



# IMHOTEP

AFRICAN JOURNAL OF PURE AND APPLIED MATHEMATICS

## Imhotep Mathematical Proceedings Volume 2, Numéro 1, (2015), pp. 76 – 87.

Synthèse de la commande supervisée d'un système totalement modélisé par Grafcet basée sur l'algorithme de Kumar

Wayang Senguel  
Bitjoka Laurent  
Gonza Mohaman  
Donfack B. Omer  
Ngako B. Axel

Equipe Automatique et  
Traitement du Signal (ATS),  
Laboratoire Energie, Signal Image  
et Automatique (ESIA),  
ENSAI-Université de Ngaoundéré,  
BP 455, Ngaoundéré.


senguel.wayang@univ-ndere.cm  
lbitjoka@univ-ndere.cm  
mohaman.gonza@univ-ndere.cm  
omer.donfack@univ-ndere.cm  
brice.ngako@univ-ndere.cm

### Abstract

Ce papier traite de la synthèse de supervision pour des systèmes modélisé par Grafcet. il donne une méthode formelle de composition des Grafcets à l'instar des automates à états finis qui sont des outils de base de la théorie de supervision. Nous avons obtenu le Grafcet superviseur mais la simulation de sa dynamique a révélé certaines incohérences car plusieurs étapes de ce Grafcet s'activaient en même temps. Il fallait donc corriger ce dernier en forçant le Grafcet superviseur à n'avoir qu'une seule étape active. Notre méthode a l'avantage de pouvoir exploiter certaines propriétés comme les divergences en **ET** divergence en ET qui n'apparaît dans la théorie des automates finis. une application de notre approche sur un système manufacturier comparativement à la théorie de supervision basée sur les automates montre que le résultat obtenu est plus permissif.

Proceedings of the 4<sup>th</sup> annual workshop on  
CRyptography, Algebra and Geometry  
(CRAG-4), 21- 25 July 2014, University of  
Dschang, Dschang, Cameroon.

<http://imhotep-journal.org/index.php/imhotep/>

Imhotep Mathematical Proceedings 

# Synthèse de la commande supervisée d'un système totalement modélisé par Grafcet basée sur l'algorithme de Kumar

Wayang Senguel, Bitjoka Laurent, Gonza Mohaman, Donfack B. Omer  
and Ngako B. Axel

**Résumé.** Ce papier traite de la synthèse de supervision pour des systèmes modélisé par Grafcet. il donne une méthode formelle de composition des Grafcets à l' instar des automates à états finis qui sont des outils de base de la théorie de supervision. Nous avons obtenu le Grafcet superviseur mais la simulation de sa dynamique à révélé certaines incohérences car plusieurs étapes de ce Grafcet s'activaient en même temps. Il fallait donc corriger ce dernier en forçant le Grafcet superviseur à n'avoir qu'une seule étape active. Notre méthode a l'avantage de pouvoir exploiter certaines propriétés comme les divergences en **ET** divergence en ET qui n'apparaît dans la théorie des automates finis. une application de notre approche sur un système manufacturier comparativement à la théorie de supervision basée sur les automates montre que le résultat obtenu est plus permissif.

**Keywords.** Théorie de supervision, Automates finis, Grafcet.

## I. Introduction

Dans le concept de la théorie de supervision (Ramadge et Wonham, 1987) les modèles automate du procédé et des spécifications sont donnés, le superviseur obtenu après le processus de synthèse n'a pour seuls rôles d'autoriser ou d'empêcher les actions entreprises par la commande. Les travaux trouvés dans la littérature (Ndjab, 1999) (Tajer, 2005) (Philipot et *al.*, 2003) (Marangé et *al.*, 2008) font une synthèse hybride. Le système à commander modélisé par Grafcet enfin de réduire le nombre d'états de manière structurel, puis transformer en automate afin utiliser les algorithmes de synthèse applicable sur les automates finis. Il est par la suite retransformé en Grafcet pour les besoins d'implantation. Notre démarche est un rapprochement à la supervision à base des automates finis. Ce rapprochement consiste à obtenir le langage suprême contrôlable. Plusieurs travaux ont développé des algorithmes à cet effet (Ramadge et Wonham, 1989; Brandin, 1989; Kumar, 1991). Nous avons choisi l'usage de l'algorithme de Kumar que le lecteur trouvera dans (Kumar, 1991).

