



FILIERE D'INGENIEUR EN ELECTRONIQUE

PROJET DE FIN D'ETUDES

RÉSEAUX DE PÉTRI



Proposé Par :

Mme. Georgette MONDON

Docteur - Ingénieur

Maître assistante à l'E.N.P.

Etudié Par :

Mlle. GUERTI Mhania

Mr. DOUKHANI Layachi

PROMOTION JUIN 1978



DEPARTEMENT ELECTRICITE

FILIERE D'INGENIEUR EN ELECTRONIQUE

PROJET DE FIN D'ETUDES

RÉSEAUX DE PÉTRI

Proposé Par :

Mme. Georgette MONDON

Docteur - Ingénieur

Maître assistante à l'E.N.P.

Etudié Par :

Mlle. GUERTI Mhania

Mr. DOUKHANI Layachi

PROMOTION JUIN 1978

Dédicaces

إلى والدي

فليس بمعجيب أن أعجز عن وصف ما أكنه لكما فوي
قلبي من عظيم الحب، فإن أكنفي من القول بالإقرار
بالعجز، والإعتراف لكما بالفضل على مدى الحياة فوي
فجأحوي الذي هو ثمرة غرسكما.

A mes frères et sœurs avec toute mon affection

A Mademoiselle Intiche Lolisa en témoignage d'amitié

A tous mes parents

A tous mes amis

Intonia GUERTI

A ma mère.

A mes sœurs et à mon frère A.

A tous mes oncles en témoignage de profonde reconnaissance

A tous mes amis

Layachi DOUKHANI

Remerciements

Madame Georgette Mondon, notre promotrice, a malgré ses nombreuses charges, permis une amélioration du contenu de ce projet. C'est avec un grand plaisir que nous lui adressons nos plus sincères remerciements.

Monsieur **TRAN VAN HOÏ**, assistant à l'ENP, a bien voulu s'intéresser à notre travail. Nous lui exprimons nos remerciements les plus chaleureux.

Nous exprimons notre vive reconnaissance, à Mademoiselle **MITICHE Latifa**, pour son aide efficace.

Que Messieurs, Le Directeur, Le Chef de département d'Électricité à l'ENP ainsi que tout le Corps enseignant de l'Université de Constantine et de l'ENP, qui a contribué à notre formation, veuillent bien trouver ici l'expression de notre profonde gratitude.

Enfin, nos remerciements vont à Monsieur **DOUADI Mabrouk** pour l'aide et la documentation qu'il nous a fournies.

Les développements des systèmes logiques de commande

Bien que les circuits de commutation fussent employés depuis plusieurs années (téléphones, chemins de fer), il fallut attendre 1938 pour voir apparaître une analyse symbolique de leurs propriétés, utilisant l'algèbre de BOOLE. Elle fut rapidement suivie d'une étude de systèmes dotés d'une capacité de mémorisation, appelés ensuite systèmes séquentiels.

Parallèlement à ces études à caractère théorique, les circuits de commutation allaient trouver un domaine d'application considérable. L'accroissement du volume de calculs à traiter, qui résulte lui-même du développement industriel et commercial a fait que les machines à calculer existantes, basées sur des principes entièrement mécaniques, ne pouvaient plus apporter une puissance suffisante. Ainsi est née l'idée d'utiliser les circuits de commutation, plus rapides puisque faisant appel à l'énergie électrique, pour la conception de systèmes de traitement des nombres. Le premier calculateur, MARK I (Harvard University) date de 1942; il employait exclusivement des circuits à relais. Il fut suivi d'ENIAC (Bell Téléphone) en 1946, premier calculateur à tube à vide. C'est en 1950 avec EDSAC (Cambridge University) et EDVAC (University of Pennsylvania) qu'apparaissent les véritables ancêtres des calculateurs actuels et les machines à programmes enregistrés telles qu'elles ont été définies par VON NEUMANN.

On discerne donc deux branches bien distinctes qui se sont formées vers 1940 : d'une part les systèmes séquentiels ou d'une façon plus générale les automates finis, d'autre part les calculateurs ou ordinateurs, ou systèmes de traitement des nombres. Comment ont évolué ces

deux branches, qu'ont elles apporté ?

La première méthode formelle pour la synthèse des systèmes séquentiels est due à D.A. HUFFMAN. Elle se décompose en quatre phases:

- Ecriture du cahier des charges sous forme graphique
- réduction de l'énoncé primitif à l'aide d'une représentation tabulaire: table des phases
- Codage impliquant la notion de variable interne associée à celle d'état
- Ecriture des équations des variables internes et des sorties.

Les circuits à réaliser sont de nature asynchrone, c'est-à-dire à évolution libre pour chaque vecteur d'entrée et états présents. On les classe en général en deux groupes: machines de MOORE, machines de MEALY dont la distinction provient de l'indépendance des sorties vis-à-vis des entrées, dans le premier groupe.

Ces résultats ont entraîné un nombre considérable de travaux portant sur les quatre parties de la méthode d'HUFFMAN:

- les idées de KLEENE ont amené l'algèbre des expressions régulières, outil de représentation du fonctionnement d'un système en termes de relations entrée - sorties.

- le concept d'équivalence d'états et d'équivalence de machines provient du souci de minimisation de la table des phases primitives.

- l'étape de codage est sans doute celle qui a le plus attiré l'attention des chercheurs, un nombre considérable de publications ayant paru et paraissant encore sur ce sujet. De multiples solutions y sont préconisées: codages par adjacence, codages assurant des transitions simples sans course, codages universels, en sont quelques exemples.

- la génération des équations constitue un problème classique de recouvrement qui a donné naissance à bon nombre de méthodes.

Un problème supplémentaire intervient après que l'étape de synthèse ait été appliquée : celui de la suppression des aléas dus à l'asynchronisme. C'est certainement le plus ardu et il ne peut être résolu sûrement qu'après que soit donnée la réalisation définitive. Il fait appel, en général, à des procédures de simulation plus ou moins élaborées selon que l'on tient compte ou non des caractéristiques réelles des modules constituant le circuit, et selon le niveau de complexité des modules.

Tout ceci montre que la théorie des systèmes séquentiels asynchrones est assez évoluée. Elle apporte certainement beaucoup du point de vue de la connaissance des problèmes qui se posent au moment de la conception d'un circuit. Quant à l'applicabilité des méthodes qu'elle contient, elle est limitée à des machines de très petite taille. Certains auteurs ont essayé d'aller plus loin en décrivant des algorithmes applicables à de "très grandes tables de fluence", soit 100 colonnes et 30 lignes approximativement, ce qui correspond à un système à 7 entrées et 5 variables internes. La limite reste donc assez basse.

Les ordinateurs ont tout de suite fait intervenir la notion de synchronisation qui consiste à assurer les transferts d'information dans un système, au rythme d'un signal de référence de fréquence fixe, appelé horloge. Ceci permet de simplifier la conception, notamment en ce qui concerne le codage et les problèmes d'aléas. Un ordinateur a pour but de traiter des données numériques codées sous forme binaire, traitement qui s'effectue dans un bloc arithmétique constitué de registres, compteurs, additionneurs. On a donc affaire ici à une structure bien définie qui, si elle n'implique pas de grandes théories mathématiques, du moins au niveau du matériel, s'est montrée

d'un tel intérêt que les constructeurs n'ont eu d'autre souci que de l'améliorer. L'accroissement des performances a été rendu possible grâce à l'apparition du transistor et au développement technologique prodigieux qui a suivi : circuits intégrés, transistors M.O.S. La miniaturisation des circuits et leur coût de plus en plus faible ont procuré une souplesse telle aux calculateurs que leur domaine d'application devient illimité. De plus, la rapidité des circuits intégrés fait que l'ordinateur utilisé en chaîne devient sous employé au niveau de l'unité centrale. Ainsi est apparu le multitraitement : prise en compte simultanée de plusieurs tâches. Ceci implique que l'unité de commande doit coordonner l'allocation de ressources (par exemple de plusieurs unités arithmétiques et logiques dans le cas de systèmes parallèles) aux utilisateurs qui en font la demande et qui agissent donc en concurrence pour avoir accès aux unités de traitement ou de stockage de l'information. Ici le synchronisme est peu avantageux, car il oblige à choisir des périodes d'horloge garantissant la bonne exécution des tâches d'où un temps mort avant leur prise en compte. De plus l'occupation plus grande des unités de commande et de traitement impose des transferts d'information beaucoup plus rapides ce qui pose des problèmes aigus de synchronisation des signaux.

Cet état de fait a amené à repenser aux systèmes asynchrones pour lesquels les inconvénients ci-dessus n'interviennent pas ; mais les théories précédemment énoncées n'ont pas pour autant trouvé là un domaine d'application. Il a été fait appel aux idées de **MULLER** qui a posé les bases des systèmes asynchrones fonctionnant en mode appel-réponse, idées concrétisées dans le calculateur **ILLIAC II** (University of ILLINOIS) et depuis un peu tombées dans l'oubli.

D'autre part, la conception de systèmes à fonctionnement

parallèle nécessite un outil de représentation. Au niveau du logiciel*, il existe de nombreux modèles de schémas de programmes, mis au point pour étudier et définir des langages de programmation parallèles et pour traiter certains types de problèmes provenant du multitraitement. Ces modèles ne sont pas avérés satisfaisants pour leur application au matériel et il a fallu se tourner une fois de plus vers des travaux antérieurs : Ceux de CARL ADAM PETRI. Le modèle auquel il a donné son nom - les réseaux de PETRI - a été étudié plus formellement par HOLT et des recherches très actives sont actuellement menées, notamment au M.I.T. afin de cerner ses propriétés. C'est PATIL qui le premier a eu l'idée d'associer l'outil de représentation que constituent les réseaux de PETRI, et les systèmes à appel-réponse également appelés, lorsqu'on leur confère certaines propriétés, indépendants de la vitesse. Ceci a amené une conception modulaire qui permet le passage direct du réseau de PETRI représentant le fonctionnement du système désiré à une réalisation physique indépendante de la vitesse. Cette approche est intéressante.

* Journal officiel du 12 Janvier 1974 : Logiciel remplace désormais le mot anglais "software".

Table des matières

Introduction

Chapitre I Définitions et marquage du réseau de PETRI

- I. 1 Définition formelle et représentation graphique
- I. 2 Marqueurs (marquage)
- I. 3 Evolution des marqueurs
- I. 4 Notion de conflit
- I. 5 Etude du marquage d'un réseau
- I. 6 Propriétés définies sur les réseaux de PETRI (RdP)

Chapitre II Représentation matricielle des RdP

- II. 1 Matrices de description ou d'incidence
- II. 2 Validation d'une transition
- II. 3 Evolution du marquage

Chapitre III Décomposition du réseau de PETRI

- III. 1 Introduction
- III. 2 Généralités
- III. 3 Graphe d'état
- III. 4 Décomposition en sous-graphe d'état
- III. 5 Macrographe (MG)
- III. 6 Propriétés du MG associé à un RdP
- III. 7 Graphes de transition d'états
- III. 8 Conclusion

Chapitre IV Pondération d'un RdP

- IV. 1 Introduction
- IV. 2 Définitions
- IV. 3 Propriété fondamentale
- IV. 4 Expressions matricielles de la pondération

IV.5	Pondération des Rdp libre-choix conformes	
IV.6	Pondération d'un Rdp quelconque vivant et à marquage fini	
IV.7	Conclusion	
<u>Annexe</u>	<u>Compléments sur les Rdp</u>	
A.1	Extensions des Rdp	
<u>Chapitre V</u>	<u>Application des Rdp à la synthèse des machines séquentielles</u>	
V.1	Réseau de PETRI et machines séquentielles	
V.2	Exemples d'utilisation	
V.3	Conclusion	
<u>Chapitre VI</u>	<u>Réduction du nombre d'états</u>	84
VI.1	Introduction	
VI.2	Buts et principes de la simplification	
VI.3	Première simplification : recherche des places implicites	
VI.4	Deuxième simplification : Fusion des places	
VI.5	Un exemple d'application	
VI.6	Processus de simplification	
VI.7	Conclusion	
<u>Chapitre VII</u>	<u>Matérialisation des systèmes décrits par Rdp</u>	93
VII.1	Introduction	
VII.2	Les principaux modules	
VII.3	Définition structurelle des modules	
<u>Chapitre VIII</u>	<u>Réalisations « Logiques programmées » une 1^{ère} approche</u>	102
VIII.1	Introduction	
VIII.2	Les mémoires à circuits intégrés	
VIII.3	Les microprocesseurs	
VIII.4	Réalisations « logiques programmées » des Rdp sans fonctionnement parallèles (graphes d'état)	
VIII.5	Réalisations « Logiques programmées » des Rdp avec évolutions simultanées	
VIII.6	Automate programmable	
VIII.7	Conclusion	
<u>Conclusion</u>		109

Introduction

Les méthodes formelles de synthèse des systèmes séquentiels n'ont jamais reçu un accueil très favorable auprès des utilisateurs à cause de leurs difficultés d'application à des problèmes complexes. Les limitations qui en sont la cause se manifestent à deux niveaux : à la description, lors de la conception.

Pour décrire le fonctionnement d'un automatisme, on utilise les tables des phases ou diagrammes des phases primitives où toutes les combinaisons d'entrées doivent être envisagées pour chaque phase, d'où un maniement très lourd et peu commode. Par contre, les méthodes dites intuitives procèdent par construction de séquences d'évolutions c'est-à-dire de suites d'événements et commandes qui sont beaucoup plus faciles à établir car n'est spécifié que ce qui apporte une modification dans le comportement du système. Cependant l'information obtenue est moins précise et si la conception s'effectue au moyen des méthodes classiques (réduction et codage par adjacence par exemple), les conditions non spécifiées le seront alors et de façon imprévisible, inconvénients qui s'ajoutent aux problèmes de courses et d'aléas que toutes ces méthodes impliquent. L'utilisation de codages à priori introduit la notion de réceptivité c'est-à-dire l'évolution d'un système seulement pour les conditions spécifiées, ce qui offre les avantages suivants : modifications faciles à apporter et compréhension simple de la réalisation grâce à son respect de la description initiale.

Malgré ceci un autre problème important demeure : la difficulté de représenter les événements prioritaires. Ceux-ci sont susceptibles de modifier le comportement d'un système quelle que soit la configuration présente et leur prise en compte dans un diagramme des transitions doit s'effectuer au niveau de chaque phase d'où une lourdeur considérable, ce qui montre que le modèle est mal adapté.

On voit alors apparaître l'intérêt d'un modèle capable de décrire

simplement des événements intervenant simultanément ou « en parallèles ». Ce modèle existe sous forme graphique : les réseaux de PETRI (RdP). Ils permettent la représentation du comportement d'un système où les évolutions simultanées, exclusives, l'attente sont explicitement schématisées, et où la notion de réceptivité intervient également.

A partir de ce parti pris, l'architecture du projet est simple. Un premier chapitre essaiera d'introduire une formalisation un peu abstraite de la notion de réseau de PETRI. Un seul chapitre sera consacré à la représentation matricielle du réseau de PETRI, le deuxième de ce projet.

Les chapitres III et IV établissent deux propriétés structurelles des réseaux de PETRI : décomposition et pondération

- la première introduit les notions de macrographes, de recouvrement de réseau de PETRI par graphes d'état qui permettent une réalisation programmée.

- la deuxième met en évidence des invariants dont l'utilisation conduit à une réalisation modulaire.

Suivent trois chapitres qui, de la formulation du cahier des charges au schéma logique de fonctionnement, essayent de montrer comment un outil proche des organigrammes peut être utilisé et comment une réalisation technologique peut en accroître l'intérêt.

Enfin, un dernier chapitre aborde les réalisations logiques programmées (mables) et donne la place qu'il mérite à un produit nouveau qu'on ne peut plus ignorer davantage : l'automate programmable.

Le projet se termine enfin par une bibliographie sur les machines séquentielles et les réseaux de PETRI.

Définitions et marquage du réseau de PETRI

I.1 Définition formelle et représentation graphique

Un réseau de PETRI R est un triplet $\langle P, T, A \rangle$ où :

- $T = \{t_1, t_2, \dots, t_m\}$ est un ensemble fini de transitions
- $P = \{p_1, p_2, \dots, p_n\}$ est un ensemble fini de places
- $A = \{a_1, a_2, \dots, a_k\}$ est un ensemble fini d'arcs orientés

qui assurent la liaison d'une place vers une transition ou d'une transition vers une place.

R est donc un graphe orienté. T et P représentent les nœuds du graphe.

Graphiquement, (figure I.1), un cercle représente une place; une barre, une transition; une flèche, un arc.

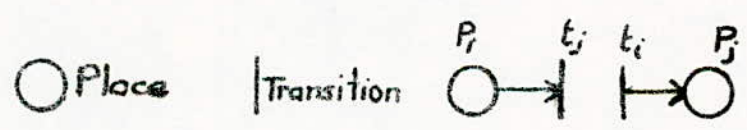


Figure I.1

Remarque

Un arc joindra toujours une place à une transition ou une transition à une place mais jamais une place à une place ou une transition à une transition.

I.2 Marqueurs (marquage)

Le marquage d'un réseau R est une fonction de P dans les nombres naturels, $N = \{0, 1, \dots\}$. (Cette fonction sera aussi appelée

fonction de marquage). On le représente en plaçant des marqueurs (ou jetons ou marques) dans les places.

La figure 1.2 représente trois marquages différents de places; le marquage du réseau est alors l'ensemble de places marquées à un instant donné. Ceci nous permet d'utiliser

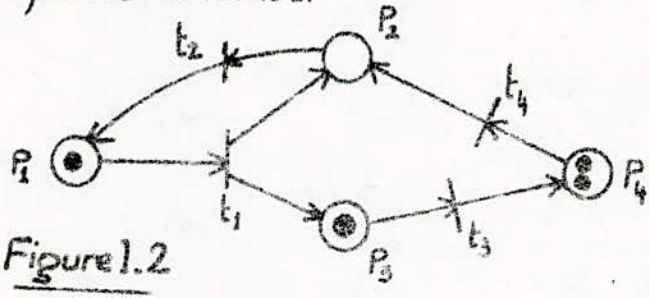


Figure 1.2

la notation $M(p_i)$ pour spécifier le nombre de marqueurs dans la place p_i .

