}, \mathcal{F}) \cdot (Y_1 \cdot \overline{Y_2}) + R(\mathcal{T}, \mathcal{T}) \cdot (Y_1 \cdot Y_2) = \mathcal{F}
\end{aligned}$$

$R(Y_1, Y_2)$ prend la forme d'une somme de 2^2 termes (autant de termes que de combinaisons sur $\{Y_1, Y_2\}$). Chacun de ces termes (par exemple : $R(\mathcal{F}, \mathcal{T}) \cdot (\overline{Y_1} \cdot Y_2)$) est le produit de deux expressions :
- $\overline{Y_1} \cdot Y_2$ est l'une des 2^2 combinaisons sur $\{Y_1, Y_2\}$.
- $R(\mathcal{F}, \mathcal{T})$ est l'image par R des valeurs ($\mathcal{F}$ ou $\mathcal{T}$) de la même combinaison sur $\{Y_1, Y_2\}$.

$$R(\mathcal{F}, \mathcal{T}) \cdot (\overline{Y_1} \cdot Y_2) \qquad \begin{matrix} Y_2 = \mathcal{T} \\ Y_1 = \mathcal{F} \end{matrix}$$

Pour n inconnues, la forme générale de $R(Y_1, ..., Y_n) = \mathcal{F}$ comporte 2^n termes :

$$\begin{aligned}
R(Y_1, Y_2, \ldots, Y_n) = {} & R(\mathcal{F}, \mathcal{F}, \ldots, \mathcal{F}) \cdot (\overline{Y_1} \cdot \overline{Y_2} \cdot \ldots \cdot \overline{Y_n}) \\
& + \ldots + R(\mathcal{T}, \mathcal{T}, \ldots, \mathcal{T}) \cdot (Y_1 \cdot Y_2 \cdot \ldots \cdot Y_n) = \mathcal{F} \qquad (8)
\end{aligned}$$

Ces 2^n termes peuvent être définis à partir de 2^n n-uplets dont les composantes sont uniquement $\mathcal{F}$ et $\mathcal{T}$. Soit Ω_n cet ensemble de n-uplets. Soit ω_j l'un de ces n-uplets et ω_j^l la $l^{ème}$ composante du n-uplet ω_j.

$$\omega_j = (\underbrace{\mathcal{F}}_{\omega_j^1}, \mathcal{T}, \ldots, \underbrace{\mathcal{T}}_{\omega_j^l}, \ldots, \mathcal{F}, \underbrace{\mathcal{T}}_{\omega_j^n})$$

Avec cette notation, $R(Y_1, \ldots, Y_n)$ peut être décrit de manière générique comme suit :

$$R(Y_1, ..., Y_n) = \sum_{j=0}^{2^n - 1} \left(R(\omega_j) \cdot \prod_{l=1}^{n} (\omega_j^l \cdot Y_l + \overline{\omega_j^l} \cdot \overline{Y_l}) \right) \qquad (9)$$

Munis de cette notation, il est alors possible d'exprimer de manière générique les solutions de (8).

Théorème 2. $R(Y_1, Y_2, \ldots, Y_n) = \mathcal{F}$ admet des solutions si et seulement si :

$$\prod_{j=0}^{2^n-1} R(\omega_j) = \mathcal{F}$$

Dans ce cas, les solutions sont :

$$\begin{cases} Y_1 = & \dfrac{\prod_{j=0}^{2^{n-1}-1} R(\mathcal{F}, \omega_j^1, \ldots \omega_j^{n-1})}{\prod_{j=0}^{2^{n-1}-1} R(\mathcal{T}, \omega_j^1, \ldots \omega_j^{n-1})} \cdot P_1 \\ \ldots & \\ Y_i = & \dfrac{\prod_{j=0}^{2^{n-i}-1} R(Y_1, \ldots, Y_{i-1}, \mathcal{F}, \omega_j^1, \ldots \omega_j^{n-i})}{\prod_{j=0}^{2^{n-i}-1} R(Y_1, \ldots, Y_{i-1}, \mathcal{T}, \omega_j^1, \ldots \omega_j^{n-i})} \cdot P_i \\ \ldots & \\ Y_n = & R(Y_1, \ldots, Y_{n-1}, \mathcal{F}) + \overline{R(Y_1, \ldots, Y_{n-1}, \mathcal{T})} \cdot P_n \end{cases}$$

où $P_1, \ldots, P_n$ sont des FBs de Γ_m dont l'ensemble des valeurs possibles permet de décrire l'espace entier des solutions.

