



HAL
open science

Génération d'une machine de Mealy à partir de spécifications algébriques à des fins de test de conformité

Anaïs Guignard, Jean-Marc Faure, Jean-Marc Roussel

► To cite this version:

Anaïs Guignard, Jean-Marc Faure, Jean-Marc Roussel. Génération d'une machine de Mealy à partir de spécifications algébriques à des fins de test de conformité. Conférence Internationale Francophone d'Automatique (CIFA2012), Jul 2012, France. pp.907-912. hal-00719363

HAL Id: hal-00719363

<https://hal.science/hal-00719363>

Submitted on 19 Jul 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Génération d'une machine de Mealy à partir de spécifications algébriques à des fins de test de conformité

Anaïs GUIGNARD, Jean-Marc ROUSSEL, Jean-Marc FAURE

LURPA, ENS Cachan
61, avenue du président Wilson, F-94230 Cachan
prenom.nom@lurpa.ens-cachan.fr

Résumé : Une méthode de construction d'une machine de Mealy équivalente à un système d'équations récurrentes qui décrit le comportement d'un système logique séquentiel est présentée dans ce papier. Cette construction repose sur la manipulation symbolique d'équations logiques et l'analyse de BDD (Binary Decision Diagram). Ce travail a été motivé par le besoin de prise en compte de spécifications hétérogènes, multi-langages, pour la génération de séquences de test de conformité.

Mot-clés : Systèmes logiques, approche algébrique, diagramme de décision binaire, machine de Mealy.

1. INTRODUCTION

Le test de conformité est une technique d'analyse qui a pour but de s'assurer que le comportement d'une implantation, vue comme une boîte noire, est conforme à celui décrit par sa spécification. L'exécution d'un test de conformité nécessite de construire au préalable une séquence de test, constituée d'une séquence d'entrées à appliquer à l'implantation et d'une séquence de sorties qui indique les réponses attendues. Dans le cas où implantation et spécification sont modélisables par des machines de Mealy, plusieurs méthodes sont proposées pour construire cette séquence dans Lee and Yannakakis (1996).

Les spécifications des systèmes de commande logiques industriels ne sont cependant pas décrites par des machines de Mealy mais sont exprimées dans des langages, souvent normalisés, adaptés aux ingénieurs automaticiens (par exemple IEC 60848 (2002), ISA-S5.2-1976 (R1992)). Il importe donc, pour tirer profit des résultats antérieurs, de développer des méthodes de traduction de ces modèles de spécification industriels. Les travaux présentés dans Provost et al. (2010) constituent un exemple d'une telle approche, dans le cas où les spécifications sont décrites uniquement en Grafset. Dans de nombreux cas cependant, les spécifications ne sont pas exprimées dans un seul langage, mais sont hétérogènes, certains langages étant par exemple plus adaptés à l'écriture des spécifications du fonctionnement normal et d'autres à celle de règles de sécurité.

Deux solutions sont alors envisageables :

- Traduction de chacun des modèles en machine de Mealy puis composition des machines obtenues,

- Traduction de chacun des modèles dans un format neutre puis conversion du modèle global, exprimé dans ce format, en une machine de Mealy.

Le travail présenté dans ce papier suppose que la deuxième solution est retenue, le format neutre étant une représentation algébrique. Cette représentation permet en effet, comme il a été montré dans Hietter et al. (2008), d'obtenir une description homogène et compacte à partir de fragments de spécifications dans des langages différents.

L'objectif de ce papier (Figure 1) est donc de proposer une méthode de construction d'une machine de Mealy équivalente à un jeu d'équations qui représente l'ensemble des spécifications servant de base à un test de conformité. Les points de départ et d'arrivée de notre travail, système d'équations récurrentes décrivant le comportement d'un système logique séquentiel (SLS) et machine de Mealy, sont présentés respectivement dans les sections 2 et 3. Les sections 4 et 5 expliquent comment l'ensemble des états atteignables de la machine ainsi que ses fonctions de transition et de sortie sont obtenus à partir du système d'équations initial ; ces contributions sont illustrées à travers un exemple simple. Quelques perspectives de travaux futurs sont enfin données en section 6.

2. REPRÉSENTATION ALGÈBRIQUE D'UN SYSTÈME LOGIQUE SÉQUENTIEL

2.1 Définition

La figure 2 présente une description algébrique d'un système logique séquentiel dont le comportement est défini par des équations récurrentes. Ce système comporte m entrées (u_1, \dots, u_m) , n variables d'état (x_1, \dots, x_n) et p sorties (y_1, \dots, y_p) .