---

Communication présentée au 4<sup>ème</sup> atelier annuel sur la CRyptographie, Algèbre et Géométrie (CRAG-4), 21 - 25 Juillet 2014, Université de Dschang, Dschang, Cameroun / Paper presented at the 4<sup>th</sup> annual workshop on CRyptography, Algebra and Geometry (CRAG-4), 21- 25 July 2014, University of Dschang, Dschang, Cameroon.

Ainsi, il est question pour nous de modéliser le système et son cahier de charges (expression propre au Grafcet) et d'appliquer directement l'algorithme de Kumar. Cependant la composition des Grafcets s'effectue selon la technique que nous développons et qui a l'avantage d'exploiter les points forts des règles d'évolution des Grafcet sans perte d'information ou de structure de départ des modèles. Dans ce papier, un *état* est un ensemble d'étapes actives des modèles de départ tandis que la *spécification* représente le cahier de charges. Le respect des spécifications revient donc à déterminer des états interdits qui doivent devenir inaccessibles par une action du superviseur en supprimant les états indésirables du Grafcet superviseur. L'algorithme de Kumar (Kumar, 1991) permet d'avoir un comportement commun à travers la composition des modèles de départ. Le graphe de comportement commun ressort tous les états atteignables par le procédé. Les états interdits sont alors supprimés. Notre objectif est de faire la synthèse d'un contrôleur basé uniquement sur le Grafcet.

## II. Etapes de synthèse d'une supervision d'un système modélisé par Grafcet

Considérons un système manufacturier qui est un exemple classique de la théorie de supervision, composé de deux machines identiques :  $M_1$  et  $M_2$ , et un stock entre les deux machines. Conformément à la figure 2, les deux machines travaillent de façon indépendante, puisent des pièces brutes en amont et rejettent des pièces usinées en aval.

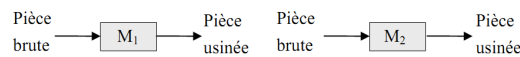


FIGURE 1. Système manufacturier

### Spécification :

Le fonctionnement de notre système manufacturier doit respecter la présence d'un stock de capacité limitée à 1, situé entre les deux machines. Nous supposons donc à présent que les machines travaillent en série.



FIGURE 2. Système manufacturier sous la contrainte de stock

Sagissant de notre méthode, la figure 1 représente les étapes nécessaires pour synthétiser le superviseur.

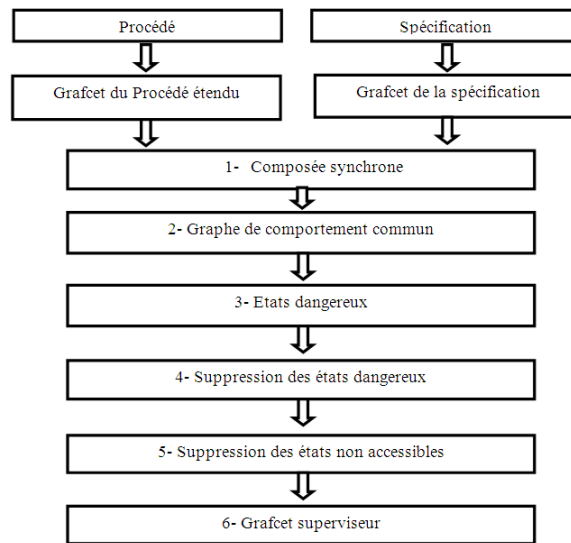


FIGURE 3. Les étapes nécessaires pour synthétiser le Grafcet superviseur.

Le point de départ consiste à modéliser le procédé et les spécifications par des Grafquets. Le comportement désiré en boucle fermée est obtenu par la composition des deux Grafquets (étape 1 dans la figure 1). La construction du Grafcet de comportement commun (étape 2) à partir du Grafcet synchronisé permettra de repérer les états dangereux (étape 3) la suppression de ces derniers (étape 4) génère quelque fois des états inaccessibles. Si les états inaccessibles existent, l'on les supprime (étape 5). On obtient alors un Grafcet superviseur contrôlable.