Ce théorème (non démontré ici, faute de place) a été obtenu en utilisant successivement n fois le théorème 1. Pour ce faire, l'équation (9) est considérée comme une équation à une seule inconnue Y_n, dont la condition d'existence de solutions conduit à une équation à $(n-1)$ inconnues. Cette nouvelle équation est alors considérée comme une équation à une seule inconnue Y_{n-1}... Une fois déterminée la forme paramétrique de Y_1, celle-ci est reportée dans l'équation décrivant Y_2 et ainsi de suite.

V. Traitement de l'exemple

Appliquons notre méthode de synthèse au calcul des six conditions de transition du RdPI donné à la figure 3.

La première étape est la formalisation, à l'aide de FBs, de toutes les données du problème : les besoins fonctionnels et la solution paramétrique donnée par le concepteur (le RdPI).

Les besoins fonctionnels de la chaîne de production peuvent être formalisés ainsi :

- Les deux machines ne sont jamais utilisées en même temps.

$$(S1): \quad Mach1 \cdot Mach2 = \mathcal{F}$$

- Une demande de l'aval est nécessaire pour que la production ait lieu.

$$(S2): \quad Mach1 + Mach2 \leq Dem$$

- La machine 1 est inutilisable lorsqu'elle est en panne.

$$(S3): \quad Mach1 \cdot Pm1 = \mathcal{F}$$

- La machine 2 est inutilisable lorsqu'elle est en panne.

$$(S4): \quad Mach2 \cdot Pm2 = \mathcal{F}$$

- Aucune des machines ne peut être utilisée lorsque le convoyeur est en panne.

$$(S5): \quad (Mach1 + Mach2) \cdot Pc = \mathcal{F}$$

- En l'absence de panne, il suffit d'une demande pour que la production ait lieu.

$$(S6): \quad \overline{(Pc + Pm1 \cdot Pm2)} \cdot Dem \leq Mach1 + Mach2$$

- La permutation de machine est interdite en l'absence de panne sur les machines.

$$(S7): \quad \overline{Pm1} \cdot \overline{Pm2} \cdot (Ct12 + Ct21) = \mathcal{F}$$

Le RdPI, solution générique proposée par le concepteur, doit également être traduit en FBs. Ce RdPI est sauf et possède un invariant de marquage : $M(P_0) + M(P_1) + M(P_2) = 1$. Il peut être formalisé à l'aide d'un ensemble d'équations récurrentes similaires à celles proposées dans [9]. De plus, le contrôle du flux de production par $Mach1$ et $Mach2$ est fonction du marquage des places du RdPI. $Mach1$ est associé au marquage de la place P_1. $Mach2$ est associé au marquage de la place P_2. Le RdPI de la figure 3 se traduit dans l'algèbre des FBs par les 8 équations suivantes :

$$\begin{cases} (CR1): & P_0(k) = P_1(k-1) \cdot Ct10 + P_2(k-1) \cdot Ct20 \\ & \quad + P_0(k-1) \cdot \overline{Ct01} \cdot \overline{Ct02} \\ (CR2): & P_1(k) = P_0(k-1) \cdot Ct01 + P_2(k-1) \cdot Ct21 \\ & \quad + P_1(k-1) \cdot \overline{Ct10} \cdot \overline{Ct12} \\ (CR3): & P_2(k) = P_0(k-1) \cdot Ct02 + P_1(k-1) \cdot Ct12 \\ & \quad + P_2(k-1) \cdot \overline{Ct20} \cdot \overline{Ct21} \\ (CR4): & Mach1 = P_1 \\ (CR5): & Mach2 = P_2 \\ (CR6): & P_0(0) = \mathcal{T} \\ (CR7): & P_1(0) = \mathcal{F} \\ (CR8): & P_2(0) = \mathcal{F} \end{cases}$$

Les équations $CR1$, $CR2$ et $CR3$ traduisent la structure et les règles d'évolution du RdPI où $P_i(k)$ est une fonction booléenne qui définit le marquage de la place P_i. $CR4$ et $CR5$ expriment l'association des sorties aux places. $CR6$, $CR7$ et $CR8$ fixent le marquage initial du RdPI.