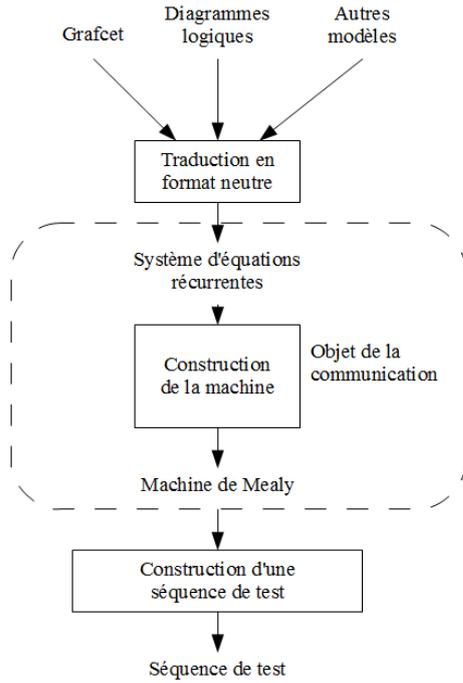


FIGURE 1. Contexte des travaux

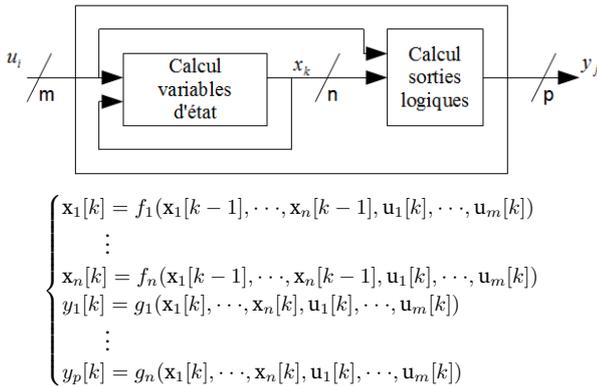


FIGURE 2. Représentation algébrique d'un SLS à l'aide d'équations récurrentes

Dans le cadre de ces travaux, deux hypothèses sont formulées :

- Le système d'équations est complet (toutes les variables sont définies) et cohérent.
- L'ensemble des sorties est un sous-ensemble de l'ensemble des variables d'état, ce qui se traduit notamment par le fait que les sorties sont indépendantes des variables d'entrée.

Cette dernière hypothèse peut se traduire mathématiquement par :

- H1 : $p \leq n$
- H2 : $\forall i \in [1, p], \exists j \in [1, n] / \forall k, y_i[k] = x_j[k]$

Elle n'est cependant pas limitative car il est possible de considérer les variables de sortie dépendantes des variables d'entrée comme des variables d'état. En effet, supposons qu'il existe une variable de sortie $y = g(x_1[k], \dots, x_n[k], u_1[k], \dots, u_m[k])$.

On définit x' tel que :

$$\begin{aligned} x' &= y = g(x_1[k], \dots, x_n[k], u_1[k], \dots, u_m[k]) \\ &= g(f_1(x_1[k-1], \dots, x_n[k-1], u_1[k], \dots, u_m[k]), \dots, \\ &\quad f_n(x_1[k-1], \dots, x_n[k-1], u_1[k], \dots, u_m[k]), \\ &\quad u_1[k], \dots, u_m[k]) \\ &= f_{n+1}(x_1[k-1], \dots, x_n[k-1], u_1[k], \dots, u_m[k]) \end{aligned}$$

Il suffit alors de redéfinir $\mathcal{X} \leftarrow \mathcal{X} \cup x'$.

Afin d'alléger les notations, nous noterons dans la suite de ce papier :

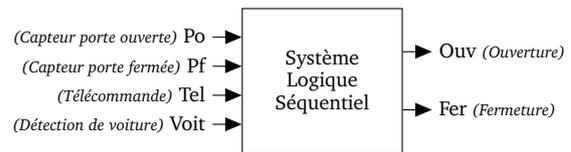
- ${}_p nom$: la valeur $nom[k-1]$ de la variable 'nom' à l'instant $(k-1)$,
- nom : la valeur $nom[k]$ de la variable 'nom' à l'instant k ,
- \mathcal{U} : le vecteur des m variables d'entrée (u_1, \dots, u_m) ,
- \mathcal{X} : le vecteur des n variables d'état (x_1, \dots, x_n) ,
- ${}_p \mathcal{X}$: le vecteur des n variables d'état $({}_p x_1, \dots, {}_p x_n)$.