Le marquage M peut donc être représenté par un vecteur colonne dont les composantes $M(p_1), M(p_2), \dots$, correspondent au nombre de marqueurs situés dans les places p_1, p_2, p_3, p_4 .

$$M = \begin{pmatrix} M(p_1) \\ M(p_2) \\ M(p_3) \\ M(p_4) \end{pmatrix} \quad \text{soit} \quad M = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 2 \end{pmatrix}$$

On voit que les places peuvent être vides ou marquées. Cette notion nous permet de compléter la définition du paragraphe 1.1 de la façon suivante:

Un réseau de PETRI marqué (R, M_0) est un graphe orienté défini par le quadruplet $\langle T, P, A, M_0 \rangle$ où :

- $T = \{t_1, t_2, \dots, t_m\}$ est un ensemble fini de transitions
- $P = \{p_1, p_2, \dots, p_n\}$ est un ensemble fini de places
- $A = \{a_1, a_2, \dots, a_k\}$ est un ensemble fini d'arcs orientés qui

connectent places et transitions et réciproquement.

- $M_0 : \{ P \rightarrow N_+ \}$ est la distribution initiale des marqueurs, c'est-à-dire l'ensemble des places qui ont, initialement, des marqueurs.

A chaque transition est associé un ensemble de places d'entrée et un ensemble de places de sortie. Les places d'entrée sont les places d'où sont issus les arcs orientés vers la transition. Les places de sortie sont les places où aboutissent les arcs orientés issus de la transition. Une place peut être à la fois place d'entrée et de sortie d'une transition. On l'appelle boucle élémentaire (Figure 1-3).

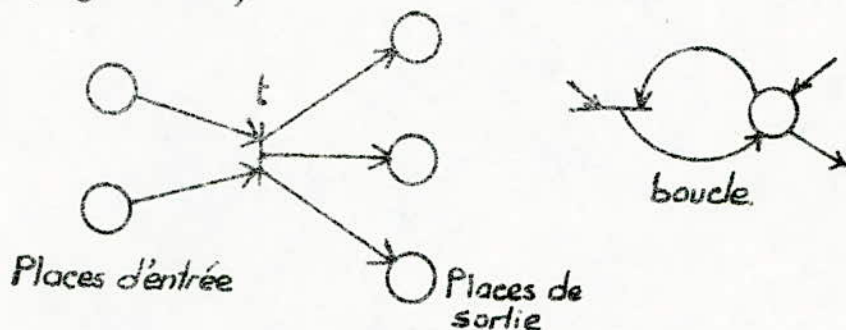
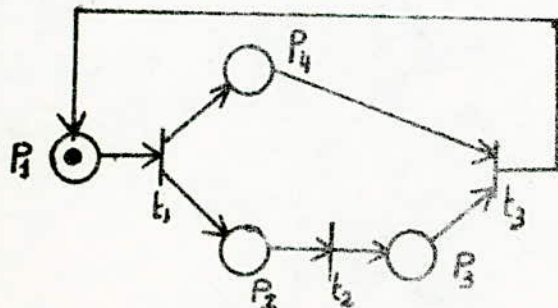


Figure 1-3

Exemple

La figure 1.4 représente un réseau de PETRI marqué $\langle T, P, A, M_0 \rangle$ qui comporte trois transitions $T = \{ t_1, t_2, t_3 \}$, quatre places $P = \{ P_1, P_2, P_3, P_4 \}$, huit arcs $A = \{ (P_1, t_1), (t_1, P_2), (t_1, P_4), (P_2, t_2), (t_2, P_3), (P_3, t_3), (t_3, P_1), (P_4, t_3) \}$ et un marquage initial M_0 tel que la place P_1 possède un marqueur, les autres places étant vides.



$$M_0 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Figure 1.4

1.3 Evolution des marqueurs

Nous avons, pour l'instant, défini un outil en quelque sorte statique. Il faut y faire intervenir une évolution temporelle. Celle-ci sera obtenue par l'évolution du marquage dans le temps. Définissons donc l'opération de franchissement d'une transition par des marqueurs.

Si une transition t est telle qu'il existe au moins un marqueur dans chaque place d'entrée, t est dite franchissable (ou validée ou sensibilisée). Seules les transitions validées peuvent être franchies par les marqueurs. Franchir t , c'est enlever un marqueur de chaque place d'entrée et ajouter un marqueur dans chaque place de sortie. Sur le réseau de la figure 1.5, on a représenté un franchissement de t .

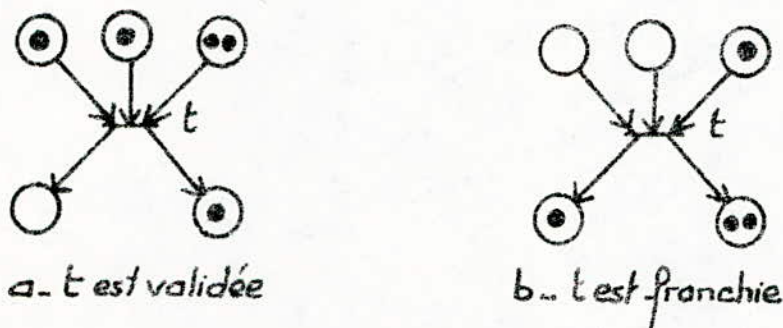
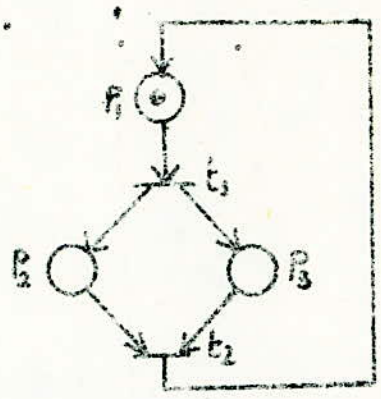


Figure 1.5

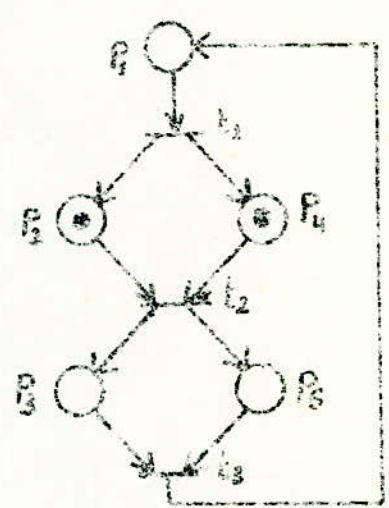
Remarques

Le franchissement d'une ou d'un ensemble de transitions validées simultanément modifie ou ne modifie pas le nombre total de marqueurs d'un réseau. Le franchissement de la transition t_1 du réseau donné par la figure 1.6 augmente le nombre de marqueurs du réseau. Le franchissement de la transition t_2 du réseau représenté par la figure 1.7 n'augmente pas le nombre de marqueurs du réseau.



t_1 est validée.

Figure 1.6



t_2 est validée

Figure 1.7

Lorsque plusieurs transitions d'un réseau sont validées leurs franchissements s'effectuent simultanément (figure 1.8).

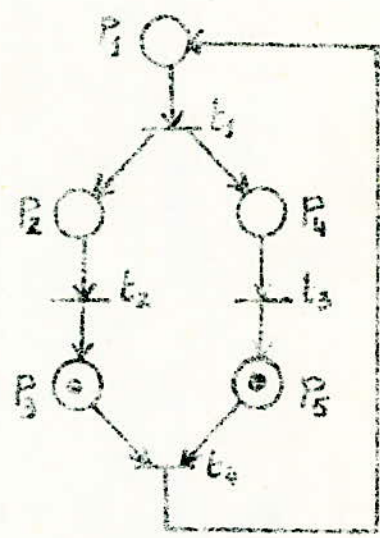
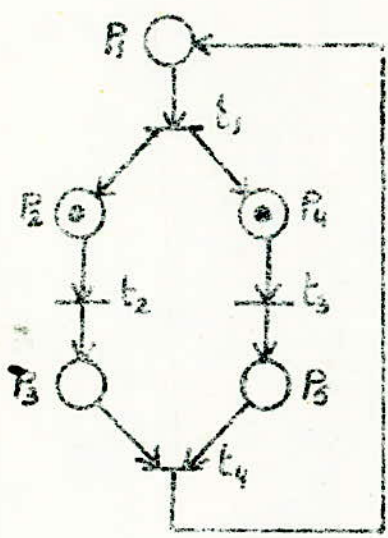


Figure 1.8

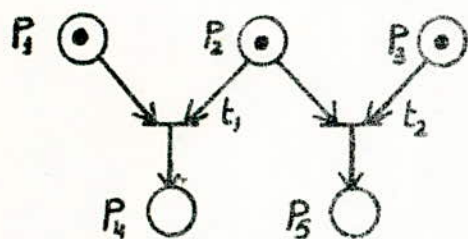
Les transitions t_2 et t_3 du réseau de la figure 1.8 sont validées par le marquage initial, elles peuvent être franchies et le marquage résultant est tel que les places p_2 et p_5 possèdent un seul marqueur, les autres places étant vides.

1.4 Notion de conflit

Lorsqu'il existe une place d'entrée commune à deux ou plusieurs transitions validées simultanément il n'est pas possible d'appliquer la règle de franchissement énoncée précédemment. Les transitions validées sont alors dites en conflit pour le marquage correspondant. L'apparition d'un conflit entraînant l'arrêt de l'évolution du marquage d'un réseau il est nécessaire dans ce cas de rendre prioritaire une des transitions validées. Le franchissement de cette transition modifie le marquage et supprime la validation des autres transitions.

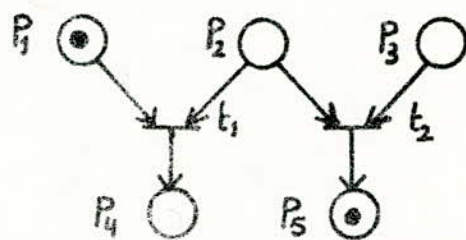
Exemple

Les transitions t_1 et t_2 du réseau élémentaire donné par la figure 1.9.a sont en conflit pour le marquage initial plaçant un marqueur en p_1 , p_2 , p_3 , la place p_2 étant une place d'entrée commune à la transition t_1 et à la transition t_2 (on dit que la place p_2 est partagée). Si on suppose la transition t_2 prioritaire celle-ci peut être franchie et mettre fin à la validation de l'autre (t_1). Le marquage résultant est représenté par la figure 1.9.b.



a. t_1 et t_2 sont validées

Figure 1.9



b. t_2 est franchie

1.5. Etude du marquage d'un réseau

L'application séquentielle à un réseau de PETRI (RdP) de la règle de validation et de franchissement d'une ou d'un ensemble de transitions

permet la détermination de l'ensemble des marquages qu'il est possible d'atteindre à partir d'un marquage initial M_0 . Nous dénoterons $\vec{M}_0 = \{M_0, M_1, \dots\}$ l'ensemble de ces marquages. Cet ensemble (classe) de marquages accessibles à partir de M_0 dépendra à la fois de la structure du réseau et du marquage initial M_0 .

Soit M_i un marquage sensibilisant la transition t_i et M_j le marquage résultant alors du franchissement de t_i . Nous écrivons ceci :

$$M_i \xrightarrow{t_i} M_j$$

Soit σ une séquence finie de transitions $t_i, t_{i+1}, \dots, t_{i+k}$ appartenant à T ; on dit que σ est une séquence de franchissement franchissable à partir de M_i si et seulement si il existe des marquages $M_{i+1}, M_{i+2}, \dots, M_{i+k+1}$ de R tels que :

$$M_i \xrightarrow{t_i} M_{i+1}, M_{i+1} \xrightarrow{t_{i+1}} M_{i+2}, \dots, M_{i+k} \xrightarrow{t_{i+k}} M_{i+k+1}.$$

Nous écrivons ceci :

$$M_i \xrightarrow{\sigma} M_{i+k+1}$$

La classe des marquages conséquents \vec{M}_0 est l'ensemble des marquages accessibles à partir du marquage M_0 par une séquence de franchissement.

$$M_i \in \vec{M}_0 \iff \exists \sigma; M_0 \xrightarrow{\sigma} M_i.$$

Exemple

Le réseau de PETRI représenté par la figure 1.10 comporte un marquage initial M_0 tel que la place p_1 possède un marqueur, les autres places étant vides. L'ensemble des marquages qu'il est possible d'atteindre à partir du marquage initial M_0 est égal à :

$$\vec{[M_0]} = \left\{ \begin{array}{c} M_0 \\ M_1 \\ M_2 \\ M_3 \end{array} \right\}$$

1
0
0
0
0
0

0
1
1
1
0
0

0
1
0
1
1
0

0
1
0
0
0
1

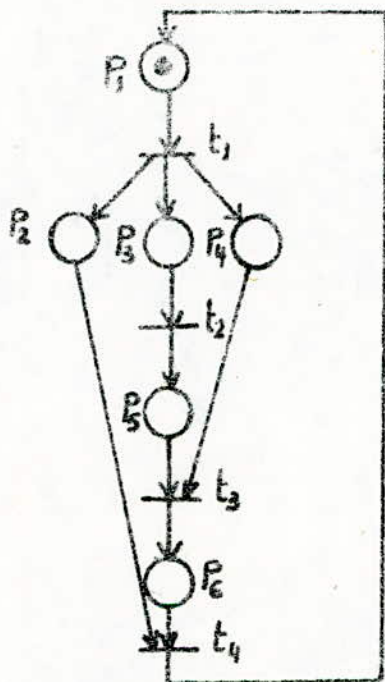


Figure I.10

1.6 Propriétés définies sur les réseaux de PETRI (RdP)

Si on peut classer les réseaux de PETRI suivant leur structure, on peut également les distinguer en considérant certaines propriétés que leur confère leur marquage initial M_0 .

Soit R un réseau de PETRI et M_0 son marquage initial, \vec{M}_0 sera la classe des marquages conséquents correspondante.

1.6.1 Réseau sauf pour un marquage initial donné

R est sauf pour M_0 si et seulement si tout marquage appartenant à \vec{M}_0 est tel qu'il y a au plus un jeton (marque) dans chaque place de R .

Autrement dit : un réseau est sauf ou sain pour M_0 si quel que soit le marquage obtenu à partir de M_0 par une séquence finie de franchissements aucune place ne possède plus d'un marqueur (figure 1.11.a), la figure 1.11.b représente un réseau non sauf car le marquage de la place p_2 augmente à chaque franchissement de la transition t_2 .

Remarque

L'ensemble \vec{M}_0 ne peut comprendre qu'un nombre fini de marquages saufs puisque le nombre de places d'un réseau est fini.

1.6.2 Réseau vivant pour un marquage initial donné

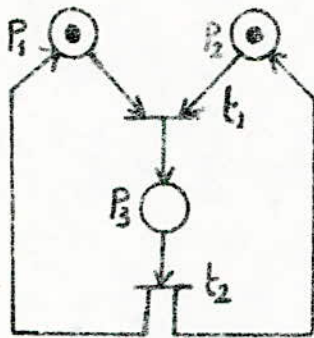
Un réseau est dit vivant pour M_0 si toute transition du réseau peut être validée et franchie par une séquence finie de franchissements.

Autrement dit : si pour toute transition t de R et pour tout marquage M_i de \vec{M}_0 il existe une séquence de franchissement qui franchit t à partir de M_i alors R est dit vivant pour M_0 . Un exemple de réseau vivant est donné par la figure 1.12.a, celui de la figure 1.12.b est non vivant.

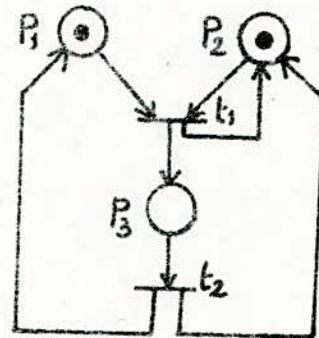
Le réseau de la figure 1.12.b n'est pas vivant puisqu'on obtient le marquage vide après avoir exécuté une séquence complète de commande. Pour le rendre vivant, il suffit de le reboucler pour que la place p_1 devienne place de sortie de la transition t_3 et de la transition à laquelle aucune action n'est associée.

Remarque

Le fait qu'un réseau marqué soit vivant a deux conséquences importantes en pratique. D'une part, cela entraîne l'absence de blocage (marquage accessible à partir duquel aucune transition n'est franchissable). D'autre part, il est certain qu'aucune partie de la commande ne deviendra inaccessible après une certaine séquence de franchissement.

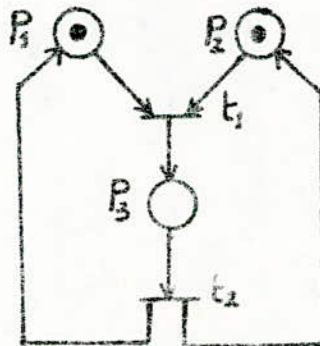


a. Réseau sauf.