### III. Méthode de composition synchrone des modèles Grafquets

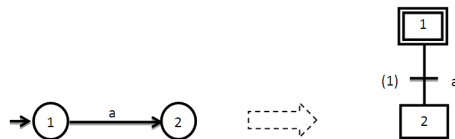
Afin de faire la composée des modèles du procédé et celui des spécifications, il est nécessaire de formaliser la composition de ces derniers.

En se basant sur le comportement d'un Grafcet et en s'inspirant des règles de composition des automates finis. Le comportement d'un Grafcet à travers ses règles d'évolution et le formalisme de composition synchrone lorsqu'ils sont exploités permettent de dépasser la limite de non simultanéité d'apparition des événements relevée dans le cas des automates finis. En effet, deux événements peuvent apparaître en même temps. Nous avons d'abord fait certaines équivalences des deux langages.

#### III.1. Equivalences entre le Grafcet et l'automate fini

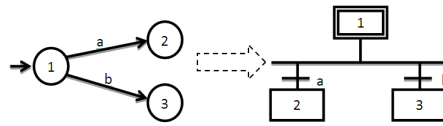
##### Equivalence 1 :

L'état initial d'un automate est identifié par une flèche-source, tandis que celui du Grafcet est identifié par un double rectangle. L'évolution d'un état à un autre par une transition simple dans un automate, est semblable à celui du Grafcet.



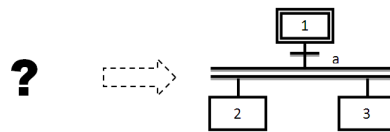
##### Equivalence 2 :

Une évolution donnant un choix dans un automate est une divergence en **OU** dans un Grafcet.



**Equivalence 3 :**

Il n'existe pas de divergence en **ET** dans un automate à états finis, pourtant cette situation est très présente en industrie.



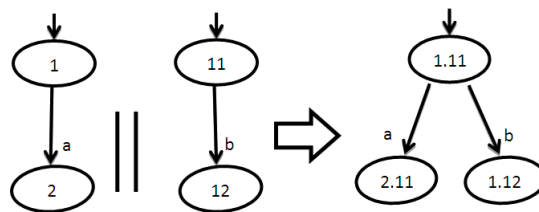
**III.2. Lois de composition des Grafcets**

A présent nous allons voir comment l'on peut composer les Grafcet afin d'avoir un comportement commun. Nous montrons d'abord comment c'est fait avec les automates afin de bien voir notre apport. Les chiffres apparaissant dans les rectangles désignent les étapes actives à cet état.

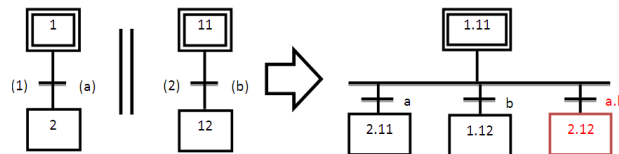
**Proposition 1 :** Composition de deux événements (réceptivités) distincts

- Avec les automates

La composition (ou produit synchrone) parallèle est noté par deux barres parallèles verticales. Dans la composition parallèle, on ne synchronise que les événements présents dans les deux automates, commun aux deux alphabets, l'état initial est la composition de deux états initiaux. (Matilde et al., 2013).



- Avec le Grafcet

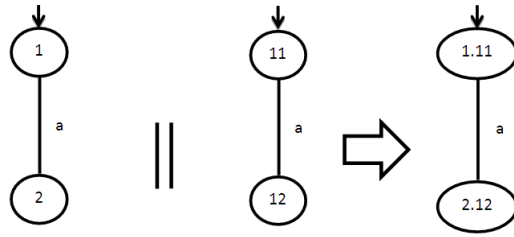


On constate alors qu'il y a apparition d'un troisième état (2.12) : conséquence de la prise en compte de la possibilité de simultanéité d'apparition de deux événements.