En accord avec ces notations et ces hypothèses, le système d'équations récurrentes décrivant le comportement d'un SLS devient :

$$\begin{cases} \mathbf{x}_1 = f_1({}_p \mathcal{X}, \mathcal{U}) \\ \vdots \\ \mathbf{x}_n = f_n({}_p \mathcal{X}, \mathcal{U}) \end{cases} \quad (1)$$

2.2 Illustration

Les différentes étapes de la méthode présentée vont être illustrées à l'aide d'un exemple (Figure 3) décrivant la commande d'une porte de garage (Roussel and Faure (2006)). Ce système comporte quatre entrées logiques $\mathcal{U} = \{Po, Pf, Tel, Voit\}$ et deux sorties logiques prenant la valeur des deux variables d'état $\mathcal{X} = \{Ouv, Fer\}$ déterminées à l'aide d'un système d'équations récurrentes. A l'instant initial, les variables d'état sont toutes deux à l'état FAUX (noté 0).



$$\begin{cases} Ouv[k] = \overline{Po[k]} \cdot (Tel[k] + \overline{Pf[k]} \cdot Voit[k]) + Ouv[k-1] \\ Fer[k] = Pf[k] \cdot Tel[k] \cdot Voit[k] \cdot (Po[k] + Ouv[k-1]) \\ Ouv[0] = 0 \\ Fer[0] = 0 \end{cases}$$

FIGURE 3. Exemple de représentation algébrique d'un SLS

3. POSITION DU PROBLEME

Une machine de Mealy M est un 6-uplet $(I_M, O_M, S_M, s_{init}, \delta_M, \lambda_M)$ avec :

- I_M : Alphabet d'entrée.
- O_M : Alphabet de sortie.
- S_M : Ensemble des états.
- s_{init} : Etat initial.
- δ_M : $S_M \times I_M \rightarrow S_M$: Fonction de transition.
- λ_M : $S_M \times I_M \rightarrow O_M$: Fonction de sortie.

Notre approche consiste en la détermination de l'ensemble des éléments définissant la machine de Mealy à partir

de la représentation algébrique d'un SLS comportant m variables d'entrée, n variables d'état et p variables de sortie. On peut alors directement déterminer :

- L'alphabet d'entrée I_M dont chaque lettre est un min-terme ¹ des variables d'entrée du système,
- l'alphabet de sortie O_M dont chaque lettre est un min-terme des variables de sortie du SLS,
- l'ensemble des au plus 2^n états S_M , chacun défini par un minterme des variables d'état du SLS,
- l'unique état initial s_{init} défini par le minterme des variables d'état initiales du SLS,

Dans le cas de l'exemple Figure 3, on obtient :

- $I_M = \{\overline{Po} \cdot \overline{Pf} \cdot \overline{Tel} \cdot \overline{Voit}, \dots, Po \cdot Pf \cdot Tel \cdot Voit\}$ avec $Card(I_M) = 2^4 = 16$.
- $O_M = \{\overline{Ouw} \cdot \overline{Fer}, Ouw \cdot \overline{Fer}, \overline{Ouw} \cdot Fer, Ouw \cdot Fer\}$.
- $S_M \subset \{\overline{Ouw} \cdot \overline{Fer}, Ouw \cdot \overline{Fer}, \overline{Ouw} \cdot Fer, Ouw \cdot Fer\}$.
- s_{init} défini par $E_{init} = \overline{Ouw} \cdot \overline{Fer}$.

Il reste donc à caractériser la fonction de transition ainsi que la fonction de sortie de la machine de Mealy. Comme la méthode de génération de ces fonctions se fait de manière purement symbolique, une fonction Δ_M , construite comme une agrégation des transitions définies par δ_M , est introduite. Ainsi, la méthode développée propose la génération d'une fonction de transition $\Delta_M : (S_M \times Exp_M \rightarrow S_M)$ avec Exp_M défini comme la disjonction des lettres de l'alphabet d'entrée associées à chaque transition dont la fonction de sortie renvoie une lettre commune :

$$\forall (s_1, s_2) \in S_M \times S_M, \forall o \in O_M, Exp_M = \left\{ \sum (i) \mid i \in I_M \text{ et } \delta(s_1, i) = s_2 \text{ et } \lambda(s_1, i) = o \right\}$$

Enfin, suite à l'hypothèse H2, la valeur de la fonction de sortie λ_M associée à une transition ne dépend que de l'état aval de la transition :

$$\forall (s_1, s'_1, s_2) \in S_M \times S_M \times S_M, \forall (i_1, i_2) \in I_M \times I_M, \begin{cases} \delta(s_1, i_1) = s_2 \\ \delta(s'_1, i_2) = s_2 \end{cases} \Rightarrow \lambda_M(s_1, i_1) = \lambda_M(s'_1, i_2)$$

En accord avec ces définitions, la représentation graphique d'une machine de Mealy peut être simplifiée par agrégation des arcs ayant le même état amont et le même état aval comme illustré figure 4.