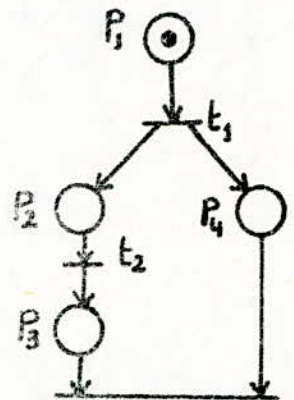


b. Réseau non sauf

Figure 1-11



a. Réseau vivant



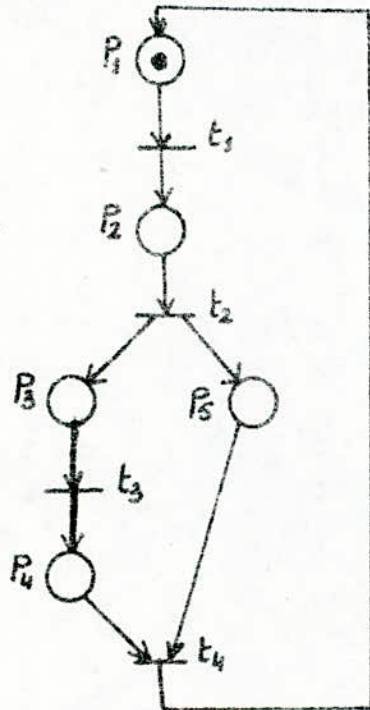
b. Réseau non vivant

Figure 1-12

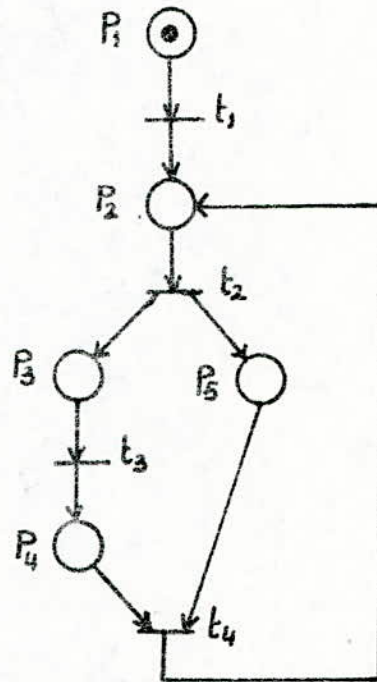
Un réseau vivant et sauf est dit conforme (figure 1-13).

1.6.3. Réseau pseudo-vivant pour un marquage initial

Un réseau est pseudo-vivant pour le marquage M_0 , si, à partir de tout marquage appartenant à \vec{M}_0 , il existe au moins une transition franchissable (figure 1-14).



Réseau conforme
Figure 1-13



Réseau pseudo-vivant
Figure 1-14

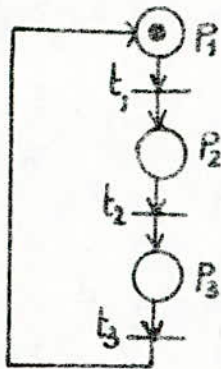
1.6.4 Réseau propre pour un marquage initial

Un réseau de PETRI est propre pour M_0 si et seulement si quel que soit $M_i \in \vec{M}_0$ il existe une séquence de franchissement σ finie telle que $M_i \xrightarrow{\sigma} M_0$ (figure 1-15)

Remarque

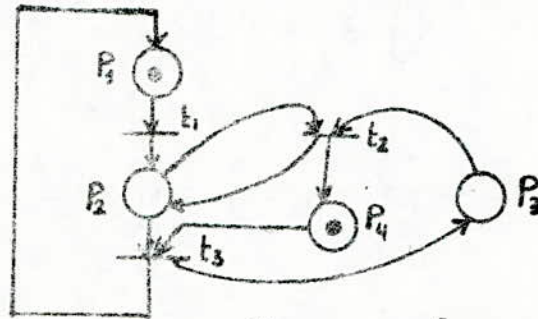
Un RdP conforme pour M_0 n'est pas nécessairement propre pour ce marquage (Figure 1-16).
En général, les systèmes de commande que l'on considère ont un

fonctionnement cyclique, c'est-à-dire lorsqu'ils ont terminé la tâche pour laquelle ils ont été conçus, ils sont prêts à la recommencer. M_0 représentant l'état initial de la commande, il est donc clair que le RdP représentant un système de commande doit être propre.



Réseau propre

Figure I.15



Réseau conforme mais non propre

Figure I.16

1.6.5 Un piège

Un piège est un ensemble P de places du réseau tel que toute transition du réseau qui a une place d'entrée dans P y a aussi une place de sortie. Un piège, une fois marqué, le reste (Figure I.17).

1.6.6. Un Verrou

Un verrou est un ensemble V de places du réseau tel que toute transition du réseau qui a une place de sortie dans V , y a aussi une place d'entrée. Un verrou, non marqué initialement, le restera (figure I.18).

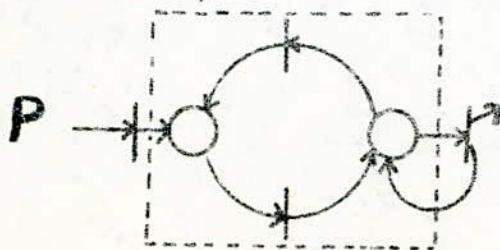


Figure I.17: Piège.

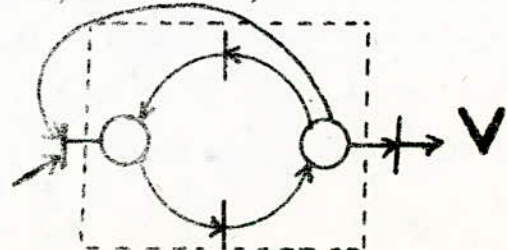


Figure I.18: Verrou.

Représentation matricielle des réseaux de PETRI

Les propriétés et l'évolution du marquage d'un réseau de Pétri $\{P, T, A, M_0\}$ peuvent être étudiées en utilisant une représentation matricielle.

11.1 Matrices de description et d'incidence

Soit $\{P, T, A, M_0\}$ un réseau de Pétri comportant ℓ places $P = \{p_1, p_2, \dots, p_\ell\}$ et m transitions $T = \{t_1, t_2, \dots, t_m\}$. On appelle matrices de description du réseau les tableaux matriciels $S(p, t)$ et $E(p, t)$ suivants:

$S(p, t) = [s_{ij}]$ ou $s_{ij} = 1$ si la place est une place de sortie de la transition t_j et $s_{ij} = 0$ dans le cas contraire.

$E(p, t) = (e_{ij})$ ou $e_{ij} = 1$ si la place p_i est une place d'entrée de la transition t_j et $e_{ij} = 0$ dans le cas contraire. Les matrices $|S(p, t)|$ et $|E(p, t)|$ comportent donc un nombre de lignes égal au nombre ℓ de places et un nombre de colonnes égal au nombre m de transitions du réseau.

Dans le cas particulier où les places d'un réseau ne sont ni place d'entrée, ni place de sortie d'une même transition, il est possible de décrire complètement le réseau par la matrice $|C(p, t)|$ telle que:

$$|C(p, t)| = |S(p, t)| - |E(p, t)|.$$

Chaque coefficient c_{ij} de cette matrice est tel que:

$c_{ij} = +1$ si la place i est une place de sortie de la transition,

$c_{ij} = -1$ si la place i est une place d'entrée de la transition,

$c_{ij} = 0$ si la place i n'est ni une place d'entrée ni une place de sortie de la transition.

EXEMPLE

Le réseau de Pétri représenté par la figure 11.1 comprend

six places $\{P_1, P_2, P_3, P_4, P_5, P_6\}$ et cinq transitions $\{t_1, t_2, t_3, t_4, t_5\}$.
 Il peut être décrit par les deux matrices $|S(p,t)|$ et $|E(p,t)|$ de dimensions $\{6 \times 5\}$ suivantes :

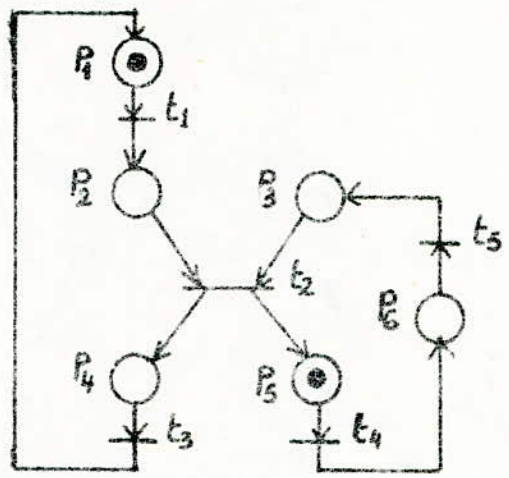


Figure II.1

$$|S(p,t)| = \begin{vmatrix} 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{vmatrix}$$

$$|E(p,t)| = \begin{vmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{vmatrix}$$

Aucune place du réseau n'étant à la fois place de sortie et place d'entrée d'une même transition. D'où la matrice $|C(p,t)|$ (d'incidence)

$$|C(p,t)| = |S(p,t)| - |E(p,t)|$$

et ainsi

$$|C(p,t)| = \begin{vmatrix} 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{vmatrix} - \begin{vmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{vmatrix} = \begin{vmatrix} -1 & 0 & 1 & 0 & 0 \\ +1 & -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & +1 \\ 0 & +1 & -1 & 0 & 0 \\ 0 & +1 & 0 & -1 & 0 \\ 0 & 0 & 0 & +1 & -1 \end{vmatrix}$$

11.2 Validation d'une transition

Une transition t_i d'un réseau de Pétri défini par ses matrices $|S(p,t)|$, $|E(p,t)|$ et son marquage initial $|M_0|$ représenté par un vecteur colonne est validée si quel que soit p , $M(p) \geq E(p,t_i)$ soit en considérant l'inégalité de deux vecteurs colonnes.

$$|M_0| \geq |E(p,t)|$$

Cette inégalité permet de définir en comparant successivement les m colonnes de la matrice $|E(p,t)|$ au vecteur $|M_0|$ l'ensemble des transitions validées par ce marquage initial. Cet ensemble peut être représenté par un vecteur de validation $|V|$. Ce vecteur est un vecteur colonne de dimension égale au nombre m de transitions. Chaque composante v_i de ce vecteur est telle que :

$v_i = 1$ si la transition t_i est validée par le marquage $|M_0|$ et
 $v_i = 0$ dans le cas contraire.

Exemple

Le marquage initial $|M_0|$ du réseau de Pétri représenté par la figure II-1 est égal à :

$$|M_0| = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

La comparaison de ce marquage aux colonnes de

la matrice $|E(p,t)|$ donnée dans l'exemple du paragraphe précédent montre que $|M_0| \geq |E(p,t_1)|$ et $|M_0| \geq |E(p,t_4)|$. Les transitions t_1 et t_4 sont les seules transitions validées pour $|M_0|$ et le vecteur $|V|$ de validation pour ce marquage est égal à :

$$|V| = \begin{vmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{vmatrix}$$

Dans le cas important en pratique où le réseau étudié est sauf le vecteur de validation $|V|$ peut être calculé directement par la relation suivante:

$$V = \overline{{}^t|E(p,t)| \otimes |M_0|}$$

Dans cette relation $|M_0|$ est le vecteur colonne obtenu en remplaçant dans $|M_0|$ les 1 par des 0 et les 0 par des 1, ce qui correspond à une complémentarion.

${}^t|E(p,t)|$ est la matrice transposée de $|E(p,t)|$ dont les coefficients ${}^t(a_{ij})$ sont égaux à a_{ji} et l'opérateur \otimes correspond au produit matriciel booléen obtenu en faisant terme à terme le produit des lignes de ${}^t|E(p,t)|$ et de la colonne $|M_0|$ puis la somme logique de ces produits.

Exemple

Le réseau de Pétri donné par la figure 11.1 est sauf. La matrice ${}^t|E(p,t)|$ est égale à :

$$|E(p,t)| = \begin{vmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{vmatrix} \quad {}^t|E(p,t)| = \begin{vmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{vmatrix}$$

De plus nous avons :

$$\begin{array}{c} \left| \begin{array}{c} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{array} \right| \end{array} \quad \text{soit } |M_0| = \begin{array}{c} \left| \begin{array}{c} 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \end{array} \right| \end{array} \quad \text{et ainsi}$$

$$\bar{V} = {}^t E(p, t) \otimes |M_0| = \begin{array}{c} \left| \begin{array}{cccccc} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array} \right| \end{array} \quad \begin{array}{c} \left| \begin{array}{c} 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \end{array} \right| \end{array} \quad \text{d'où } |V| = \begin{array}{c} \left| \begin{array}{c} 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{array} \right| \end{array}$$

II.3 Evolution du marquage

Nous limiterons l'étude matricielle de l'évolution du marquage d'un réseau de Petri aux réseaux pouvant être décrits par une matrice $|C(p, t)|$.

Dans ce cas les marquages successifs M_1, M_2, M_3, M_4 obtenus à partir du marquage initial M_0 par les franchissements successifs des transitions t_i, t_j, t_k, \dots sont tels que

$$|M_1| = |M_0| + |C(p, t_i)|$$

$$|M_2| = |M_1| + |C(p, t_j)|$$

$$|M_3| = |M_2| + |C(p, t_k)|$$

et pour une séquence finie de franchissements il est possible de relier le marquage final $|M_n|$ au marquage initial $|M_0|$ par une relation de la forme

$$|M_n| = |M_0| + |C(p,t)| \cdot |D| \quad (1)$$

Dans cette expression $|D|$ est un vecteur colonne d'une dimension égale au nombre m de transitions du réseau. Chaque composante d_j de ce vecteur est un entier positif qui correspond au nombre de franchissements de la transition t_j dans la séquence donnée.

Exemple

Le réseau de Pétri représenté par la figure 11.2 peut être décrit par la matrice $|C(p,t)|$ suivante

$$|C(p,t)| = \begin{vmatrix} +1 & -1 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & +1 \\ -1 & +1 & 0 & 0 & 0 \\ 0 & +1 & -1 & 0 & 0 \\ +1 & 0 & 0 & -1 & 0 \\ 0 & 0 & +1 & +1 & -1 \end{vmatrix}$$

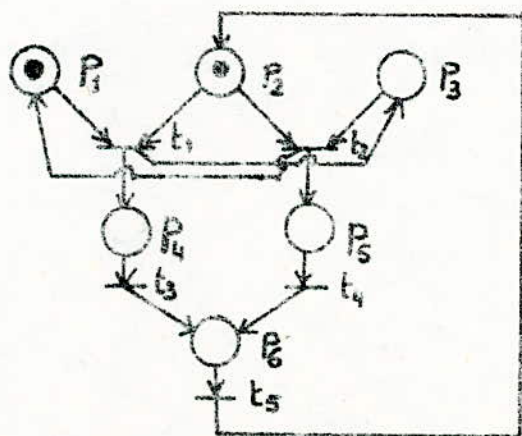


Figure 11.2

Le marquage initial de ce réseau est égal à :

$$\begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

et après la séquence de franchissements

qui correspond au vecteur

$$|D| = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 1 \end{pmatrix}$$

Le marquage obtenu est donné par

$$\begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} +1 & -1 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & +1 \\ -1 & +1 & 0 & 0 & 0 \\ 0 & +1 & 1 & 0 & 0 \\ +1 & 0 & 0 & -1 & 0 \\ 0 & 0 & +1 & +1 & -1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

Remarque

La relation (1) s'écrit si il existe une séquence de franchissements redonnant le marquage initial.

$$|M_0| = |M_0| + |C(p,t)| \cdot |D|$$

soit

$$|C(p,t)| \cdot |D| = 0 \tag{2}$$

L'existence d'un vecteur $|D|$ à composantes entières strictement positives et solutions du système linéaire à coefficients entiers donné par la relation (2) conditionne l'existence d'une séquence de franchissements correspondant à une évolution cyclique du marquage.

Exemple

Cherchons le vecteur $|D|$ qui redonne le marquage initial pour l'exemple précédent

$$|C(p,t)| \cdot |D| = 0$$

$$\begin{pmatrix} +1 & -1 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & +1 \\ -1 & +1 & 0 & 0 & 0 \\ 0 & +1 & -1 & 0 & 0 \\ +1 & 0 & 0 & -1 & 0 \\ 0 & 0 & +1 & +1 & -1 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad \text{on aura le système}$$

$$\begin{aligned} a_1 - a_2 &= 0 & a_1 &= a_2 \\ -a_1 - a_2 + a_5 &= 0 & a_5 &= 2a_1 = 2a_2 \\ -a_1 + a_2 &= 0 & a_1 &= a_2 \\ a_2 - a_3 &= 0 & a_2 &= a_3 \\ a_1 - a_4 &= 0 & a_1 &= a_4 \\ a_3 + a_4 - a_5 &= 0 & a_5 &= 2a_3 = 2a_2 = 2a_4 \end{aligned}$$

Le système admet une infinité de solutions, par exemple

$$a_1 = 1 ; a_2 = a_3 = a_4 = a_1 = 1 ; a_5 = 2a_1 = 2.$$

D'où

$$|D| = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 2 \end{pmatrix}$$

Décomposition du réseau de l'ETRI

III.1 Introduction

Dans ce chapitre, nous donnons une méthode pour recouvrir le réseau de Pétri par des graphes d'état. Cette opération nécessite la définition des macroplaces et des macrographes, notions, qui, de plus, apparaissent comme des facilités pour l'analyse des "gros" réseaux de Pétri.

III.2 Généralités

Vocabulaire

Appelons « nœud ET » une transition ayant plusieurs places d'entrée et/ou plusieurs places de sortie (fig III-1.a).

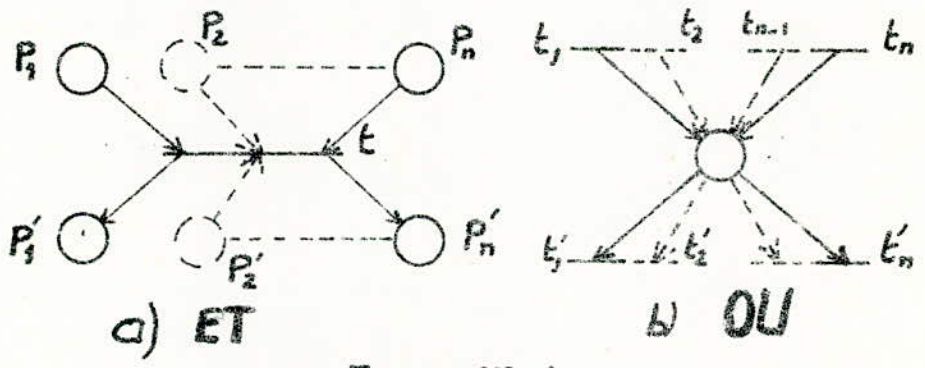


Figure III-1

Un « nœud OU » est une place qui est place de sortie de plusieurs transitions et/ou place d'entrée de plusieurs transitions (fig III-1.b).