Deux propriétés particulières doivent également être formalisées. La première exprime que le RdP doit être déterministe, c.-à-d. que les conditions associées aux transitions en aval d'une même place doivent être exclusives :

$$\begin{cases} (PA1): Ct01 \cdot Ct02 = \mathcal{F} \\ (PA2): Ct10 \cdot Ct12 = \mathcal{F} \\ (PA3): Ct21 \cdot Ct20 = \mathcal{F} \end{cases}$$

La deuxième propriété impose que les marquages atteignables du RdP doivent être stables, c'est-à-dire que deux conditions de transition successives doivent être exclusives :

$$\begin{cases} (PA4): (Ct01 + Ct21) \cdot (Ct10 + Ct12) = \mathcal{F} \\ (PA5): (Ct02 + Ct12) \cdot (Ct20 + Ct21) = \mathcal{F} \\ (PA6): (Ct10 + Ct20) \cdot (Ct01 + Ct02) = \mathcal{F} \end{cases}$$

Finalement, le système d'équations à résoudre se compose de 21 équations. 7 équations expriment les attentes fonctionnelles de la chaîne de production ($S1$ à $S7$), 8

équations traduisent la structure du RdPI ($CR1$ à $CR8$) et les 6 dernières équations sont nécessaires pour exprimer les propriétés attendues de ce RdP ($PA1$ à $PA6$). Les inconnues du problème sont les 6 conditions de transition ($Ct01$, $Ct02$, $Ct10$, $Ct12$, $Ct20$ et $Ct21$). Ce sont *a priori* des FBs des entrées et des variables d'état.

Ce système d'équations peut se ramener à une seule équation de la forme de (5). Pour trouver cette équation, chaque relation est réécrite comme une égalité dont le second membre est $\mathcal{F}$. Certaines équations ne nécessitent pas de réécriture, comme ($S1$), les inclusions comme ($S2$) sont réécrites en employant (2), et les égalités comme ($CR1$) sont réécrites par l'utilisation de (3). Finalement, en utilisant (4), toutes ces équations sont fusionnées en une seule équation (qui est trop longue pour être entièrement écrite ici) :

$$Mach1 \cdot Mach2 + (Mach1 + Mach2) \cdot \overline{Dem} \ldots = \mathcal{F} \quad (10)$$

L'équation (10) peut également être reformulée sous sa forme générique :

$$\begin{array}{l} R(\mathcal{F},\mathcal{F},\mathcal{F},\mathcal{F},\mathcal{F},\mathcal{F}) \cdot \overline{Ct01} \cdot \overline{Ct02} \cdot \overline{Ct10} \cdot \overline{Ct12} \cdot \overline{Ct20} \cdot \overline{Ct21} + \ldots \\ + R(\mathcal{T},\mathcal{T},\mathcal{T},\mathcal{T},\mathcal{T},\mathcal{T}) \cdot Ct01 \cdot Ct02 \cdot Ct10 \cdot Ct12 \cdot Ct20 \cdot Ct21 = \mathcal{F} \end{array}$$

où, par exemple :

$$\begin{array}{rl} R(\mathcal{F},\mathcal{F},\mathcal{F},\mathcal{F},\mathcal{F},\mathcal{F}) = & (P_1(k-1) + P_2(k-1)) \cdot (\overline{Dem} + Pc) \\ & + P_1(k-1) \cdot Pm1 + P_2(k-1) \cdot Pm2 \\ & + \overline{P_1(k-1)} \cdot \overline{P_2(k-1)} \cdot Dem \cdot \overline{Pc} \cdot (\overline{Pm1} + \overline{Pm2}) \end{array}$$

En utilisant les résultats du théorème 2, l'existence de solutions est d'abord vérifiée. La forme paramétrique de ces solutions est la suivante :

$$\left\{ \begin{array}{l} Ct01 = Dem \cdot \overline{Pc} \cdot \overline{Pm1} \cdot (Pm2 + P_{Ct01}) \\ Ct02 = Dem \cdot \overline{Pc} \cdot \overline{Pm2} \cdot (Pm1 + \overline{P_{Ct01}}) \\ Ct10 = \overline{Dem} + Pc + Pm1 \cdot Pm2 \\ Ct12 = Dem \cdot \overline{Pc} \cdot Pm1 \cdot \overline{Pm2} \\ Ct20 = \overline{Dem} + Pc + Pm1 \cdot Pm2 \\ Ct21 = Dem \cdot \overline{Pc} \cdot Pm2 \cdot \overline{Pm1} \end{array} \right.$$

où P_{Ct01} est une FB « paramètre » de Γ_m , dont l'ensemble des valeurs décrit l'espace entier des solutions.

La dernière étape de notre approche est le choix d'une solution spécifique. Seules les deux conditions de transition $Ct01$ et $Ct02$, qui doivent permettre d'orienter les flux vers l'une des deux machines en l'absence de panne, dépendent d'un paramètre. Pour fixer ce paramètre, plusieurs stratégies sont possibles. La première solution est d'accorder la priorité à une machine donnée. Si la machine 1 a la priorité sur la machine 2 (respect. la machine 2 a la priorité sur la machine 1), $Ct01$ doit être maximisée en fixant $P_{Ct01} = \mathcal{T}$ (respect. $P_{Ct01} = \mathcal{F}$).