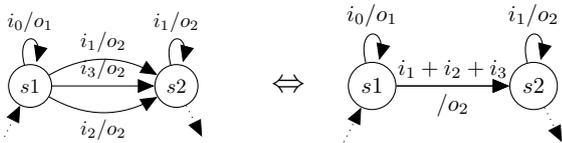


FIGURE 4. Agrégation des arcs de la machine de Mealy

La méthode de calcul de l'ensemble d'états S_M et de la fonction de transition Δ_M est proposée dans la section suivante. La connaissance de ces deux éléments permet de générer la fonction de sortie λ_M , comme indiqué en section 5.

4. DÉTERMINATION DE S_M ET DE Δ_M

Cette section s'intéresse à l'obtention de la fonction de transition ainsi que de l'ensemble d'états de la machine

1. Un minterme est une expression logique des n variables utilisant uniquement l'opérateur logique ET (\cdot) et l'opérateur complément ($\bar{}$)

de Mealy à partir du système d'équations récurrentes (1). Trois étapes sont nécessaires :

- Calcul de l'expression donnant toutes les évolutions potentielles du SLS.
- Détermination des états atteignables depuis l'état initial par algorithme de recherche de point fixe.
- Calcul de l'expression donnant les évolutions entre les seuls états atteignables.

L'étape 1 revient à déterminer l'ensemble des évolutions possibles des variables d'état sans considération de l'état initial puis la recherche des états atteignables, lors de l'étape 2, permet de déterminer l'ensemble S_M des états de la machine de Mealy. Enfin, l'étape 3 propose une opération de restriction de l'expression caractérisant les évolutions du SLS aux seules évolutions entre les états atteignables et permet d'obtenir Δ_M .

4.1 Expression donnant les évolutions potentielles du SLS

Les notations suivantes doivent être introduites pour le calcul de cette expression, notée E_G .

Notations Soient une variable logique v et une fonction logique $f(p, \mathcal{X}, \mathcal{U})$, soit le booléen $a \in \{0, 1\}$. On notera :

$$\begin{aligned} - v^0 &= \bar{v} & - v^1 &= v \\ - f^0(p, \mathcal{X}, \mathcal{U}) &= \overline{f(p, \mathcal{X}, \mathcal{U})} & - f^1(p, \mathcal{X}, \mathcal{U}) &= f(p, \mathcal{X}, \mathcal{U}) \\ - f^a(p, \mathcal{X}, \mathcal{U}) &= f(p, \mathcal{X}, \mathcal{U}) \cdot a + \overline{f(p, \mathcal{X}, \mathcal{U})} \cdot \bar{a} \end{aligned}$$

Cette notation sera étendue au vecteur $V = (v_1, \dots, v_k)$. Le terme V^A , où $A = (a_1, \dots, a_k) \in \{0, 1\}^k$ est un vecteur de booléens de dimension k , correspond alors au minterme :

$$V^A = \prod_{i=1}^k v_i^{a_i} = v_1^{a_1} \cdot \dots \cdot v_k^{a_k}$$

D'autre part, selon Brown (2003), chaque fonction $f_i(p, \mathcal{X}, \mathcal{U})$ de (1) peut être exprimée sous une forme canonique, obtenue à l'aide d'une décomposition de Shannon appliquée à chaque terme de p, \mathcal{X} . Le vecteur de booléens $T \in \{0, 1\}^n$ est alors introduit afin d'autoriser une forme compacte de cette décomposition. La notation p, \mathcal{X}^T représente donc un minterme défini sur l'ensemble des variables d'état, considérées à l'instant $(k-1)$. Il vient alors pour chaque fonction $f_i(p, \mathcal{X}, \mathcal{U})$:

$$f_i(p, \mathcal{X}, \mathcal{U}) = \sum_{T \in \{0, 1\}^n} f_i(T, \mathcal{U}) \cdot p, \mathcal{X}^T \quad (2)$$

Chaque minterme p, \mathcal{X}^T satisfait :

$$\forall (T_1, T_2) \in \{0, 1\}^n, p, \mathcal{X}^{T_1} \cdot p, \mathcal{X}^{T_2} = 0 \text{ si } T_1 \neq T_2 \quad (3)$$

La forme canonique (2) vérifie les propriétés suivantes :

$$\begin{aligned} \overline{f_i(p, \mathcal{X}, \mathcal{U})} &= \sum_{T \in \{0, 1\}^n} \overline{f_i(T, \mathcal{U})} \cdot p, \mathcal{X}^T \\ f(p, \mathcal{X}, \mathcal{U}) + g(p, \mathcal{X}, \mathcal{U}) &= \sum_{T \in \{0, 1\}^n} (f(T, \mathcal{U}) + g(T, \mathcal{U})) \cdot p, \mathcal{X}^T \\ f(p, \mathcal{X}, \mathcal{U}) \cdot g(p, \mathcal{X}, \mathcal{U}) &= \sum_{T \in \{0, 1\}^n} (f(T, \mathcal{U}) \cdot g(T, \mathcal{U})) \cdot p, \mathcal{X}^T \end{aligned}$$

Première forme algébrique La détermination de E_G repose sur les propriétés suivantes :

- chaque état potentiel (atteignable ou non) du SLS correspond à un minterme défini sur les variables d'état du SLS,
- la disjonction de ces mintermes est toujours vraie (il existe toujours un état actif).