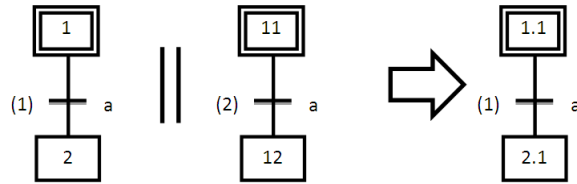
**Proposition 2 :** Composition de deux événements présents de part et d'autre

Le produit est une intersection (Matilde et al., 2013).

- Avec les automates



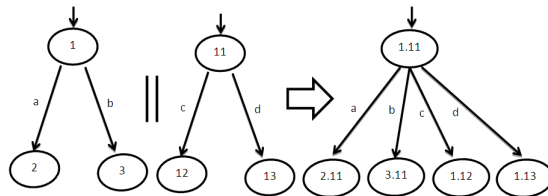
• Avec le Grafctet



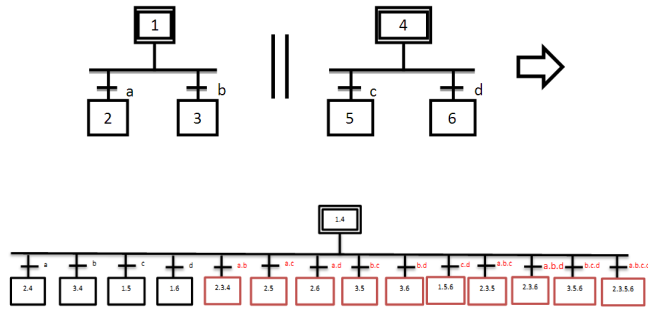
Le résultat est pareil à celui des automates finis.

**Proposition 3 :** Extension de la composition des évènements distincts.

• Avec automates



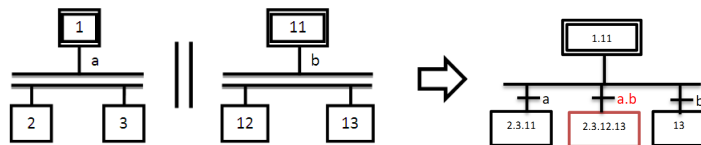
• Avec le Grafctet



On remarque qu'il y a dix états en plus, à cause de la simultanéité des évènements.

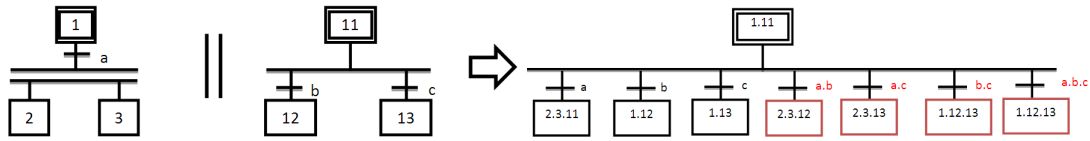
**Proposition 4 :** Composition des divergences en **ET**.

Etant donné qu'il n'existe pas de divergence en **ET** dans un automate, nous ne saurons faire une comparaison. Tout de même en s'inspirant de la logique du comportement Grafctet, nous avons établi ce formalisme :



L'état en rouge est celui qui n'existerait pas pour le cas les automates finis.

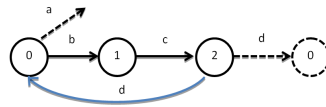
**Proposition 5 :** Composition des divergences en textbfET et en OU.



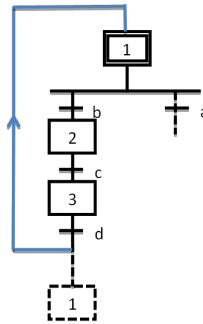
**Proposition 6 :** Règles d'évolution

Lors de la construction de l'automate de comportement commun, lorsque l'on trouve un état qui existe déjà dans le réseau, l'arc est orienté vers ce dernier afin de ne pas avoir deux situations identiques dans un même réseau. Nous adopterons ce principe dans la construction du Grafcet du comportement commun.

- Pour les automates finis



- Pour le Grafcet



## IV. Application

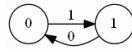
Nous ferons une synthèse de superviseur par une modélisation du système par les automates que nous allons comparer à notre méthode de synthèse basée sur Grafcet . L'algorithme de Kumar sera adapté pour celle à base du Grafcet.