Un autre objectif de production peut être d'équilibrer le temps de travail des deux machines. Dans ce cas, une nouvelle entrée peut être ajoutée (par exemple « c », comme clock, qui change de valeur toutes les 12 heures). Quand $c = 1$, la machine 1 doit être utilisée (la machine 2 doit être employée si $c = 0$). Dans ce cas, il est suffisant de choisir $P_{Ct01} = c$ pour traduire cette politique. La solution devient alors :

$$\left\{ \begin{array}{l} Ct01 = Dem \cdot \overline{Pc} \cdot \overline{Pm1} \cdot (Pm2 + c) \\ Ct02 = Dem \cdot \overline{Pc} \cdot \overline{Pm2} \cdot (Pm1 + \overline{c}) \\ Ct10 = \overline{Dem} + Pc + Pm1 \cdot Pm2 \\ Ct12 = Dem \cdot \overline{Pc} \cdot Pm1 \cdot \overline{Pm2} \\ Ct20 = \overline{Dem} + Pc + Pm1 \cdot Pm2 \\ Ct21 = Dem \cdot \overline{Pc} \cdot Pm2 \cdot \overline{Pm1} \end{array} \right.$$

VI. Conclusion

La méthode de synthèse présentée a été développée pour concevoir des contrôleurs logiques. Dans cette communication, nous avons montré comment cette approche peut être employée pour résoudre le cas spécifique où le savoir-faire du concepteur lui permet *a priori* d'établir la forme globale de la solution (par un modèle à états paramétrique). Le cas d'un RdPI a été choisi comme exemple d'illustration, mais la loi de commande pourrait être exprimée en employant n'importe quel autre modèle à états interprété (automate, Sequential Function Chart, statecharts, ...). En fait, tout problème de commande d'un système logique qui peut être exprimé sous la forme d'un système d'équations sur l'algèbre des fonctions booléennes peut être résolu en employant le théorème 2.

Néanmoins, deux types de problème restent aujourd'hui à traiter. Le premier est lié à la complexité du calcul pour résoudre le système d'équations. Nous avons développé un prototype sous Mathematica® qui nous permet de traiter des cas d'étude. Ces expériences montrent que la complexité du calcul symbolique exige le développement d'un outil spécifique plus performant. Nous devons également développer des techniques d'optimisation qui nous permettront de trouver la meilleure solution, modulo un critère d'optimisation.

La deuxième classe de problèmes est liée aux difficultés pour traduire les besoins fonctionnels, les règles de sûreté, les conditions de changement de mode, ... en équations algébriques. Nous avons déjà plusieurs contributions qui seront développées dans de futurs travaux.

Références

[1] PJG Ramadge et WM Wonham. The control of discrete event systems. *Proceedings of the IEEE*, 77(1) :81–98, 1989.

[2] J.-M. Roussel et A. Guia. Designing dependable logic controllers using the supervisory control theory. 16^e *IFAC World Congress, CDROM paper n°04427, Praha (CZ)*, Jul 2005.

[3] Y. Hietter, J.-M. Roussel, et J.-J. Lesage. Algebraic synthesis of dependable logic controllers. 17^e *IFAC World Congress*, 2008.

[4] R.P. Grimaldi. *Discrete and Combinatorial Mathematics : An Applied Introduction.* Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 5 edition, 2004. ISBN : 0-321-21103-0, 980 pages.

[5] C.E. Shannon. A symbolic analysis of relay and switching circuits. Master's thesis, Massachusetts Institute of Technology, Dept. of Electrical Engineering, 1940.

[6] B.A. Bernstein. Note on the Condition that a Boolean Equation Have a Unique Solution. *American Journal of Mathematics*, 54(2) :417–418, 1932.

[7] R.M. Toms. Systems of Boolean Equations. *The American Mathematical Monthly*, 73(1) :29–35, 1966.

[8] V.S. Levchenkov. Boolean equations with many unknowns. *Computational Mathematics and Modeling*, 11(2) :143–153, 2000.

[9] J. Machado, B. Denis, J.-J. Lesage, J.-M. Faure, et J. Ferreira Da Silva. Logic controllers dependability verification using a plant model. 3^e *IFAC Workshop on Discrete-Event System Design : DESDes'06, Rydzyna (Poland)*, pages 37–42, Sep 2006.