Il vient donc :

$$E_G = \sum_{H \in \{0,1\}^n} \mathcal{X}^H = 1 \quad (4)$$

En utilisant la propriété d'idempotence puis en introduisant les expressions des variables d'état selon (1), il vient :

$$E_G = \sum_{H \in \{0,1\}^n} \mathcal{X}^H \cdot \mathcal{X}^H \quad (5)$$

$$= \sum_{H \in \{0,1\}^n} \mathcal{X}^H \cdot \left(\prod_{i=1}^n x_i^{h_i} \right) \quad (6)$$

$$= \sum_{H \in \{0,1\}^n} \mathcal{X}^H \cdot \left(\prod_{i=1}^n f_i^{h_i}({}_p\mathcal{X}, \mathcal{U}) \right) \quad (7)$$

Une application de la décomposition de Shannon à chacune des fonctions $f_i({}_p\mathcal{X}, \mathcal{U})$ permet alors d'extraire les mintermes portant sur l'état précédent des variables d'état ${}_p\mathcal{X}$.

$$E_G = \sum_{H \in \{0,1\}^n} \left(\mathcal{X}^H \cdot \sum_{T \in \{0,1\}^n} \prod_{i=1}^n f_i^{h_i}(T, \mathcal{U}) \cdot {}_p\mathcal{X}^T \right) \quad (8)$$

On obtient finalement :

$$E_G = \sum_{H \in \{0,1\}^n} \sum_{T \in \{0,1\}^n} \left({}_p\mathcal{X}^T \cdot \mathcal{X}^H \cdot \prod_{i=1}^n f_i^{h_i}(T, \mathcal{U}) \right) \quad (9)$$

L'interprétation de cette expression (équation (9)) en termes d'évolutions entre états est donnée ci-dessous.

Interprétation Comme indiqué précédemment, chacun des états correspond à un minterme défini sur les variables d'état du SLS. La forme générale d'un de ces mintermes est donnée par les équations (10) à (12).

$$\forall H \in \{0,1\}^n, \quad \mathcal{X}^H = \prod_{i=1}^n x_i^{h_i} = \prod_{i=1}^n f_i^{h_i}({}_p\mathcal{X}, \mathcal{U}) \quad (10)$$

$$= \prod_{i=1}^n \left(\sum_{T \in \{0,1\}^n} f_i^{h_i}(T, \mathcal{U}) \cdot {}_p\mathcal{X}^T \right) \quad (11)$$

$$\mathcal{X}^H = \sum_{T \in \{0,1\}^n} \left(\prod_{i=1}^n (f_i^{h_i}(T, \mathcal{U})) \cdot {}_p\mathcal{X}^T \right) \quad (12)$$

Selon l'expression (12), l'état caractérisé par le minterme \mathcal{X}^H est actif si et seulement si :

- l'état actif précédent est l'état caractérisé par le minterme ${}_p\mathcal{X}^T$,
- les valeurs des variables d'entrée satisfont la relation $\prod_{i=1}^n f_i^{h_i}(T, \mathcal{U}) = 1$.

Ces conditions sont représentées graphiquement par la Figure 5.

L'expression E_G est une disjonction (ou somme logique) de mintermes \mathcal{X}^H . Elle exprime donc, sous forme symbolique, un ensemble de $(2^n \times 2^n)$ évolutions entre états.

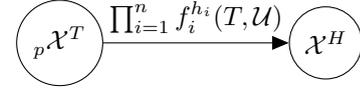


FIGURE 5. Forme générique d'une évolution entre deux états d'un SLS

Seconde forme algébrique Il apparaît que l'expression (9) est de complexité $2^{2 \cdot n}$ pour un SLS comportant n variables d'état. Le processus d'automatisation de la méthode développée privilégie cependant l'utilisation d'une seconde forme de cette expression, de complexité linéaire :

$$E_G = \prod_{i=1}^n \left((x_i \cdot f_i({}_p\mathcal{X}, \mathcal{U})) + (\overline{x_i} \cdot \overline{f_i({}_p\mathcal{X}, \mathcal{U})}) \right) \quad (13)$$

Cette expression peut alors être directement calculée comme le produit de n expressions, chacune issue d'une des n équations récurrentes. L'égalité entre les expressions (9) et (13) est montrée ci-dessous.