III.3 Graphe d'état

On appelle graphe d'état un réseau de Pétri tel que la fonction de marquage ne place un marqueur que dans une place à la fois.

Un graphe d'état (GE) est donc un réseau où n'est représenté aucun fonctionnement parallèle.

Autrement dit un graphe d'état est un réseau de Pétri Sans "nœud ET." (figure III.2).

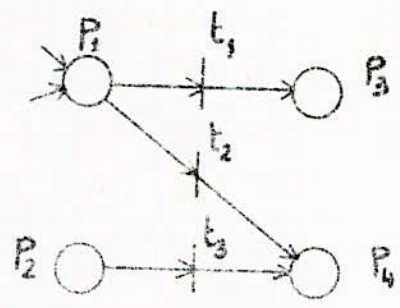


Figure III.2

Chaque transition a une seule place d'entrée et une seule place de sortie.

III.4 Décomposition en sous-graphes d'état

Nous allons chercher à décomposer un réseau de Pétri en sous-réseaux tels que chaque sous-réseau soit un graphe d'état (on parlera de décomposition en sous-graphes d'état).

III.4.1 Définition

Un sous-graphe d'état d'un réseau de Pétri est un ensemble de places et de transitions du réseau possédant les propriétés suivantes :

- a) Une seule place peut être marquée à la fois.
- b) Chaque transition ne peut avoir plus d'une place d'entrée et une place de sortie.
- c) L'ensemble est connecté : il est possible de trouver une chaîne (chemin non orienté) conduisant d'une place quelconque à une autre.

Dans un réseau de Pétri conforme, les sous-graphes d'état

s'obtiennent en supprimant les transitions qui ne peuvent pas appartenir à un sous-graphe (propriété b) et en recherchant les sous-ensembles interconnectés.

En effet, les sous-ensembles obtenus sont bien des sous-graphes d'état. Ils vérifient par construction les propriétés b et c). La propriété a) est également vérifiée puisque, si plusieurs places étaient marquées dans un sous-ensemble, on pourrait trouver une séquence de transitions plaçant un marqueur dans une des places déjà marquées, ce qui est contraire à la définition d'un réseau conforme.

Exemple de décomposition en sous-graphes (figure III.3.).

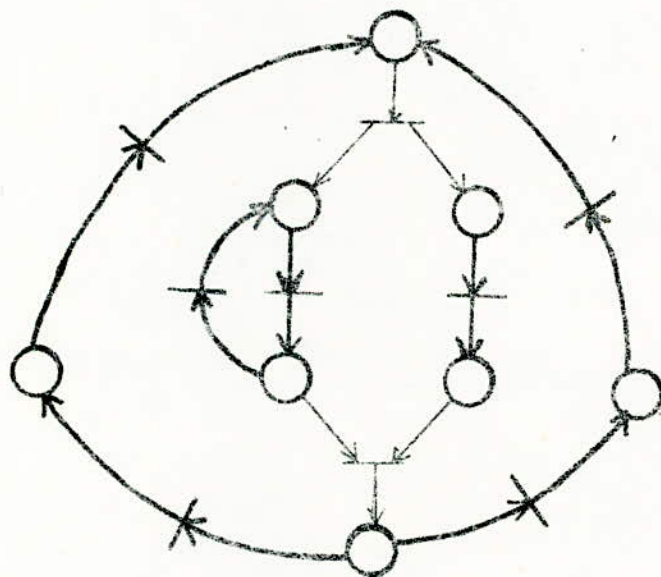


Figure III.3.

On met en évidence trois sous-graphes d'état (représentés en traits forts) (figure III.3.).

III.4.2 Hypothèses

On ne considère que des réseaux de Pétri (RdP) vérifiant :

H₁ - Une seule place initialement marquée ;

H₂ - Une seule place finale rebouclée sur la place initiale (éventuellement cette place peut être initiale elle-même).

Ces deux hypothèses ne sont pas restrictives, il est immédiat de modifier un RdP donné pour qu'il en soit ainsi.

H₃ - Le RdP ne présente ni nœuds puits, ni nœud source.

Cette hypothèse n'est pas fondamentale mais elle est nécessaire pour qu'un réseau vérifiant H₁ et H₂ si on veut qu'il soit vivant.

III.5. Macrographe

III.5.1 Constituants connexes

La décomposition du RdP commence par la coupure des ET». Cette opération consiste à supprimer dans le réseau tous les arcs entrant et sortant des transitions ET.

On obtient alors plusieurs constituants connexes sans nœud ET ; ce sont des graphes d'état (GE).

Afin d'obtenir des GTE définissons la notion de macroplace.

III.5.2 Macroplace

Soit G_h un GE obtenu du traitement précédent.
On effectue sur G_h une partition en (g_h¹, g_h², ... g_hⁿ) (C₁)

(C₁) Dans un sous-graphe g_h^i pour toute place d'entrée il existe au moins un chemin menant à chaque place de sortie.

On appelle macroplace P_i associée à un sous-graphe g_h^i une place ayant pour entrées toutes les entrées de g_h^i et pour sorties toutes ses sorties (fig. III.4).

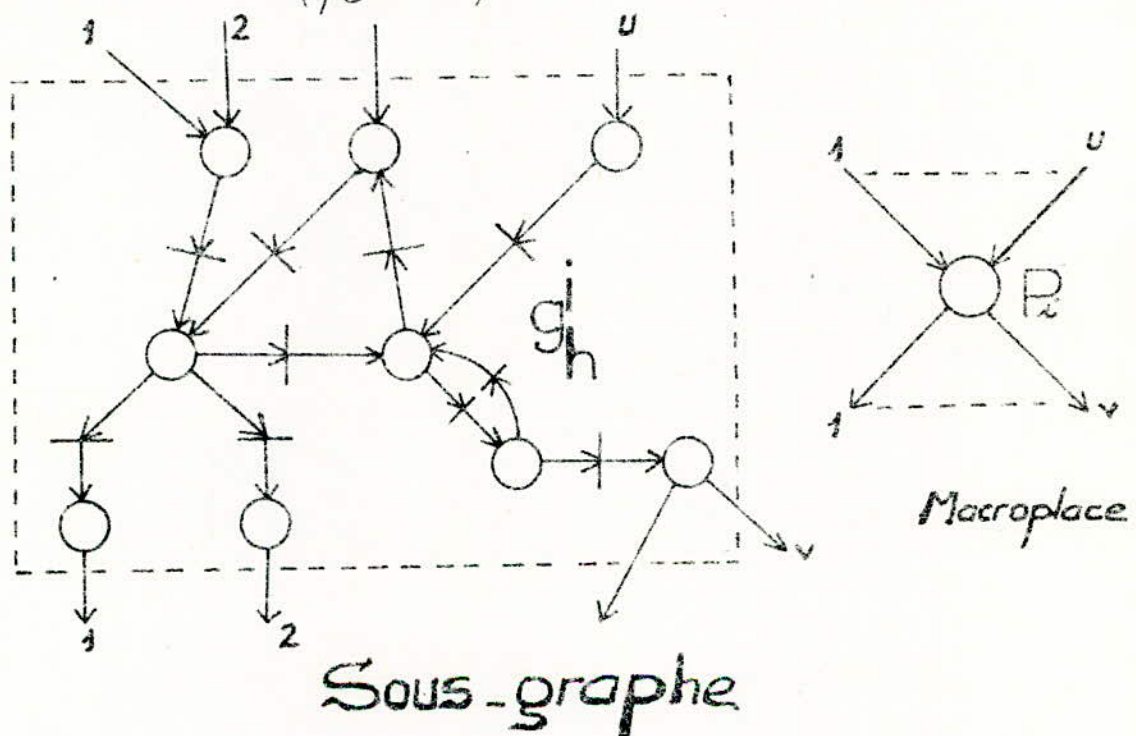


Figure III.4

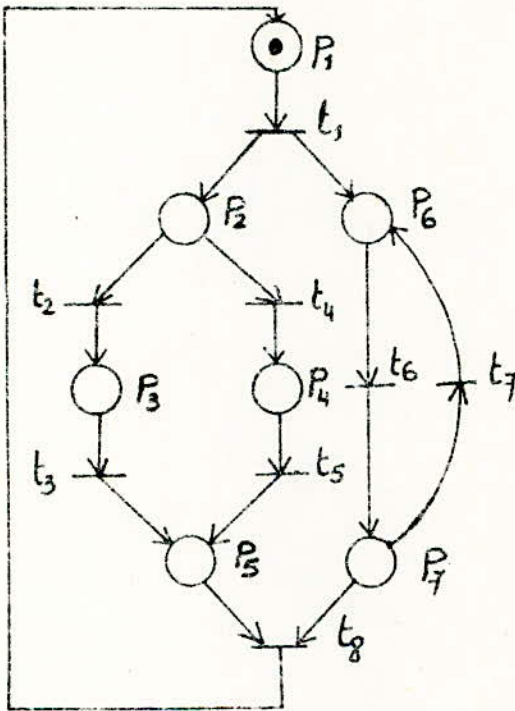
III-5.3 Macrographe associé à un Rdp

En remplaçant dans le Rdp initial tous les sous-graphes, obtenus par partition des parties connexes, par les macroplaces associées, on obtient un nouveau réseau de nœuds et d'étoiles appelé macrographe (MG) associé au Rdp.

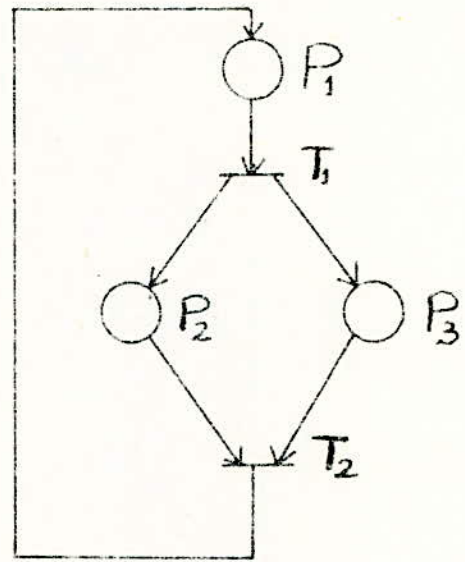
Ses nœuds sont les macroplaces notées P_1, P_2, \dots, P_n et les étoiles sont les transitions T_1, \dots, T_p .

- $\mathcal{P} = \{P_1, \dots, P_n\}$ = ensemble des macroplaces ;
- $\mathcal{T} = \{T_1, \dots, T_p\}$ = ensemble des transitions du macrographe ;
- \mathcal{ICT} ensemble des transitions du RdP initial.

Exemple



RdP initial



MG associé

Figure III-5

$$\mathcal{P} = \{P_1, P_2, P_3\}$$

$$\mathcal{T} = \{T_1, T_2\}$$

III-5-4 Marquage d'une macroplace

On dit qu'une macroplace possède m marques, quand le sous-graphe associé renferme m marques. La remarque (R_2), ci-dessous, justifie cette définition.

Les règles d'évolution du marquage d'un MG sont les mêmes que celles d'un RdP généralisé.

III-6 Propriétés du MG associé à un RdP

Il s'agit de reconnaître sur le MG des propriétés du RdP associé et inversement.

Compte tenu des hypothèses ($H_1 - H_2 - H_3$), des règles d'évolution des RdP généralisés et de la construction des macroplaces, il convient d'établir les remarques suivantes:

R_1 - Le MG d'un RdP vérifiant $H_1 - H_2 - H_3$ est un RdP généralisé, à une seule macroplace initialement marquée, et cyclique.

R_2 - Le nombre de marques est conservatif dans un GE (les marques ne peuvent ni être créées, ni disparaître à l'intérieur d'un GE).

R_3 - Dans un GE il est possible, à partir d'une place marquée, de faire évoluer la marque suivant n'importe quel chemin appartenant au GE et ayant pour origine la place considérée.

R_4 - Quand une marque pénètre dans une macroplace on peut la faire aller dans n'importe quelle place de sortie de la macroplace.

R_5 - Toute transition d'un sous-graphe associé à une macroplace peut être franchie s'il est possible d'entrer des marques par toutes les entrées de la macroplace et autant de fois que l'on veut.

Notons que ces propriétés sont purement structurelles:

Il ne faut pas tenir compte à ces niveaux des événements et des conditions.

À l'aide de ces remarques on établit les propriétés suivantes:

Propriété 1. $RdP \text{ sauf} \iff MG \text{ sauf}:$

1°. $MG \text{ sauf} \implies RdP \text{ sauf}$: s'établit en utilisant (R_2) ;

2°. $MG \text{ non sauf} \implies RdP \text{ non sauf}$: en faisant appel à R_3 et R_4

Propriété 2.

1°. $MG \text{ graphe marqué vivant} \implies RdP \text{ vivant}$: R_4 permet d'établir que tous les ET peuvent être franchis. R_1 et R_5 conduisent à la conclusion que toute transition du RdP peut être franchie;

2°. $RdP \text{ vivant} \implies MG \text{ vivant}$: par R_4 ,

d'où le théorème :

$$RdP \text{ conforme} \implies MG \text{ conforme}$$

Application

Le MG est un outil d'analyse des RdP. En effet, pour la synthèse nous voulons des RdP saufs et vivants (conformes). L'étude du MG associé nous permet de savoir si le RdP vérifie ces propriétés. Le MG a l'avantage d'être un graphe plus simple que le RdP original.

III.7 Graphes de transition d'états

III.7.1 Principe

Cette étape n'est abordée que si le RdP s'est avéré conforme. Donc chaque macroplace ne peut comporter qu'une marque

Simultanément : chaque sous-graphe associé à une macroplace est un GTE.

On peut recouvrir le Rdp par autant de GTE qu'il existe de macroplaces. Il est souvent intéressant de regrouper plusieurs GTE en un GTE unique, la seule condition à respecter est l'indépendance des macroplaces réunies.

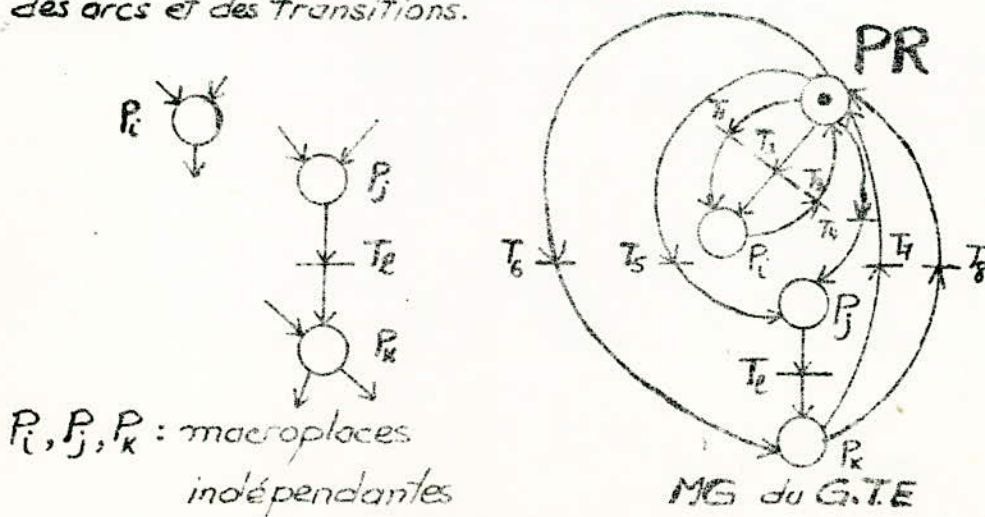
III.7.2 Construction d'un GTE

Les solutions retenues pour la réalisation conduisent à choisir des GTE fortement connexes et à un seul état initial.

L'existence d'un seul état initial conduit à ajouter au GTE une place appelée place de repos (PR) sauf dans le cas où le GTE contient la place initialement marquée : elle joue le rôle de PR. La connexité du graphe nécessite la création d'arcs et de transitions supplémentaires.

La place PR possède la propriété de pouvoir atteindre toutes les entrées des macroplaces et être accessible de toutes les sorties des macroplaces appartenant au GTE.

La figure III-6 montre sur un exemple la création de PR, des arcs et des transitions.