### IV.1. Synthèse de superviseur à base des automates finis

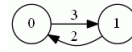
Après avoir modélisé la commande et les spécification sous forme d'automate, nous avons obtenu l'automate de fonctionnement en boucle fermée. A l'aide du logiciel TCT développé par Wonham et téléchargeable sur sa page personnelle (Wonham, 2013), nous avons ressortie le fonctionnement en boucle fermé. Dans ce logiciel, les évènements incontrôlables doivent être des chiffres pairs, alors que les évènements contrôlables sont des chiffres impairs. Le codage est donné comme suit :

$$1 \rightarrow c_1, 0 \rightarrow f_1, 3 \rightarrow c_2, 2 \rightarrow f_2.$$

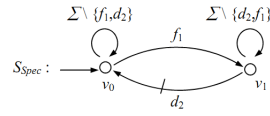
• **Commande**  
Machine 1



Machine 2



• **Spécification**



• **Fonctionnement en boucle fermée**

La figure 4 représente le fonctionnement en boucle fermée.

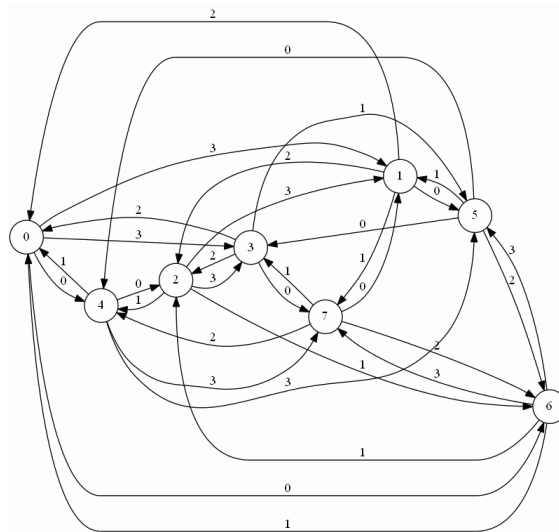


FIGURE 4. Automate du fonctionnement en boucle fermé

• **Superviseur optimal**

Après le processus de synthèse à l'aide de l'algorithme de Kumar, on a le superviseur optimal suivant :

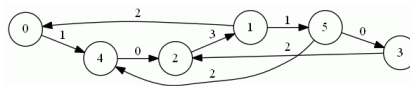


FIGURE 5. Automate superviseur

On a alors 6 états et 8 transitions.



**IV.2. Synthèse de superviseur à base du Grafcet**

• **Commande**

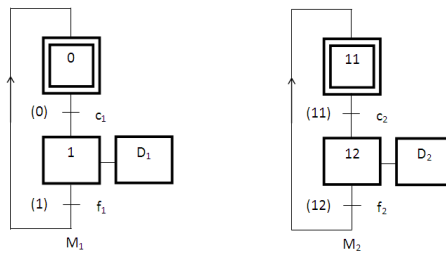


FIGURE 6. Grafkets de commande

• **Spécification**

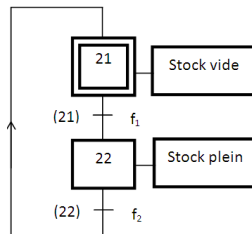


FIGURE 7. Grafcet de spécification

• **Grafcet de comportement commun**

Après l'application de notre formalisme de composition de Grafkets, l'on obtient le Grafcet de comportement commun de la figure 8.

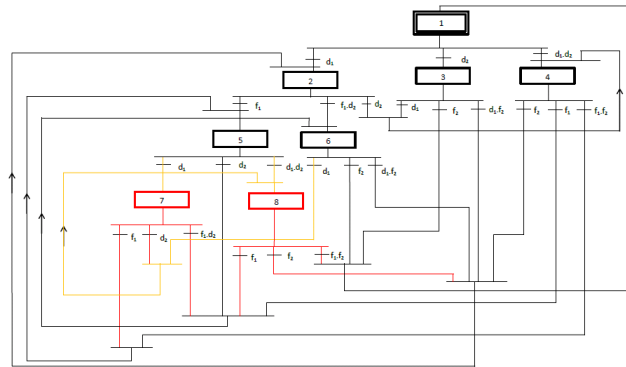


FIGURE 8. Grafcet de comportement commun

On a alors 6 étapes et 15 transitions.