Démonstration. Pour chaque équation de (1), on peut écrire :

$$(x_i \cdot f_i({}_p\mathcal{X}, \mathcal{U})) + (\overline{x_i} \cdot \overline{f_i({}_p\mathcal{X}, \mathcal{U})}) = 1 \quad (14)$$

Car si v_1 et v_2 sont deux variables booléennes,

$$v_1 = v_2 \Rightarrow v_1 \cdot v_2 + \overline{v_1} \cdot \overline{v_2} = 1$$

D'où :

$$E_G = \prod_{i=1}^n \left((x_i \cdot f_i({}_p\mathcal{X}, \mathcal{U})) + (\overline{x_i} \cdot \overline{f_i({}_p\mathcal{X}, \mathcal{U})}) \right) = 1 \quad (15)$$

D'après (4) et (15) les deux formes (9) et (13) sont égales et utilisent les mêmes variables \mathcal{X} , ${}_p\mathcal{X}$ et \mathcal{U} ; on peut donc conclure qu'elles sont équivalentes. \square

Exemple Dans le cas de l'exemple considéré, à partir du système d'équations récurrentes donné Figure 3, on obtient l'expression décrivant toutes les évolutions potentielles du SLS, sans considération de l'état initial :

$$E_G = \left[(Ouw \cdot (\overline{Po} \cdot (Tel + (\overline{Pf} \cdot Voit) + {}_pOuw))) + (\overline{Ouw} \cdot (\overline{Po} \cdot (Tel + (\overline{Pf} \cdot Voit) + {}_pOuw))) \right] \cdot \left[(Fer \cdot (\overline{Pf} \cdot \overline{Tel} \cdot \overline{Voit} \cdot (Po + \overline{{}_pOuw}))) + (\overline{Fer} \cdot \overline{Pf} \cdot \overline{Tel} \cdot \overline{Voit} \cdot (Po + \overline{{}_pOuw})) \right] \quad (16)$$

On observe, par exemple, qu'en l'absence d'activation de capteur ($Po = Pf = Tel = Voit = 0$), l'expression devient :

$$E_G = Ouw \cdot \overline{Fer} \cdot \overline{{}_pOuw} + \overline{Ouw} \cdot Fer \cdot \overline{{}_pOuw}$$

Cette expression signifie que dans le cas considéré (porte en position intermédiaire car $Po = Pf = 0$) :

- la commande à l’ouverture de la porte reste active si elle l’était précédemment,
- si elle ne l’était pas, c’est que la commande à la fermeture est active.

La porte ne reste donc pas en position intermédiaire pour ces valeurs des capteurs, ce qui est normal.

4.2 Détermination des états atteignables depuis l’état initial

Le calcul de l’expression donnant l’ensemble des états atteignables depuis un état initial fixé, notée E_R , peut être réalisé à l’aide d’un algorithme de recherche de point fixe (algorithme 1). Pour cette opération, la seule connaissance de l’existence d’une transition entre un état s_i et un état s_j donnés, et non de la lettre de l’alphabet d’entrée qui cause cette transition, est nécessaire.

L’algorithme 1 va donc être appliqué sur une expression simplifiée E_{GDev} issue de E_G en ignorant toutes les variables d’entrée \mathcal{U} . Ceci est réalisé à l’aide d’une fonction de développement appliquée sur les variables d’entrée appelée Dev :

$$E_{GDev} = Dev(E_G, \mathcal{U}) \quad (17)$$

Cette fonction est ainsi définie : soient (a,b,c) trois variables logiques et $E_1(a, b, c)$ une expression sur ces variables, le développement de la variable ‘a’ dans l’expression $E_1(a, b, c)$ donne $E_2(b, c)$ où :

$$\begin{aligned} E_2(b, c) &= Dev(E_1(a, b, c), a) \\ &= E_1(a, b, c) |_{a \leftarrow 1} + E_1(a, b, c) |_{a \leftarrow 0} \end{aligned}$$

Dans le cadre de l’exemple, la forme développée de l’équation (16) est :

$$\begin{aligned} E_{GDev} &= Dev(E_G, \{Po, Pf, Tel, Voit\}) \\ &= \overline{Ouv} + \overline{Fer} \end{aligned}$$

Cette expression, ainsi que l’expression de l’état initial du système $E_{Init} = \overline{Ouv} \cdot \overline{Fer}$ sont les données d’entrée de l’algorithme 1.

Algorithme 1 Calcul symbolique des états atteignables d’une machine de Mealy depuis son état initial

Require: E_{Init}, E_{GDev}
{Initialisation}
 $E_R = E_{Init}$
{Recherche de point fixe}
repeat
 $E_{Amont} = E_R$
 $E_{Trans} = E_{Amont} \cdot E_{GDev}$
 $E_{Aval} = Dev(E_{Trans}, \mathcal{U})$
 $E_R = E_R + Renommer(E_{Aval}, [{}_p x_i \leftarrow x_i])$
until ($E_R == E_{Amont}$)
return E_R

Appliqué à l’exemple, l’algorithme converge dès la seconde itération. On obtient alors l’expression décrivant l’ensemble des états atteignables depuis l’état initial :

$$E_R = \overline{{}_p Ov} + \overline{{}_p Fer}$$

L’ensemble S_M est donc $\{00, 01, 10\}$.