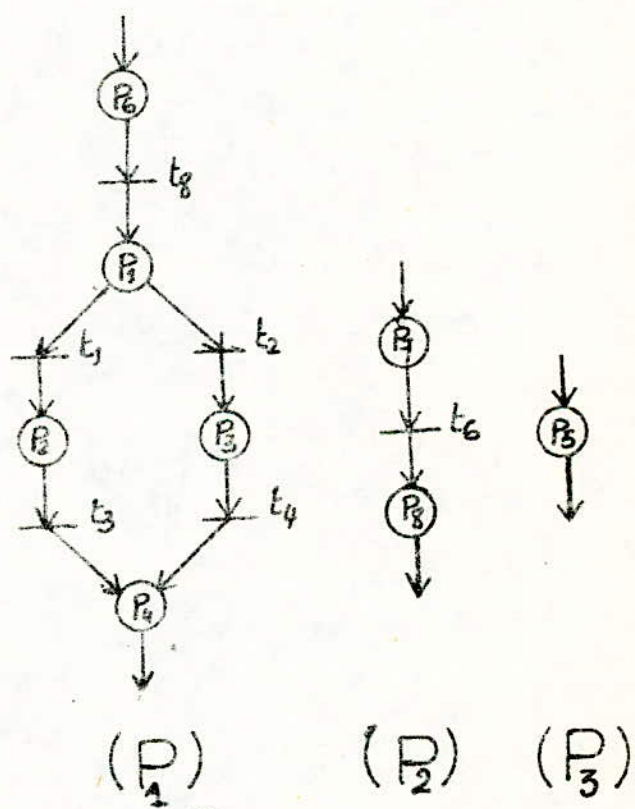
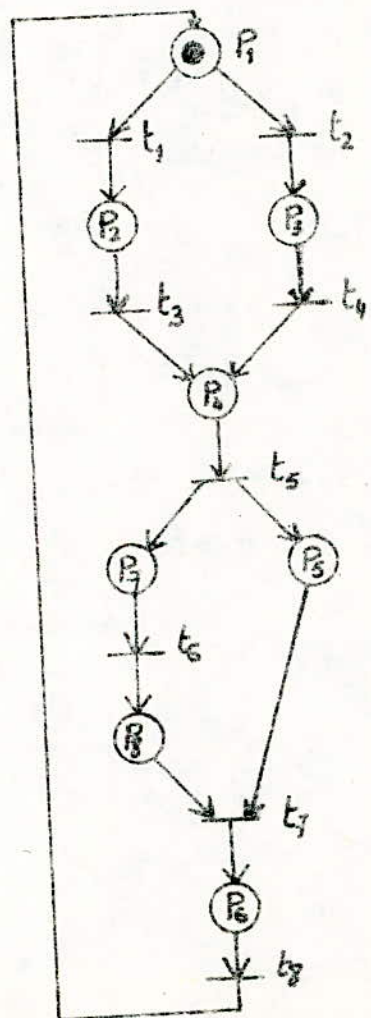
P_i, P_j, P_k : macroplaces indépendantes

Figure III-6

EXEMPLE

Sur le réseau de Pétri de la figure III-7-a, on détermine trois macroplaces P_1, P_2 et P_3 (représentées sur la figure III-7.b), ce qui conduit au macrographe de la figure III-7.c.

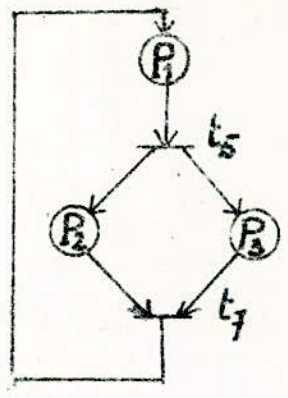
On peut recouvrir le Rdp initial avec deux graphes d'état: g_1 formé des macroplaces indépendantes P_1 et P_3 et g_2 formé de la macroplace P_2 (figure III-7.d).



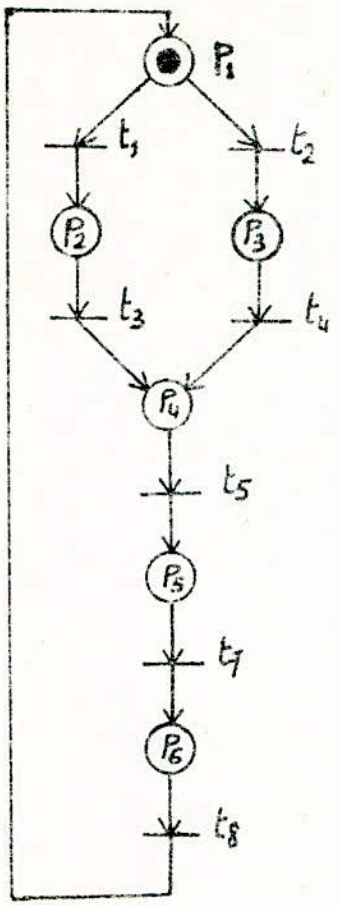
b. Macroplaces

a. Réseau de Pétri

Figure III-7

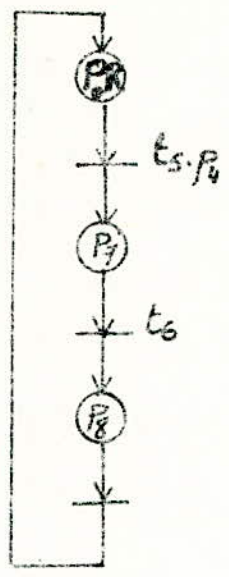


c. Macrographe



GTE₁

d. Graphes de transition d'états



GTE₂

Figure III-7

III.8 Conclusion

Nous avons obtenu une méthode de recouvrement des Rdp par des GTE qui permet de réaliser un système de commande parallèle, de type microprogrammé.

Pondération d'un réseau de PETRI

IV.1 Introduction

Nous avons envisagé une réalisation modulaire. Sa mise en œuvre nous a conduits à établir une propriété structurelle des Rdp : la pondération.

La pondération permet de définir une quantité invariante au cours de l'évolution du marquage.

IV.2 Définitions

IV.2.1 Définition de la pondération d'un réseau de PETRI

On appelle pondération d'un réseau de PETRI l'attribution à toute place p_i de ce réseau d'un nombre entier strictement positif p_i constant, indépendant du marquage, appelé poids de p_i tel que, pour toute transition t , la somme des poids des places d'entrée de t soit égale à la somme des poids des places de sortie de t .

Exemple 1 :

La figure IV.1 montre un réseau de PETRI. La définition ci-dessus conduit au système d'équations :

$$\text{transition a : } p_0 = p_1 + p_2,$$

$$\text{transition b : } p_2 = p_3 + p_4,$$

$$\text{transition c : } p_1 + p_3 = p_5,$$

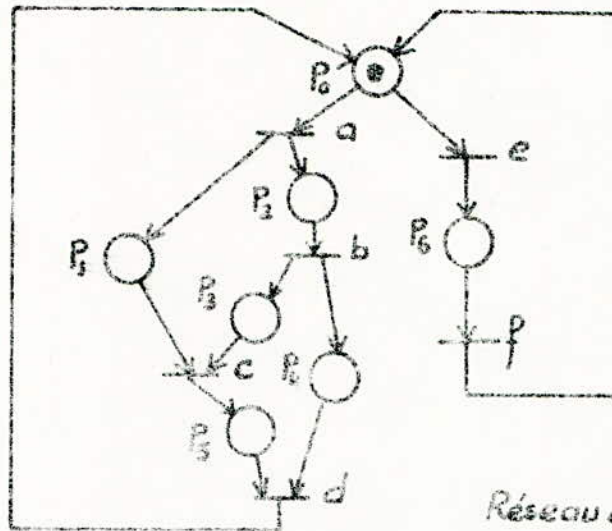
$$\text{transition d : } p_4 + p_5 = p_6,$$

$$\text{transition e : } p_0 = p_6$$

$$\text{transition f : } p_6 = p_0.$$

Ce système admet une infinité de solutions, par exemple :

$$p_0 = 3; \quad p_1 = p_3 = p_4 = 1; \quad p_2 = p_5 = 2; \quad p_6 = 3$$



Réseau de PÉTRI conforme

Figure IV.1

IV.2.2 Définition du poids total

On appelle poids total d'un réseau la quantité $\sum_{(\text{réseau})} M_i p_i$

où M_i est le nombre maximal de marqueurs que peut recevoir une place ($M_i = 1$ pour un réseau conforme).

IV.2.3 Définition du poids total marqué

On appelle poids total marqué d'un réseau la quantité $\sum_{(\text{réseau})} m_i p_i$

où m_i est le nombre de marqueurs de la place p_i (pour le réseau conforme $m_i = 0$ ou 1) et p_i son poids.

Dans l'exemple de la figure IV.1, le poids total est de 13, le poids total marqué est de 3. Soit $\sum M_i p_i = 1 \times 3 + 1 \times 1 + 1 \times 2 + 1 \times 1 + 1 \times 1 + 1 \times 2 + 1 \times 3 = 13$ et $\sum m_i p_i = 1 \times 3 = 3$.

IV.3 propriété fondamentale

Dans un réseau pondéré le poids total marqué reste constant à partir du marquage initial pour toute séquence de franchissements de transitions. En effet, un franchissement enlève exactement une marque de chaque place d'entrée de la transition franchie et met une marque dans chaque place de sortie. La somme des poids perdus en entrée étant égale à la somme des poids gagnés en sortie, la somme totale ne varie pas.

IV.4 Expressions matricielles de la pondération

IV.4-1 Vecteur de pondération des places :

Les composantes de ce vecteur sont les poids des places.

IV.4-2 Vecteur de pondération des transitions :

$$[T] = {}^t[C] \times [P]$$

où ${}^t[C]$ est la transposée de la matrice d'incidence.

Les composantes de $[T]$ sont égales à la différence de la somme des poids des places d'entrée et de la somme des poids des places de sortie de chaque transition.

Pondérer un réseau c'est chercher un vecteur $[P]$ qui annule le vecteur $[T]$ donc chercher les solutions entières positives de l'équation matricielle :

$${}^t[C] \times [P] = [0].$$

Nous proposons ci-après deux méthodes de pondération :

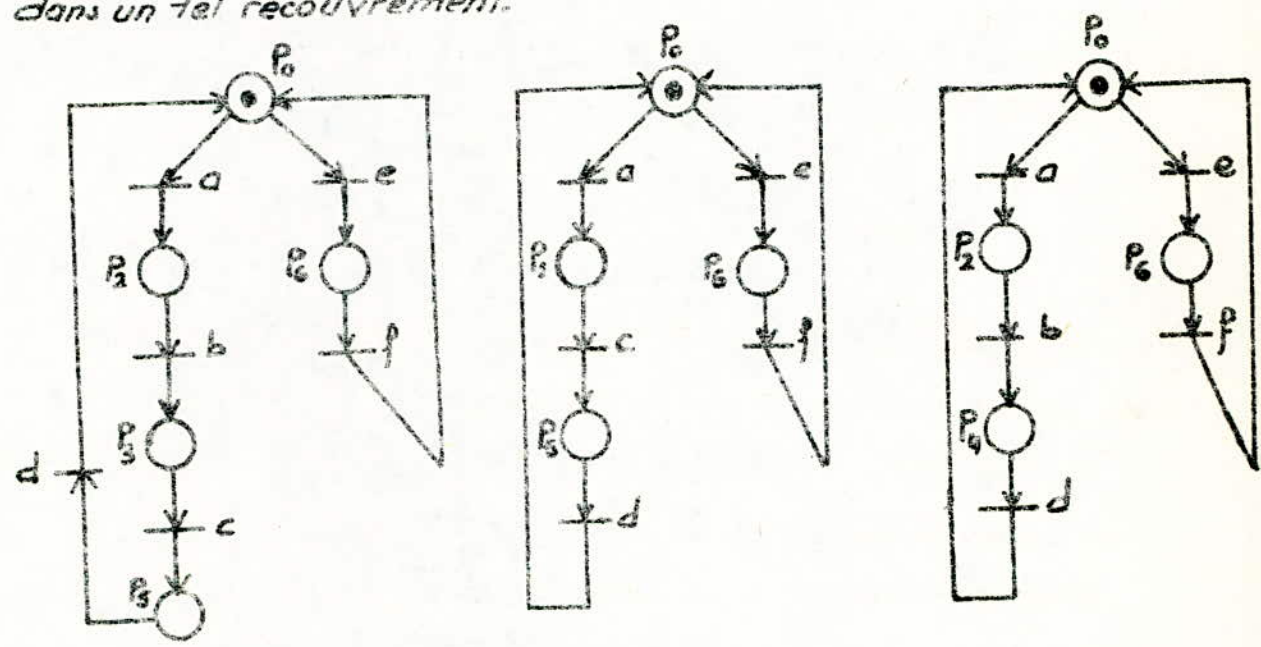
— La première ne s'applique qu'aux réseaux de PETRI libre-choix conformes, elle a essentiellement un intérêt théorique car elle montre les liens qui existent entre la pondération et la théorie de Hack.

— La deuxième méthode s'applique à tous les réseaux de PETRI vivants et à marquage fini. Elle est d'application plus aisée, mais présente

l'inconvénient d'imposer parfois une modification du réseau.

IV.5 Pondération des réseaux de PETRI libre. choix conformes

La méthode découle immédiatement du théorème de Hock. Elle consiste à appliquer l'algorithme de recouvrement par des graphes d'état fortement connexes, puis à pondérer le réseau en attribuant à toute place un poids égal au nombre de graphes d'état auxquels elle appartient dans un tel recouvrement.



Pondération de l'exemple 1 par décomposition en graphes d'état.

Figure IV.2

La figure IV.2 montre le recouvrement du graphe de l'exemple 1 par trois graphes d'état. La place p_0 reçoit un poids égal à 3 car elle appartient aux trois graphes d'état; on obtient de même $p_5 = 3$; $p_1 = p_3 = p_6 = 1$; $p_2 = p_5 = 2$;

Le poids total du réseau est égal à 13

Le poids total marqué est égal à 3.

Nous présentons dans le paragraphe suivant une méthode d'application plus simple et générale, à condition d'accepter de modifier

le réseau. Comme la pondération de certains réseaux complexes est difficile à obtenir directement, nous avons introduit, pour simplifier la pondération dans ^{tous} les cas, une possibilité de modifier le réseau en dédoublant certaines places afin d'obtenir un réseau équivalent aisément pondérable.

IV.6 Pondération d'un réseau de PETRI quelconque vivant et à marquage fini

Nous appelons cette méthode : méthode des chemins fictifs. Nous envisageons le cas le plus simple où le réseau est un graphe marqué; puis nous présentons le cas général.

IV.6.1 Cas d'un graphe marqué

On suppose que le graphe a seulement une place initialement marquée p_0 (Sinon se reporter au cas général ci-après). On commence par couper le graphe afin d'obtenir un graphe acyclique. Pour cela on coupe toutes les places qui ont plus d'un arc incident ou plus d'un arc sortant, et toutes les places initialement marquées conformément à la définition ci-dessous :

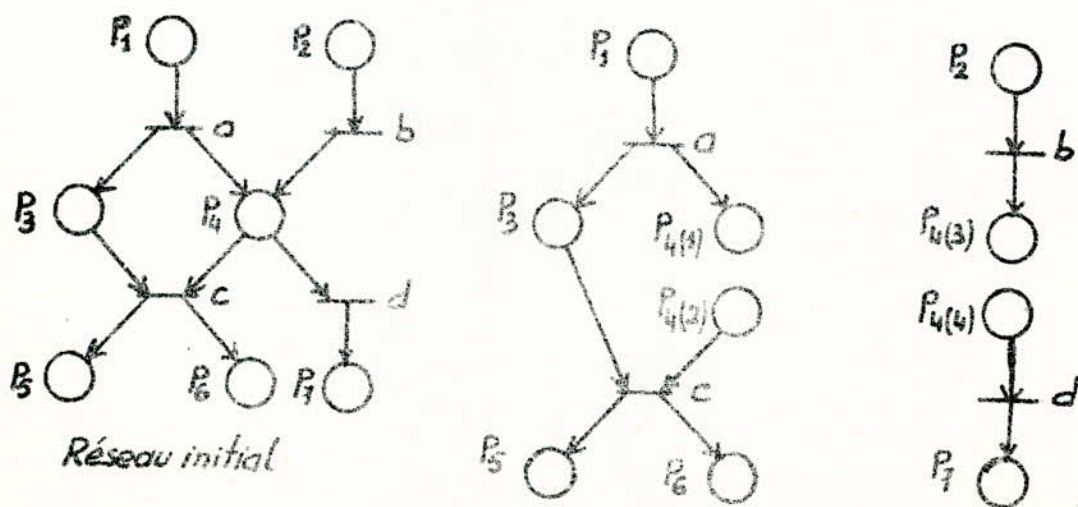


Figure IV.3

Réseau coupé

Couper une place p_i d'un réseau de PETRI c'est la remplacer par autant de places (p_{i_1}, \dots, p_{i_e}) qu'il y a d'arcs reliés à p_i dans le réseau de PETRI initial, et associer chacune des places obtenues à un de ces arcs (fig IV.3).

IV.6.2. Algorithme de pondération d'un graphe marqué acyclique

Le réseau coupé obtenu peut être pondéré par application de l'algorithme suivant :

Phase 1: Attribuer à toute place du réseau le poids 1.

Donc le vecteur de pondération est

$$[P_i] = \begin{pmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}.$$

Phase 2: Calculer $[T_i] = -[C] \times [P_i]$; si $[T_i] = [0]$ le réseau est pondéré par $[P_i]$; sinon : aller en phase trois (3).

Phase 3: Décomposer $[T_i]$ en parties positive et négative :

$$[T_i] = [T_i^+] - [T_i^-].$$

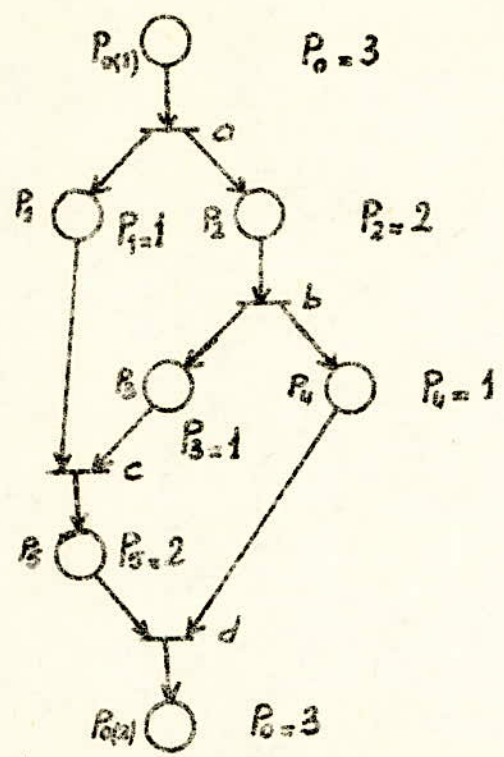
Extraire une matrice $[C_e]$ de la matrice $[C]$ en ne conservant dans chaque colonne qu'une case +1 et une case -1 (ou aucune s'il n'y en a pas dans la matrice $[C]$). On retiendra systématiquement par exemple le premier terme rencontré dans une exploration des colonnes.