Etats du comportement commun	Etats actifs des Grafjets de base
1	0.11.21
2	1.11.21
3	0.12.21
4	1.12.21
5	0.11.22
6	0.12.22
7	1.11.22
8	1.12.22

FIGURE 9. Tableau des Etats du Grafjet de comportement commun

Les situations (1.11.22) et (1.12.22) sont interdites car la machine 1 ne peut être en marche lorsque le stock est encore plein.

Les arcs rouges sont ceux qui découlent des états interdits, tandis que ceux qui sont oranges désignent les arcs qui vont vers les états interdits. D'après l'algorithme de Kumar, les arcs rouges et oranges doivent être supprimés. Dans notre cas, il n'y a pas d'état faiblement interdit car il n'existe pas une séquence d'évènement incontrôlable qui mène vers un état interdit.

- **Grafjet superviseur**

Après suppression des arcs rouges et orange, nous obtenons le Grafjet superviseur de la figure 9.

Le superviseur est contrôlable.

### IV.3. Simulation de la dynamique du Grafjet superviseur

Lors de la simulation par le logiciel GRAFCET2D, nous avons remarqué une situation qui ne nous arrangeait pas : le cas où l'étape 1 est active et les réceptivités a et b vraies au même moment.

Les étapes 1, 2 et 3 s'activent en même temps comme illustre la figure 10.

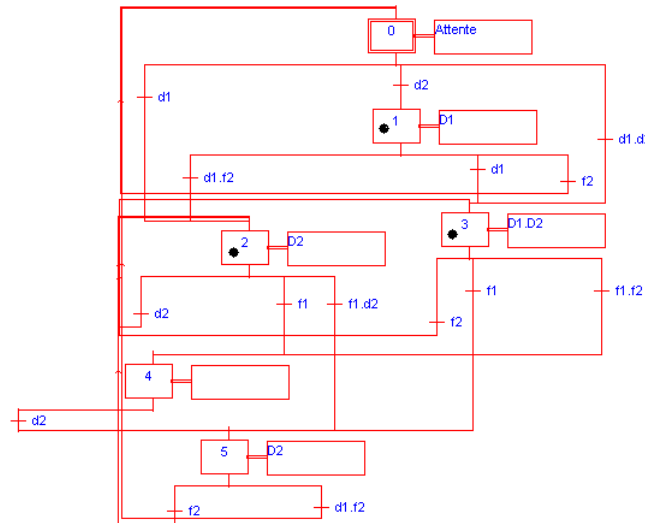


FIGURE 11. Simulation dans GRAFCET2D

Cette situation est normale d'après la 4<sup>e</sup> règle d'évolution du grafjet, ce qui valide même le logiciel. En réalité, cette situation est indésirable dans notre contexte. Au départ, nous avons choisi de faire une synthèse de superviseur par rapprochement à celle des automates finis. Nous avons vu que dans l'automate superviseur, un seul état est actif à la fois. Pour répondre à cette