Il convient de remarquer que l’état où la porte est simultanément commandée en ouverture et en fermeture n’est

pas atteignable, ce qui est en accord avec le fonctionnement souhaité.

4.3 Expression donnant les évolutions entre les seuls états atteignables

Cette expression, notée E_M , décrit de manière algébrique l’ensemble des évolutions entre les seuls états atteignables. Elle est donc obtenue simplement par produit logique entre les expressions E_G et E_R , soit :

$$E_M = E_G \cdot E_R \quad (18)$$

Comme toute expression logique, E_M peut être décrite par un BDD (Binary Decision Diagram, Bryant (1992)) ; l’analyse du BDD associé à E_M permet d’obtenir les fonctions de transition Δ_M et de sortie λ_M . Il n’est cependant pas possible de détailler dans ce papier l’algorithme d’analyse développé lors de ce travail pour des raisons de place ; seule une illustration sur la base de l’exemple de la figure 3 est présentée dans ce qui suit.

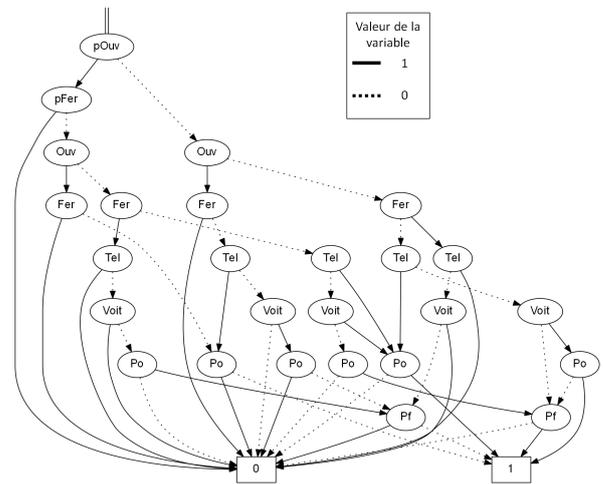


FIGURE 6. Représentation de E_M par un BDD

Le BDD décrivant en extension l’expression (18) pour cet exemple est donné à la figure 6, les tests sur les variables étant ordonnés de la manière suivante : tests sur les éléments de ${}_p \mathcal{X}$ (état antérieur), tests sur les éléments de \mathcal{X} (état courant), tests sur les éléments de \mathcal{U} (variables d’entrée). Les valeurs de Δ_M pour chaque état atteignable et chaque disjonction des lettres de l’alphabet d’entrée s’obtiennent en recherchant l’ensemble des chemins partant de la racine du BDD et se terminant à la feuille portant la valeur 1. Avec l’ordre des tests sur les variables qui a été retenu, les nœuds traversés par chacun de ces chemins fournissent en effet :

- une combinaison booléenne des variables d’état qui définit un ou plusieurs états amont de transition(s) de la machine de Mealy,
- une combinaison booléenne des variables d’état qui définit un ou plusieurs états aval de transition(s) de la machine de Mealy,
- une condition sur les variables d’entrée qui doit être vraie pour que la transition de l’état amont vers l’état aval se produise.

Dans le cas de la figure 6, si l'on parcourt par exemple le chemin passant par les noeuds \overline{pOuw} , \overline{Ouw} , \overline{Fer} , \overline{Tel} et Po , on obtient les définitions de deux transitions. Le terme \overline{pOuw} pouvant en effet s'écrire $\overline{pOuw} \cdot (\overline{pFer} + pFer)$, le chemin considéré indique que la machine de Mealy comporte deux transitions ainsi définies :

- les états amont de ces transitions sont $\overline{pOuw} \cdot \overline{pFer}$ codé 01 et $\overline{pOuw} \cdot pFer$ codé 00,
- ces deux transitions ont le même état aval $\overline{Ouw} \cdot \overline{Fer}$ (ou 00),
- chacune de ces transitions est franchie si l'expression $\overline{Tel} \cdot Po$ est vérifiée, l'état amont étant bien entendu actif.

L'obtention des expressions complètes des conditions de transition requiert d'analyser la totalité des chemins passant par les noeuds \overline{pOuw} , \overline{Ouw} , \overline{Fer} . Cette analyse montre que les deux transitions considérées sont conditionnées par la même expression :

$$Exp_M = Po \cdot (Tel + Voit) + Pf \cdot \overline{Tel}$$

La dernière information à obtenir pour connaître complètement la machine de Mealy est la fonction de sortie λ_M .