On peut décomposer $[C_e]$: $[C_e] = [C_e^+] - [C_e^-]$.

Calculer alors $[P_{i+1}] = [P_i] + [C_e^-] \times [T_i^-] + [C_e^+] \times [T_i^+]$
puis revenir à la phase 2.

Exemple :

La figure IV-4 montre un graphe marqué à une place initialement marquée P_0 que l'on a coupée en $P_0(1)$ et $P_0(2)$.



Pondération d'un graphe marqué

Figure IV-4

Matrice d'incidence du réseau:

	a	b	c	d
$P_0(1)$	-1	0	0	0
P_1	+1	0	-1	0
P_2	+1	-1	0	0
P_3	0	+1	-1	0
P_4	0	+1	0	-1
P_5	0	0	+1	-1
$P_0(2)$	0	0	0	+1

Phase 1:

$$[R] = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Phase 2:

$$[T_1] = - \begin{bmatrix} -1 & +1 & +1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & +1 & +1 & 0 & 0 \\ 0 & -1 & 0 & -1 & 0 & +1 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & +1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ +1 \\ +1 \end{bmatrix}$$

$$[T_1] \neq [0]$$

Phase 3:

$$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

$$[R] = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Retourner en phase 2 :

$$[T_2] = - \begin{vmatrix} -1 & +1 & +1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & +1 & +1 & 0 & 0 \\ 0 & -1 & 0 & -1 & 0 & +1 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & +1 \end{vmatrix} \times \begin{vmatrix} 2 \\ 1 \\ 2 \\ 1 \\ 1 \\ 2 \\ 2 \end{vmatrix} = \begin{vmatrix} -1 \\ 0 \\ 0 \\ +1 \end{vmatrix},$$

$$[T_2] \neq [0].$$

Phase 3.

On calcule $[P_3] = [P_2] + [C_e^-] \times [T_2^-] + [C_e^+] \times [T_2^+]$

$$\begin{vmatrix} 2 \\ 1 \\ 2 \\ 1 \\ 1 \\ 2 \\ 2 \end{vmatrix} + \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{vmatrix} \times \begin{vmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{vmatrix} + \begin{vmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \times \begin{vmatrix} 0 \\ 0 \\ 0 \\ 1 \end{vmatrix},$$

$$[P_3] = \begin{vmatrix} 3 \\ 1 \\ 2 \\ 1 \\ 1 \\ 2 \\ 3 \end{vmatrix}$$

Retour en phase 2 :

$$[T_3] = - \begin{vmatrix} -1 & +1 & +1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & +1 & +1 & 0 & 0 \\ 0 & -1 & 0 & -1 & 0 & +1 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & +1 \end{vmatrix} \times \begin{vmatrix} 3 \\ 1 \\ 2 \\ 1 \\ 1 \\ 2 \\ 3 \end{vmatrix} = \begin{vmatrix} 0 \\ 0 \\ 0 \\ 0 \end{vmatrix}.$$

$$[T_3] = [0].$$

Donc le réseau est pondéré par $[P_3]$.

$$[P_3] = \begin{array}{|l} P_0(1) = 3 \\ P_1 = 1 \\ P_2 = 2 \\ P_3 = 1 \\ P_4 = 1 \\ P_5 = 2 \\ P_0(2) = 3 \end{array} ,$$

Le poids total marqué est $\sum m_i p_i = 3$. Le poids total du réseau est $\sum M_i P_i = 10$.

IV. 6.3 Cas général

Pour étendre la méthode précédente au cas général, nous coupons le graphe afin d'obtenir des graphes marqués acycliques. Il apparaît alors une difficulté : Une même place risque de recevoir plusieurs poids distincts. Nous montrons qu'il est possible de pondérer le réseau si on accepte d'attribuer à certaines places un poids nul, en procédant par diminutions successives de poids ; puis nous modifions le réseau en créant des places supplémentaires pour qu'il n'y ait plus de poids nul.

Pour décomposer le réseau de PETRI en graphes marqués acycliques, il suffit de couper les places qui sont reliées à plusieurs arcs en entrée ou en sortie, ainsi que toutes les places initialement marquées.

Il est alors possible de pondérer séparément chaque graphe marqué obtenu par la méthode décrite ci-dessus (IV. 6.1), mais il apparaît une difficulté pour revenir au réseau initial : les places P_{i1}, \dots, P_{il} créées par coupure d'une même place P_i du réseau initial risque d'avoir des poids distincts. On choisit alors le plus petit de ces poids, ce qui amène à diminuer les poids d'autres places.

La méthode de diminution des poids risque d'aboutir à des poids nuls, contraire à la définition donnée précédemment.

Montrons qu'il est équivalent d'attribuer un poids nul à une place P_i d'un réseau de PETRI ou de modifier le réseau par dédoublement de la place P_i conformément à la définition suivante :

Dédoublet une place P_i , c'est lui associer une place \bar{P}_i telle que la somme des nombres de marques dans P_i et \bar{P}_i soit égale à $M_i = m_i + \bar{m}_i$ et que les poids soient égaux : $p_i = \bar{p}_i$. La figure IV-5 montre comment on dédouble une place P_i : les transitions d'entrée de P_i sont les transitions de sortie de \bar{P}_i et inversement. Avant dédoublement, les équations de pondération étaient :

$$(a) \quad p_1 + p_2 = p_3 + p_i ;$$

$$(b) \quad p_4 + p_i = p_5 + p_6 ;$$

après dédoublement, le système devient :

$$(a) \quad p_1 + p_2 + \bar{p}_i = p_3 + p_i \quad p_1 + p_2 = p_3 ;$$

$$(b) \quad p_4 + p_i = p_5 + p_6 + \bar{p}_i \quad p_4 = p_5 + p_6 .$$

Ce dernier système est identique au premier si on fait $p_i = 0$. Il est donc possible d'attribuer un poids 0 à une place d'un réseau de PETRI, à condition de dédoublet cette place, d'où la méthode proposée.

La méthode de pondération décrite ci-dessous procède par diminutions successives des poids. Elle converge nécessairement car elle est bornée par une solution toujours possible qui consiste à attribuer le poids 0 à toutes les places dans le cas le plus défavorable.

Enoncé de la méthode

Phase 1 : Décomposer le graphe en graphes marqués acycliques

Comme indiqué plus haut et pondérer les graphes marqués obtenus par la méthode du paragraphe IV.6.1

Phase 2:

Si les poids obtenus par les places coupées sont égaux, aller en phase 5 sinon aller en phase 3.

Phase 3:

Soit une place $P_{i,j}$ qui reçoit un poids supérieur à une place $P_{i,k}$ de même origine dans le réseau initial. Diminuer d'une unité le poids de $P_{i,j}$.

Phase 4:

Puis extraire une matrice $[C_2]$ de la matrice d'incidence $[C]$ en retenant dans chaque colonne une seule case $+1$ et une seule case -1 : On retiendra les cases qui correspondent aux places de poids les plus élevés. On ne retiendra pas de case correspondant à des places de poids nul.

Calculer $[P_{i+1}] = [P_i] - [C_2^+] [T_i^-]$ dans le cas où $P_{i,j}$ est une place d'entrée, ou $[P_{i+1}] = [P_i] - [C_2^-] [T_i^+]$ dans le cas où $P_{i,j}$ est une place de sortie. Calculer à nouveau $[T_{i+1}]$. Si $[T_{i+1}] = [0]$ revenir phase 2, sinon recommencer la phase 4.

Phase 5:

Les places reçoivent un poids positif ou nul. Toute place recevant un poids nul est dédoublée.

Phase 6 (facultative)

On peut réduire parfois le nombre des places supplémentaires créées par dédoublement de la façon suivante: si, dans un sous-réseau connexe d'un graphe marqué obtenu en phase 1 toutes les places reçoivent

un poids 0, on peut remplacer les places créées par dédoublement par une seule place dans les conditions suivantes :

1°. Les places de ce sous-réseau appartiennent à des chemins issus d'une même transition (a) et aboutissant à une même transition (b).

2°. S'il existe dans ce sous-réseau un chemin qui n'est pas issu de (a), alors il n'y a pas de chemin qui n'aboutisse pas en (b); inversement, s'il existe un chemin qui sort du sous-réseau sans passer par (b), il n'y a pas de chemin qui rentre sans être issu de (a). (Ces conditions sont suffisantes et non nécessaires pour permettre la réduction, mais une étude générale serait très longue.)

Si les deux conditions sont vérifiées, on associe au sous-réseau considéré une place supplémentaire unique ayant (b) comme transition d'entrée et (a) comme transition de sortie. Cette place doit être initialement marquée.

L'ensemble formé par le sous-réseau et la place supplémentaire est pondéré par des chemins fictifs, on est ramené au cas simplifié d'un graphe marqué à une seule place initiale.

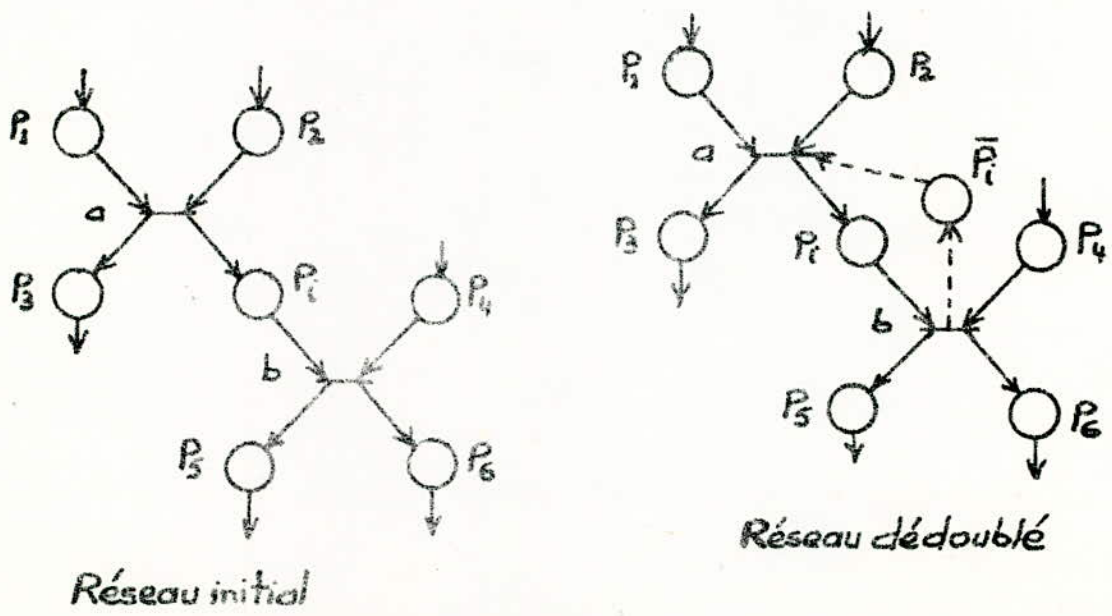


Figure IV.5

Exemple :

Nous reprenons l'exemple de la figure IV.1 et nous le pondérons par la méthode des chemins fictifs (figure IV.6).

Il apparaît deux graphes marqués (phase 1), la place P_0 est coupée en quatre places. La pondération des deux blocs donne à $P_0(1)$ et $P_0(2)$ le poids 3 et à $P_0(3)$ et $P_0(4)$ le poids 1 (phase 2).

Les poids reçus par P_0 étant distincts, on diminue deux fois le poids de $P_0(1)$. On obtient alors

$$P_0(1) = P_0(2) = P_0(3) = P_0(4) = P_2 = P_4 = P_6 = 1 ; \quad P_1 = P_3 = P_5 = 0.$$

Dans la phase 5 on constate que $P_1 = P_3 = P_5 = 0$. Il y a trois places dédoublées. On peut réduire deux de ces places dédoublées en une seule (phase 6), soit par exemple P_3 et P_5 car les places P_3, P_5 appartiennent à un chemin issu de la transition (a) et conduisant à la transition (d).

Il y a un chemin qui passe par P_6 et n'est pas issu de (a) : celui qui vient de P_3 ; mais il n'y a pas de chemin qui sorte du sous-réseau considéré et qui ne soit pas incident à (d), la réduction est donc possible.

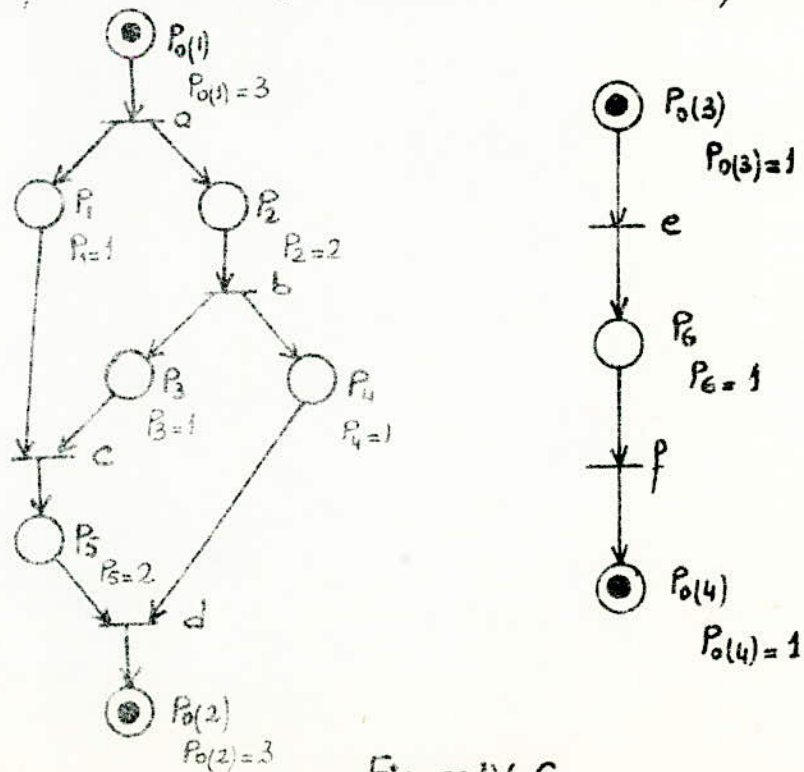


Figure IV.6

On crée une place $\bar{P}_{1,5}$ qui remplace l'ensemble \bar{P}_1, \bar{P}_5 et qui a pour transition d'entrée (d) et de sortie (a); chacune des places $P_1, P_5, \bar{P}_{1,5}$ reçoit un poids 1.

Finalement, la méthode a conduit à créer deux places supplémentaires: $\bar{P}_{1,5}$ et \bar{P}_3 (figure IV.7).

Le poids total du réseau est égal à $\sum M_i P_i = 9$. Le poids total marqué est: $\sum m_i p_i = 3$.

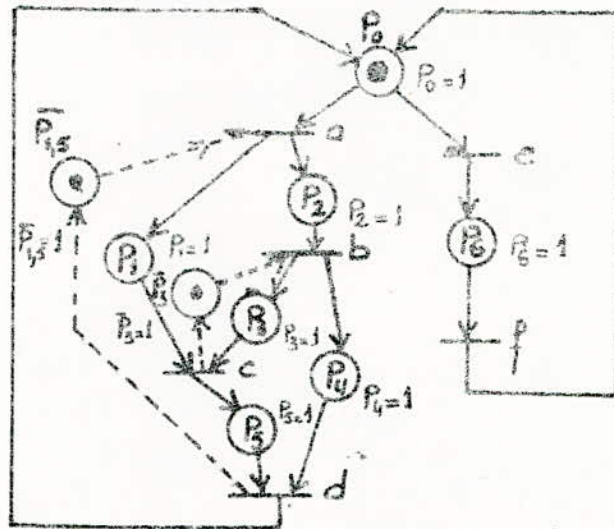


Figure IV.7

IV.7. Conclusion

Nous avons proposé deux méthodes algorithmiques de pondération des réseaux de PETRI conformes: la première qui ne s'applique qu'aux Rdp libre choix ne modifie pas le réseau. La seconde qui est plus générale et plus simple, risque de modifier le réseau par adjonction de places supplémentaires.

Le poids total marqué constant obtenu par pondération peut jouer le rôle du nombre K dans le code détecteur d'erreurs " K parmi n ". (En effet: en fonctionnement normal, le poids total marqué est un invariant dont on peut contrôler la constance; on peut faire une réalisation pour laquelle toute faute conduise à une variation définitive du poids total marqué pour permettre la détection des pannes et la sûreté de fonctionnement).

Annexe Compléments sur les réseaux de PETRI

A.1. Extensions des réseaux de PETRI

De nombreuses extensions des réseaux de PETRI ont été proposées. Nous allons citer quelques unes.

A.1.a. Réseaux de PETRI généralisés

Ils sont définis en affectant à chaque arc une capacité (ou poids) C correspondant au nombre de marqueurs qu'il peut transférer.

Une transition t ne sera validée que si chacune de ses places d'entrée possède au moins C marqueurs (C étant la capacité de l'arc reliant la place correspondante à la transition t). Lors du franchissement de la transition, un arc de capacité C enlèvera (placera) C marqueurs à la place correspondante d'entrée (de sortie).

Sur la figure A-1, on a représenté les notations des réseaux de PETRI généralisés (Figure A.1.a) et le franchissement d'une transition (Figures A.1.b et A.1.c).