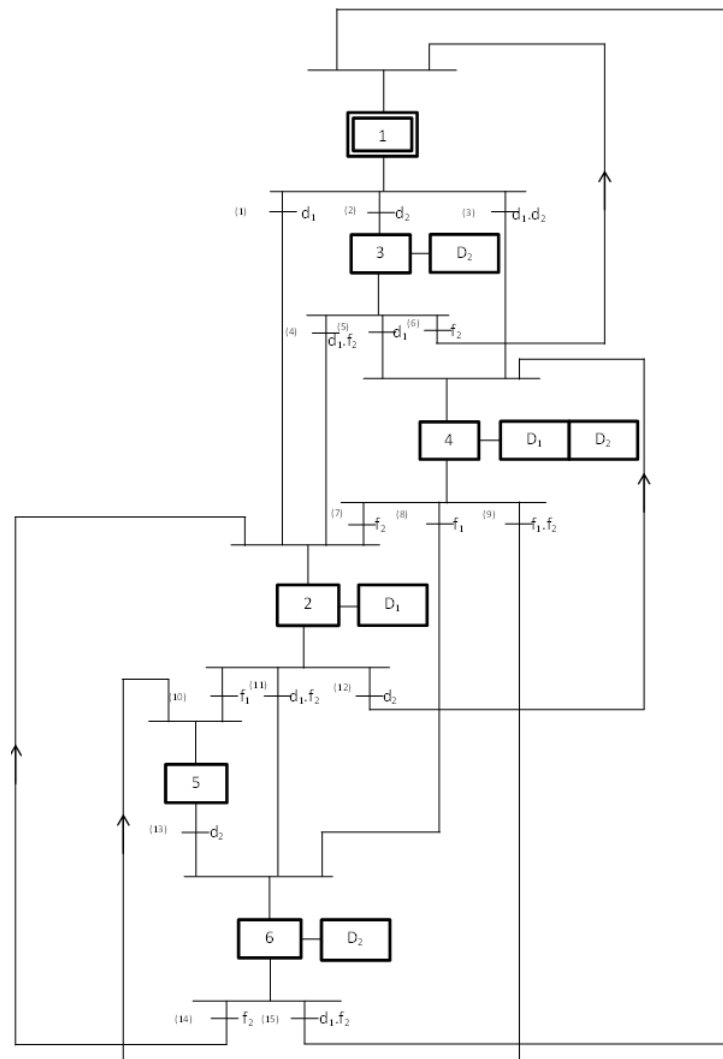


FIGURE 10. Grafcet superviseur

exigence, nous allons contraindre notre Grafcet à évoluer suivant ce principe. Ainsi, l'évolution du Grafcet vers la branche de la transition (réceptivité  $a$ ) ne sera possible que si la réceptivité  $b$  n'est pas validée. Ce principe sera donc appliqué à toutes les autres transitions se trouvant dans cette situation.

On a alors le Grafcet corrigé de la figure 11 :

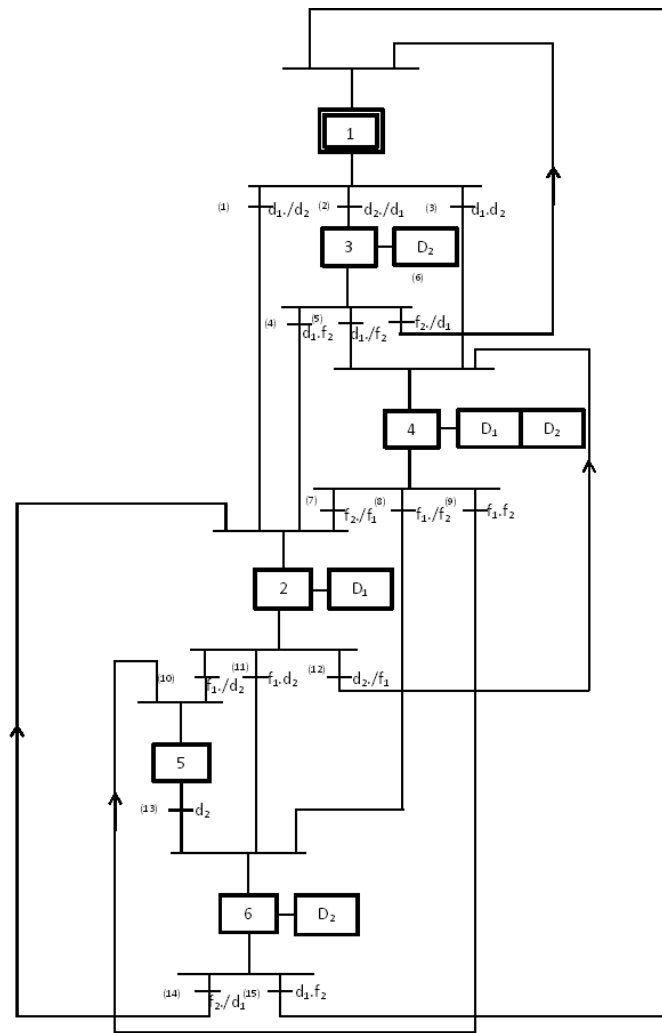


FIGURE 12. Grafcet superviseur corrigé

Nous avons simulé une fois de plus et le résultat est satisfaisant cette fois-là.