5. DÉTERMINATION DE λ_M

Chaque état de la machine de Mealy correspond à une combinaison des variables d'état du SLS. Etant donné l'hypothèse H2 (les variables de sortie sont un sous-ensemble des variables d'état), la valeur de la fonction de sortie λ_M pour chaque transition est celle des variables d'état codant l'état aval de cette transition, pour assurer la cohérence. L'ensemble des valeurs de λ_M s'obtient donc lors de l'analyse des chemins du BDD qui décrit l'expression de E_M .

Par exemple, pour les deux transitions considérées en 4.3 qui conduisent toutes deux à l'état $\overline{Ouw} \cdot \overline{Fer}$ (00), la valeur de λ_M est également $\overline{Ouw} \cdot \overline{Fer}$. Cette valeur est d'ailleurs commune à toutes les transitions conduisant à cet état. La figure 7 donne la machine de Mealy équivalente au système de la Figure 3. Dans cette figure, la notation E_{ij-kl} est introduite pour désigner l'étiquette de la transition qui relie l'état ij à l'état kl ; la première partie de cette expression (partie avant le symbole /) indique la valeur de Δ_M pour la transition, la seconde partie la valeur de λ_M .

6. CONCLUSION ET PERSPECTIVES

Ce papier a proposé une méthode de construction d'une machine de Mealy équivalente à un système d'équations récurrentes qui décrit le comportement d'un système logique séquentiel; cette méthode a été totalement automatisée. Lorsque le système d'équations est obtenu à partir de spécifications dans différents langages, cette contribution ouvre la voie à la génération de séquences de test de conformité à partir de spécifications industrielles hétérogènes.

Afin d'accroître les possibilités de ce travail, la levée de l'hypothèse imposant que les sorties logiques ne dépendent que des variables d'état doit être envisagée dans un futur proche. Ceci requiert une nouvelle définition des états de la machine de Mealy, par exemple en fonction des

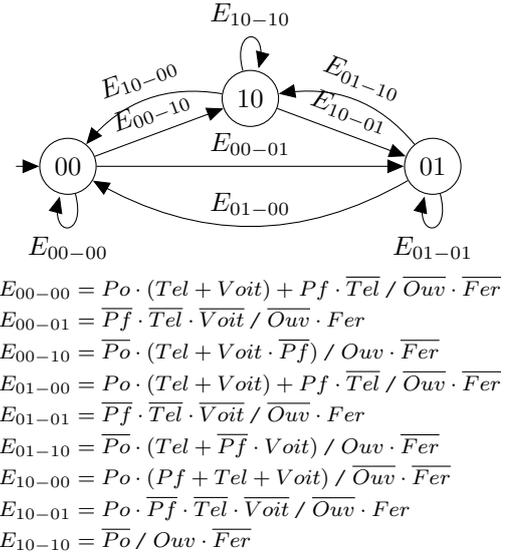


FIGURE 7. Machine de Mealy générée pour l'exemple Figure 3

variables d'état et de sortie, et pourrait permettre la présence dans les spécifications de grafjets comportant des actions conditionnelles. Une seconde piste de recherche concerne la comparaison de l'approche présentée, basée sur la traduction des fragments de spécification en un format neutre, à celle basée sur la composition des machines de Mealy déduites de ces fragments.

Remerciements

Ce travail a été réalisé dans le cadre du projet TESTEC (ANR TLOG 07-022).

RÉFÉRENCES

- Brown, F.M. (2003). *Boolean Reasoning : The Logic of Boolean Equations*. Dover Publications.
- Bryant, R.E. (1992). Symbolic boolean manipulation with ordered binary-decision diagrams. *ACM Comput. Surv.*, 24, pp. 293–318.
- Hietter, Y., Roussel, J.M., and Lesage, J.J. (2008). Algebraic synthesis of dependable logic controllers. In *Proceedings of 17th IFAC World Congress, 2008*, pp. 4132–4137. Seoul, Corée, République De.
- IEC 60848 (2002). *GRAFCET Specification language for sequential function charts (2nd ed.)*. International Electrotechnical Commission.
- ISA-S5.2-1976(R1992). *Binary Logic Diagrams for Process Operations*. International Society of Automation.
- Lee, D. and Yannakakis, M. (1996). Principles and methods of testing finite state machines-a survey. *Proceedings of the IEEE*, 84(8), pp. 1090–1123.
- Provost, J., Roussel, J.M., and Faure, J.M. (2010). Translating Grafjet specifications into Mealy machines for conformance test purposes. *Control Engineering Practice*, 19(9), pp. 947–957.
- Roussel, J.M. and Faure, J.M. (2006). Designing dependable logic controllers using algebraic specifications. *Control Engineering Practice*, 14(10), pp. 1143–1155.