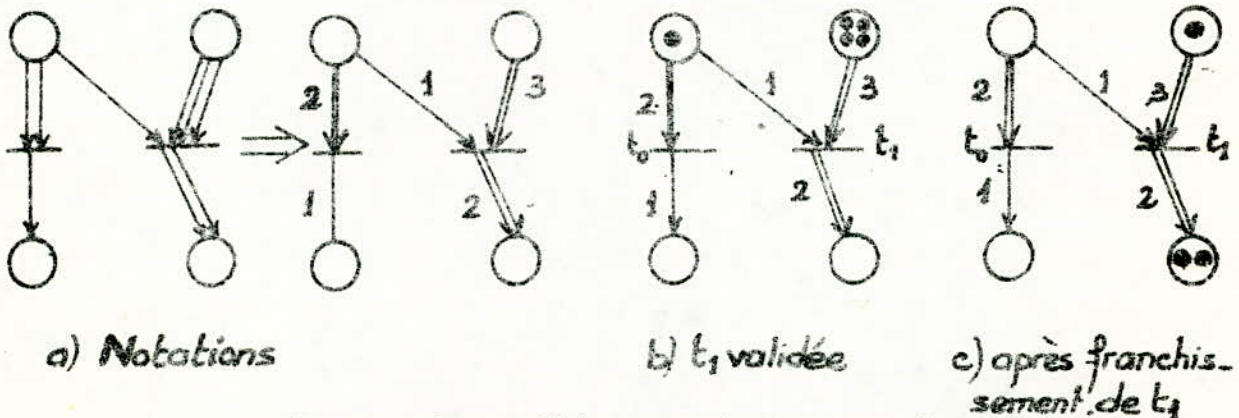


Figure A.1. Réseaux de PETRI généralisés

Flacke a montré que ces réseaux sont équivalents aux réseaux de PETRI ordinaires.

A.1.b. Arcs complémentés

Cette notation a été introduite pour permettre le "test à zéro". Tout comme on a affecté une capacité à un arc, on va à présent ajouter la notion de complémentation. Dans ce cas, si une place d'entrée p d'une transition t est reliée à cette transition par un arc complémenté (on dit aussi "arc inhibiteur"), il est nécessaire pour que t soit franchissable que la place p soit vide.

Un exemple est donné figure A.2 (l'arc $p_2 t_1$ est complémenté). La place p_3 est nécessaire pour éviter que la transition t_1 soit validée en permanence lorsque p_2 est vide.

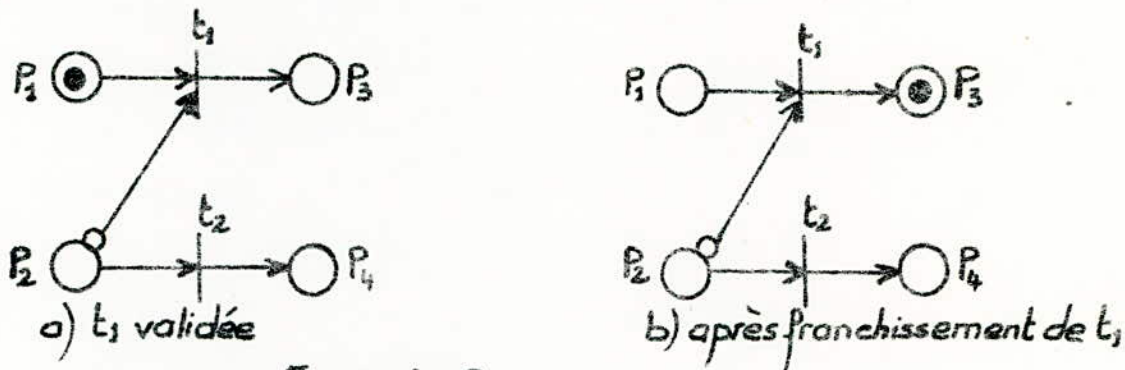


Figure A.2

A.1.c. Réseau libre choix

Chaque arc reliant une place à une transition est soit le seul arc issu de la place, soit le seul arc arrivant sur la transition (Figure A.3).

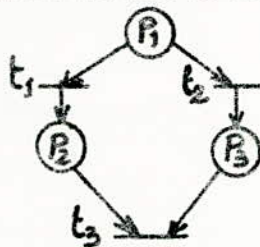


Figure A.3

Théorème de Hack :

Tout réseau de PETRI libre choix conforme peut être recouvert par des graphes d'état fortement connexes à une seule marque.

Application des réseaux de PETRI à la Synthèse des machines séquentielles

V.1 Réseau de PETRI et machines séquentielles

V.1.1 Introduction

Nous en savons assez sur la formalisation des réseaux de PETRI pour aborder, maintenant, leur application à la synthèse de machines séquentielles.

Qu'est-ce que faire la synthèse d'une machine séquentielle ? C'est établir une coordination entre des événements (les entrées) et des conditions (les sorties). De façon générale, dans un système séquentiel, la réalisation d'une condition C débute par l'occurrence d'un événement e_1 , et prend fin par l'occurrence d'un événement e_2 .

Il nous a semblé nécessaire de définir qu'est-ce qu'un cahier des charges d'une machine séquentielle:

Le cahier des charges d'un «automatisme logique» est un document résumant les rapports entre le fournisseur concepteur d'un matériel de commande et son client, utilisateur futur de ce matériel.

Qu'on l'appelle cahier des charges, desiderata du client ou spécifications techniques, ce document - le plus en amont du processus de synthèse - est à la base de la conception d'un système logique.

Prenez un exemple très simple, mais qui donne une bonne idée des problèmes rencontrés dans la conception d'automatismes industriels (ne serait-ce que la présence de ce bouton poussoir M qu'on doit manipuler impunément pendant tout le cycle).

Exemple élémentaire: aller-retour d'un chariot

Cahier des charges

On considère un dispositif tel que celui qui est schématisé sur la figure V.1. Y et Z sont deux moteurs alimentés par des relais

de même nom. Quand Y est alimenté, le chariot C se déplace de la droite vers la gauche ; quand Z est alimenté, le chariot C se déplace de la gauche vers la droite. A et B sont des relais de fin de course. On désire construire une machine séquentielle telle que, si

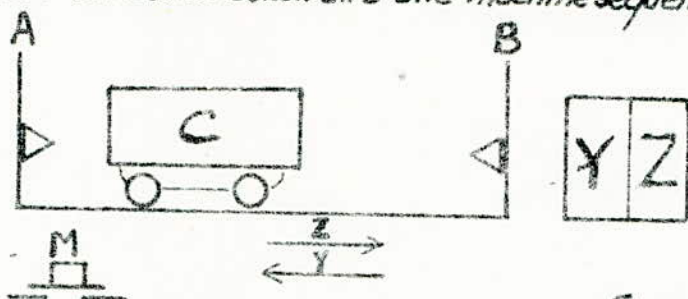


Figure V-1

l'on appuie sur le bouton de mise en marche M lorsque le chariot est au repos en A, ce dernier quitte A, arrive en B et revient en A où il s'arrête. Une fois que le chariot est en déplacement, on s'interdit d'appuyer de nouveau sur M.

Sur la figure V-2-a, on a "représenté" la condition "marche à gauche" du chariot. Ce graphisme laisse apparaître la manière dont nous allons utiliser les réseaux de PETRI pour la synthèse des machines séquentielles :

- les événements (entrées) seront associés aux transitions,
- les conditions (sorties) seront associées aux places,
- les successions d'événements et d'états seront associés aux arcs.

condition: le chariot est en marche à gauche

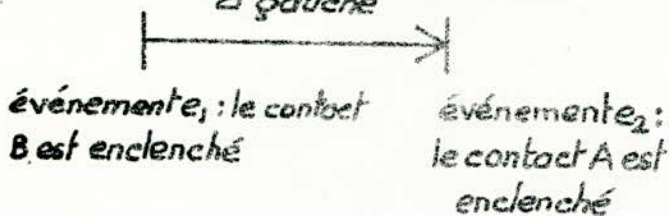


Figure V-2-a

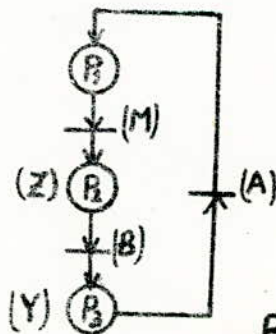


Figure V-2-b

Le fonctionnement du chariot peut alors être représenté par le réseau de PETRI de la figure V.2.b.

À partir de cette première démarche intuitive, il convient maintenant d'ordonner un peu les choses. En effet, il est clair que, si probablement toute machine séquentielle est représentable par un réseau de PETRI, tout réseau n'est pas forcément un automatisme séquentiel. Nous allons donc préciser un certain nombre de notions utiles.

V.1.2 Définitions. Notations

Nous allons commencer par considérer les formes possibles de réseaux de PETRI (RdP) et définir plus précisément les règles du franchissement des transitions pour les machines séquentielles. Nous verrons ensuite la correspondance qui peut exister entre le RdP et la table d'une machine séquentielle.

V.1.2.a Structures de réseau

De façon générale, un réseau de PETRI, comportant les quatre configurations C_1, C_2, C_3 et C_4 de la figure V.3, permet de représenter le fonctionnement d'un automatisme séquentiel. Il contient l'ensemble des informations suffisantes pour sa synthèse.

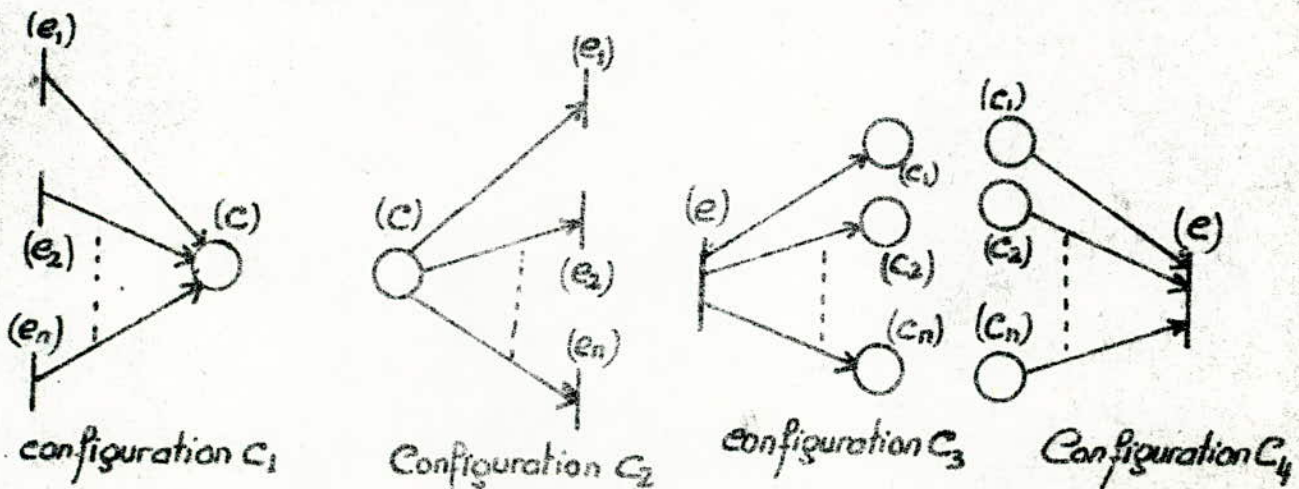


Figure V.3

Une remarque fondamentale S'impose :

Le graphe d'état d'un système séquentiel ne peut comporter les configurations C_3 et C_4 , dans la mesure où une seule place peut être marquée à la fois (la notion d'état est associée à la place). Au contraire, dans un réseau de PETRI, c'est le marquage (ensemble de places marquées) qui représente l'état du système (le graphe d'état correspondant serait donné par le graphe des marquages).

V.1.2-b Transitions

En outre, si nous nous intéressons à une portion du réseau comme celle de la figure V-4, nous voyons que :

- la condition C sera réalisée tant que la place p sera marquée ;
- la transition t_2 pourra être franchie lorsqu'elle sera validée (ici p marquée) et que l'événement associé e_2 sera vrai.

Exemple :

La place p_1 du graphe de la figure V-5-a est marquée, la condition (C_1) est réalisée. Si l'événement (e) se produit soit (e) = 1 la transition t peut être franchie. Les places p_2 et p_3 sont alors marquées et les conditions (C_2) et (C_3) prennent les valeurs (C_2) = 1 et (C_3) = 1 (figure V-5-b).

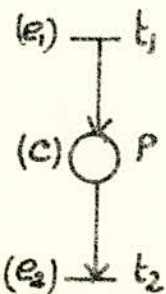


Figure V-4

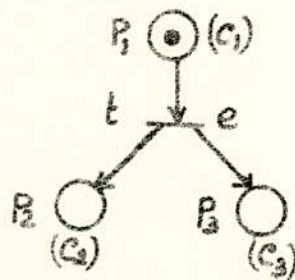


Figure V-5-a

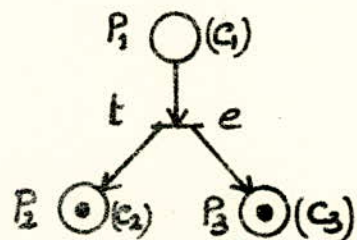


Figure V-5-b

A tout marquage du réseau est associé un état de la machine séquentielle.

V.1.2.C Notion d'alphabet et d'événement

On appelle alphabet d'entrée (sortie) l'ensemble de toutes les configurations possibles des entrées (sorties) exprimées sous forme littérale.

Il importe de définir la notion d'événements dont nous avons parlé à plusieurs reprises.

Si une machine séquentielle a n entrées physiques $\{x_1, x_2, \dots, x_n\}$, son alphabet d'entrée X comportera 2^n termes $\{x_0, x_1, \dots, x_{2^n-1}\}$.

Nous conviendrons d'appeler :

- événement tout sous-ensemble de X
- événement certain " λ " l'ensemble X
- événement impossible " Φ " l'ensemble vide
- événement élémentaire tout élément x_i de X .

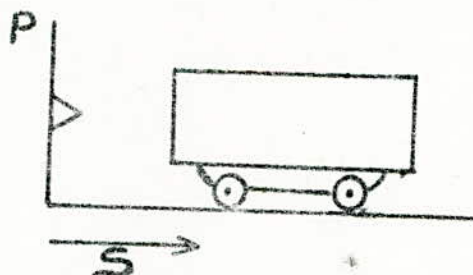
Ajoutons deux définitions :

- deux événements e_1 et e_2 sont exclusifs explicitement ($e_1 \cap e_2 = \emptyset$) ou implicitement (incompatibilités physiques), s'ils ne peuvent être vrais en même temps.

- deux événements e_1 et e_2 sont indépendants s'il n'existe aucune relation entre l'occurrence de l'un et celle de l'autre.

Notion d'incompatible physique

Soit une entrée primaire correspondant au signal que délivre un contact P (de fin de course par exemple), l'ouverture et la fermeture de ce contact dépendant d'un ordre S de commande d'un matériel.

Exemple:

Lorsque $s=1$, le contact P s'ouvre, impliquant $P.s=0$. Cette relation d'incompatibilité qui lie P et s doit être une donnée du cahier des charges. On appellera le couple P, s incompatible physique.

V.1.2.d Conflits

Plusieurs transitions qui partagent une ou plusieurs transitions sont dites en conflit si elles sont simultanément franchissables.

Dans un réseau conforme, comme nous ne savons pas partager un marqueur et que nous ne possédons aucun élément pour régler ce conflit en franchissant une transition plutôt qu'une autre, nous chercherons à éliminer les conflits, c'est-à-dire à obtenir des réseaux déterministes.

Nous nous assurerons, lorsqu'il y a partage de place(s) que, parmi les transitions validées, il n'existe pas plusieurs transitions simultanément franchissables, c'est-à-dire que les événements provoquant leur franchissement sont exclusifs.

Nous pouvons affirmer, à présent, que toute machine séquentielle est représentable par un réseau de PETRI conforme et sans conflit et nous ne travaillerons, par la suite, que sur de tels réseaux.

Exemple:

Le réseau de PETRI conforme de la figure V.7 résume le fonctionnement de l'ensemble processus P , structure de commande S bouclé donné par la figure V.6.

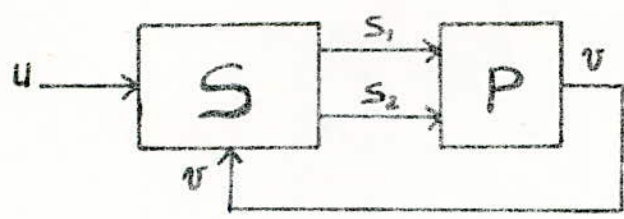


Figure V.6

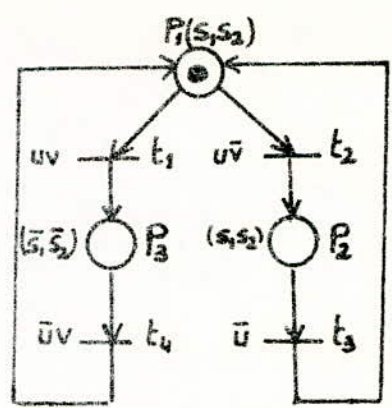


Figure V.7

L'alphabet d'entrée de la structure séquentielle est égal à : $\{uv, u\bar{v}, \bar{u}v, \bar{u}\bar{v}\}$. L'alphabet de sortie est égal à : $\{s_1, s_2, s_1\bar{s}_2, \bar{s}_1s_2, \bar{s}_1\bar{s}_2\}$.

Etudions l'évolution du marquage du réseau de PETRI. A partir du marquage initial M_0 où seule la place P_1 est marquée. L'on obtient soit un seul marqueur dans la place P_3 après le franchissement de la transition t_1 , soit un seul marqueur dans la place P_2 après le franchissement de la transition t_2 . Les deux transitions t_1 et t_2 ne sont jamais en conflit les événements associés uv et $u\bar{v}$ étant exclusifs. Dénotons M_1, M_2 les marquages correspondant aux places P_2 occupée, P_3 occupée. L'ensemble $[M]$ des marquages du réseau est égal à $[M] = \{M_0, M_1, M_2\}$. La machine séquentielle matérialisant la structure de commande possède trois états q_0, q_1, q_2 associés aux marquages M_0, M_1, M_2 . Son fonctionnement est résumé par les matrices de sortie données par les figures V.8 et V.9. Ces matrices sont obtenues par analyse du réseau.