## V. Conclusion

Il était question de développer une méthode de synthèse de superviseur totalement à base du Grafcet. Nous avons établi des règles pour la composition des Grafcets. Nous avons considéré le cas d'une possible apparition simultanée d'évènements, cependant il faut corriger le Grafcet final en le forçant de n'emprunter qu'une seule voie lors de l'évolution du système. Nous avons appliqué la méthode sur un système manufacturier et nous avons obtenu un superviseur plus permissif que le superviseur obtenu en utilisant les automates finis. Le problème de l'explosion combinatoire persiste, c'est ce qui nous amène à penser à une autre méthode structurelle par rapprochement aux Réseaux de Petri.

## Bibliographie

(**Brandin, 1989**) B. A. Brandin : Supervisory control of Discrete Event Systems with forcible events, Thesis for the degree of master of applied science, University of Toronto, 1989.

(**Kumar, 1991**) R. Kumar : Supervisory Synthesis Techniques for Discrete Event Dynamical Systems, Thesis for the degree of Doctor of Philosophy, Université du Texas, Austin, Août 1991.

(**Marangé et al., 2008**) Pascale Marangé, Abdelouahed Tajer, François Gellot, Véronique Carré-Ménétrier : Etude du comportement global d'un SED en vue de la validation de sa commande spécifiée par Grafcet ,STIC–UFR Sciences Exactes et Naturelles de Reims - Moulin de la Housse , hal-00385483, version 1 - 19 May 2009.

(**Matilde et al., 2013**) Matilde Machin, Jérémie Guichet, David Powell, Hélène Waeselynck : Introduction à la synthèse de superviseur, LAAS-CNRS ,hal-00804879 version 1, num 13067, 2013.

(**Ndjab, 1999**) C. H. Ndjab, : synthèse de la commande des SED par Grafcet, thèse de doctorat de l'université de Reims Champagne Ardenne, soutenue en 1999.

(**Philippot et al., 2003**) A. Philippot., A. Tajer, F. Gellot., V. Carré-Ménétrier, « Synthèse de la commande spécifiée en Grafcet : application à un préhenseur pneumatique » Actes de MSR 03, pp 61-75, 6-8, Metz, octobre 2003

(**Ramadge et Wonham, 1987**) P. J. Ramadge, and W. M. Wonham, Supervisory control of a class of discrete event processes, SIAM Journal of Control and Optimization, 25 (1) :206-230, Janvier 1987.

(**Tajer, 2005**) Abdelouahed TAJER : Contribution aux approches formelles de synthèse de commande spécifiée par grafcet. Thèse de doctorat de l'université de Reims Champagne Ardenne, 2005.

(**Wonham, 2013**) Page personnelle de W. M. Wonham [www.control.toronto.edu /people /profs/wonham/wonham.html](http://www.control.toronto.edu/people/profs/wonham/wonham.html), visité le 12/09/2013

Wayang Senguel, Bitjoka Laurent, Gonza Mohaman, Donfack B. Omer and Ngako B. Axel  
Equipe Automatique et Traitement du Signal (ATS), Laboratoire Energie, Signal Image et Automatique (ESIA), ENSAI-Université de Ngaoundéré, BP 455, Ngaoundéré

e-mail: [senguel.wayang@univ-ndere.cm](mailto:senguel.wayang@univ-ndere.cm)

e-mail: [lbitjoka@univ-ndere.cm](mailto:lbitjoka@univ-ndere.cm)

e-mail: [mohaman.gonza@univ-ndere.cm](mailto:mohaman.gonza@univ-ndere.cm)

e-mail: [omer.donfack@univ-ndere.cm](mailto:omer.donfack@univ-ndere.cm)

e-mail: [brice.ngako@univ-ndere.cm](mailto:brice.ngako@univ-ndere.cm)