	$\bar{u}\bar{v}$	$\bar{u}v$	uv	$u\bar{v}$
q_0	q_0	q_0	q_2	q_1
q_1	q_0	q_0	q_1	q_1
q_2	q_2	q_0	q_2	q_2

Figure V.8

q_0	s_1, s_2
q_1	s_1, \bar{s}_2
q_2	\bar{s}_1, \bar{s}_2

Figure V.9

V.1.2.e Les définitions d'une machine séquentielle

La représentation tabulaire classique d'une machine séquentielle revient à définir celle-ci par un quintuplet $\mathcal{M} = (X, Z, Q, f, g)$:

- X est l'ensemble fini, non vide, des entrées (alphabet d'entrée).
- Z est l'ensemble fini non vide, des sorties (alphabet de sortie).
- Q est l'ensemble fini, non vide, des états.
- f est la fonction "état suivant" (elle établit l'application $Q \times X \rightarrow Q$).
- g est la fonction "sortie" (elle établit, dans le cas général, l'application $Q \times X \rightarrow Z$).

f et g sont représentées grâce aux tables.

Par analogie, la représentation réticulée d'une machine séquentielle sera associée à la définition d'un sextuplet $(E, S, M, \rho, \mu, \nu)$:

- $E = \{e_1, \dots, e_n\}$ est un ensemble fini, non vide, d'événements.
 - $S = \{s_1, \dots, s_p\}$ est un ensemble fini, non vide, de sorties.
 - $M = \{m_1, \dots, m_q\}$ est un ensemble fini, non vide, de marquages.
- ρ est une application de M dans E . On l'appelle fonction de réceptivité. À chaque marquage, elle fait correspondre un sous-ensemble de E auquel la machine est réceptive.

μ est la fonction "marquage suivant" (application de ρ dans M).

ν est la fonction de sortie (application de M dans S).

De même que f et g étaient tabulées dans la représentation classique, ρ , μ et ν seront représentées sur le réseau de PETRI.

Remarque 1:

La définition de ν correspond à la définition d'une machine de MOORE, puisqu'un "état" au sens classique du terme est associé à chaque marquage. Nous verrons, dans le chapitre relatif à la

réduction d'un réseau de PETRI, l'extension de cette définition au cas de machines de MEALY.

Une importante notion a été introduite : Celle de réceptivité.

Une machine séquentielle représentée par un réseau de PETRI est : réceptive, à un instant donné, à un événement e , si cet événement est capable de faire évoluer le marquage du réseau.

Exemple : Lampe commandée par un bouton poussoir.

Soit le problème suivant : Une lampe doit alternativement s'éclairer et s'éteindre chaque fois qu'on presse un unique bouton (figure V-10).

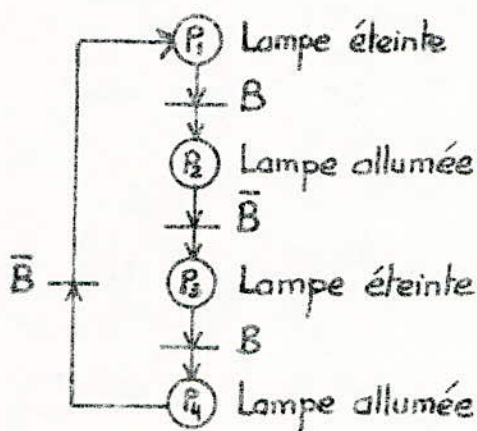


Figure V-10

La lampe est actuellement éteinte ; pressons le bouton, elle s'éclaire. Son système de commande avait donc une réceptivité pour l'information « bouton pressé ». La satisfaction de cette réceptivité a fait modifier l'événement engendré par le système, qui « lampe éteinte » est devenu « lampe éclairée », comme le montre le réseau de la figure V-10.

Si nous maintenons le bouton pressé la lampe ne doit pas s'éteindre ; par contre, si nous lâchons le bouton, puis le pressons à nouveau, la lampe doit s'éteindre ; ce qui suppose qu'à cet instant le système a acquis une nouvelle réceptivité pour « bouton pressé ». L'acquisition de cette nouvel.

le réceptivité n'a pas pu se faire aléatoirement, sinon, elle aurait pu se faire avant qu'on ait relâché le bouton. Elle a donc été la conséquence de l'information « bouton lâché »; il a bien fallu que le système possède une réceptivité pour cette information, réceptivité dont la satisfaction n'a pas provoqué de modification de l'événement engendré par le système, mais a provoqué une modification de sa réceptivité.

Remarque 2:

Le passage du réseau de PETRI à la table de fluence se fait simplement en effectuant la correspondance marquage \rightarrow état et en décomposant tous les événements liés aux transitions en événements élémentaires (colonnes de la table).

Exemple:

On désire commander l'allumage ou l'extinction d'une lampe L à l'aide de deux boutons poussoirs m et a . Initialement la lampe L est éteinte et les deux boutons poussoirs relâchés. Son allumage est obtenu par action fugitive sur le bouton poussoir m , le bouton poussoir a étant relâché. Son extinction est obtenue par action sur le bouton poussoir a . Le réseau de PETRI obtenu est représenté par la figure V-11. L'ensemble M des marquages $[M] = \{M_0, M_1\}$ est caractérisé par le tableau matriciel de la figure V-12.

Remarque 2:

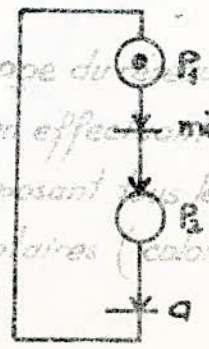


Figure V-11
Exemple

	M_0	M_1
P_1	1	0
P_2	0	1

Figure V-12

Le passage du réseau de PETRI à la table de fluence se fait simplement en effectuant la correspondance marquage \rightarrow état et en décomposant tous les événements liés aux transitions en événements élémentaires (colonnes de la table)

On désire commander l'allumage ou l'extinction d'une lampe L à l'aide de deux boutons poussoirs m et a .

Alphabet d'entrée $\{\bar{m}\bar{a}, \bar{m}a, m\bar{a}, ma\}$; alphabet de sortie $\{l, \bar{l}\}$.
 L'association de deux états q_0, q_1 aux marquages M_0, M_1 permet la représentation de la matrice des états (figure V-13).

	$\bar{m}\bar{a}$	$\bar{m}a$	$m\bar{a}$	ma
q_0	q_0	q_0	q_0	q_1
q_1	q_1	q_0	q_0	q_1

Figure V-13

V.2 Exemples d'utilisation

V.2.1 Exemple 1

Proposons-nous de décrire le fonctionnement d'un automate chargé de mettre en marche et d'arrêter un moteur thermique afin de le maintenir en température.

L'état initial M correspond à l'arrêt du moteur. Un événement θ_1 , fourni par un détecteur de température commande le lancement du démarreur (état D). Lorsque le moteur tourne de lui-même, un contacteur centrifuge coupe l'alimentation du démarreur (événement C); le nouvel état M est la marche du moteur. Le moteur s'arrêtera (retour à l'état \bar{M}) lorsque sa température sera suffisante (événement θ_2). Perfectionnons le système. θ_1 actionnera un temporisateur T_1 (On ajoute T_1 à la place D) lequel coupera l'alimentation du démarreur (événement FT_1) si au bout du temps t_1 C ne l'a pas déjà fait. Un second temporisateur T_2 permet une relaxation de la batterie de démarrage durant un temps t_2 , au bout duquel (événement FT_2) un nouvel essai de mise en marche s'effectue. Nous parlerons plus tard de la place X .

Ceci nous conduit à la figure V-14, qui représente un réseau de PETRI relatif à une machine séquentielle:

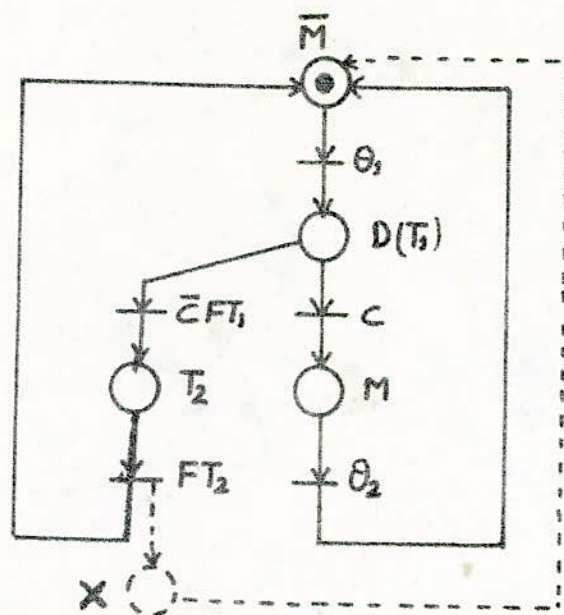


Figure V-14

- places et transitions se succèdent alternativement : il est en effet évident que des « générations spontanées » d'états ne sauraient se produire et que tout événement doit avoir une conséquence ;

- Un événement n'a d'effet que lorsque tous les états directement antécédant (origine des flèches incidentes) existent : On dit que la transition est validée ou franchissable par le marquage des places d'entrée ;

- la conséquence d'un événement est le marquage de toutes les places de sortie de la transition concernée et le démarquage des places d'entrée ;

- le réseau doit être vivant : à partir du marquage initial (mettre un marqueur dans la place \bar{M}) on doit pouvoir marquer toutes les places du réseau en produisant successivement les événements ;

- le réseau doit être sauf : aucune place ne doit posséder plus

d'un marqueur ; cette éventualité conduirait à un blocage. Ainsi, la place X est marquée après le premier démarrage manqué, elle est démarquée par l'événement θ_2 (si le deuxième essai de mise en route a été concluant); par contre si le deuxième essai est sans résultat, il y a blocage (volontaire) car FT_2 n'est pas franchissable.

N.B :

Si, par hasard, les événements FT_1 et C se présentent simultanément, les transitions FT_1 et C seront simultanément franchies en générant des marquages contradictoires. Ce conflit est évité en remplaçant FT_1 par $\bar{C}.FT_1$; cela empêche également le marquage de T_1 si le démarrage s'effectue correctement.

V.2.2 Exemple 2 :

Cherchons à représenter le système séquentiel associé au dispositif de la figure V.15 et dont le fonctionnement doit être le suivant :

- au départ, le plateau est en A ($H_1 = V_1 = 1$), tous les moteurs arrêtés).
- on appuie sur D. Ceci provoque le mouvement du plateau suivant le cycle ACDA. De retour en A, le plateau doit s'arrêter. En outre, durant la marche CDA, une lampe L, portée par P (symbolisant par exemple le travail d'un outil) doit s'allumer.

V.2.2.a Construction du réseau de PETRI

Il va se déduire de l'énoncé très simplement. Les événements sont les occurrences des entrées D, H_1 , V_1 , H_2 , V_2 . Les conditions sont les sorties (MD, MH, MG, MB, L).

A partir de la place initiale (tous moteurs à l'arrêt), l'occurrence de D va provoquer l'apparition de deux sorties (MD et MH) pour amener le plateau de A vers C. Ces deux mouvements a priori indépendants. MD sera

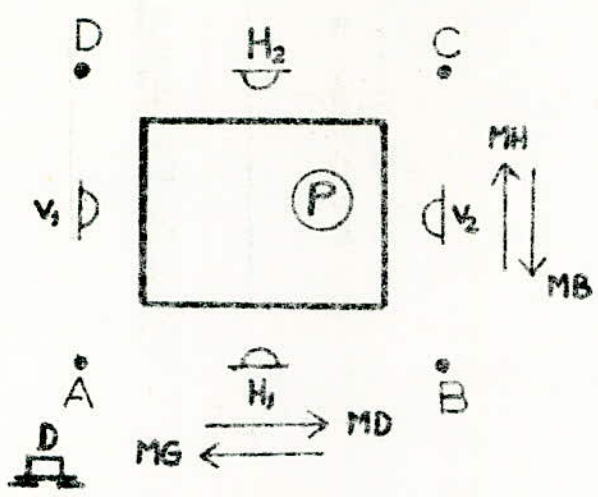


Figure V-15

Transition	Places(s) d'entrées / P. de sortie
t_1	$P_1 / P_2, P_6$
t_2	P_2 / P_3
t_3	$P_3, P_7 / P_4, P_8$
t_4	P_4 / P_5
t_5	$P_5, P_8 / P_1$
t_6	P_6 / P_7

Figure V-17

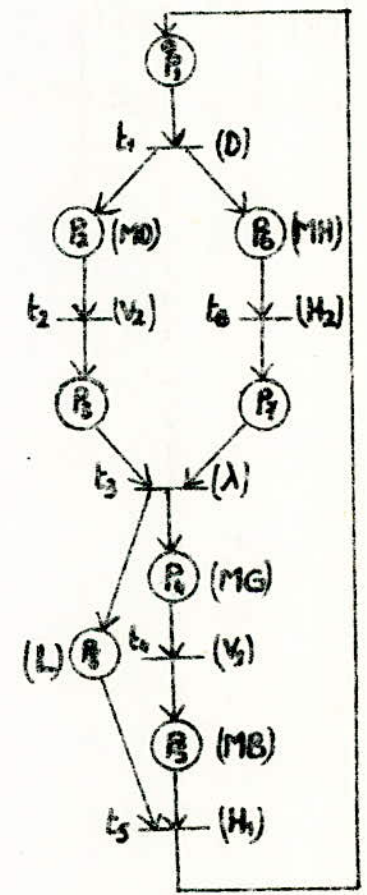


Figure V-16

arrêté par l'occurrence de V_2 et MH par celle de H_2 . Lorsque ces deux événements se seront produits, cela correspondra à la présence du plateau en C et donc on pourra réaliser les conditions allumage de L et marche à gauche (MG). L restera allumée jusqu'à l'arrivée en A. Par écarte, MG sera supprimé par l'occurrence de V_3 (qui traduit l'arrivée en D) qui provoquera simultanément le mouvement de descente (MB) arrêté par l'occurrence de H_3 .

A droite de chaque place ou transition, on a indiqué entre parenthèses la sortie ou l'entrée associée (figure V.16).

V.22.b - Remarques sur le réseau de PETRI

1. En général, un système logique est muni d'une procédure d'initialisation qui permet de le placer dans une configuration particulière avant utilisation.

Dans le cas des machines séquentielles, on emploie le terme état initial, et pour les réseaux de PETRI, le terme marquage initial.

Ici, d'après la description précédente, c'est p_3 qui sera marquée initialement et elle seule.

2. Ce réseau peut être représenté structurellement par l'ensemble des couples place(s) d'entrée / place(s) de sortie de chaque transition (fig. V.17).

3. A partir du tableau de la figure V.17 et du marquage initial, il est simple d'obtenir tous les marquages du réseau et toutes les transitions validées : pour un marquage examiné m , une transition t est validée si toutes ses places d'entrée sont incluses dans le marquage. Le nouveau marquage après franchissement de t , est obtenu en substituant dans m les places de sortie aux places d'entrée de t .

D'où le tableau de la figure V.18 (complet lorsque tous les marquages ont été examinés).

On obtient ainsi l'ensemble des marquages accessibles à

partir de m_0 . Le graphe des marquages (graphe d'état) est donné (Figure V.19) sous forme de réseau de PETRI.

L'examen de la colonne des marquages du tableau V.18. permet de vérifier que le réseau est *sauv* (aucun marquage ne contient plusieurs fois la même place).

Si le graphe des marquages est fortement connexe c'est-à-dire si de tout marquage m_i on peut atteindre un marquage m_j par une séquence de franchissements de transitions. Ce qui est le cas dans notre exemple.

L'examen de la colonne "transition validée" permet de vérifier que le réseau est *vivant* (chaque transition y apparaît au moins une fois).

Enfin les conflits éventuels peuvent être détectés au niveau de chaque marquage examiné en considérant, s'il y a partage de places, la fonction de réceptivité.

Ici, le réseau est conforme et sans conflit et comporte 7 marquages

-4. Le point le plus important à remarquer est la structure du réseau. L'occurrence de D , p_1 étant marquée, provoque le marquage en parallèle de deux places, p_2 et p_6 . Ceci permet de représenter avec une structure en vérité très simple l'asynchronisme de MH et MD. A priori, le concepteur sait qu'il s'agit de deux conditions initialisées par le même événement. Par contre, on ne sait rien sur l'occurrence des événements V_2 et H_2 qui doivent les supprimer. Les propriétés du réseau de PETRI font qu'on n'a pas à envisager les trois possibilités (contact V_2 avant contact H_2 , contact H_2 avant contact V_2 , contacts H_2 et V_2 simultanés) : Il suffit de les écrire sur deux branches en parallèle. Cet outil est donc particulièrement agréable pour traiter des événements a priori indépendants, c'est-à-dire asynchrones.

Nous verrons par la suite que ce réseau peut être réalisé directement à partir de modules séquentiels. Mais nous pouvons nous pa

Marquage examiné (m)	Marquage suivant	Transition validée	Fonction de réceptivité $\rho(m)$
$m_0 : P_1$	P_2, P_6	t_1	D
$m_1 : P_2, P_6$	P_2, P_7	t_6	H_2
	P_2, P_6	t_2	V_2
$m_2 : P_2, P_7$	P_2, P_7	t_2	V_2
$m_3 : P_2, P_6$	P_2, P_7	t_6	H_2
$m_4 : P_2, P_7$	P_2, P_7	t_3	λ
$m_5 : P_6, P_6$	P_6, P_7	t_4	V_1
$m_6 : P_2, P_7$	P_1	t_5	H_1

Figure V. 18

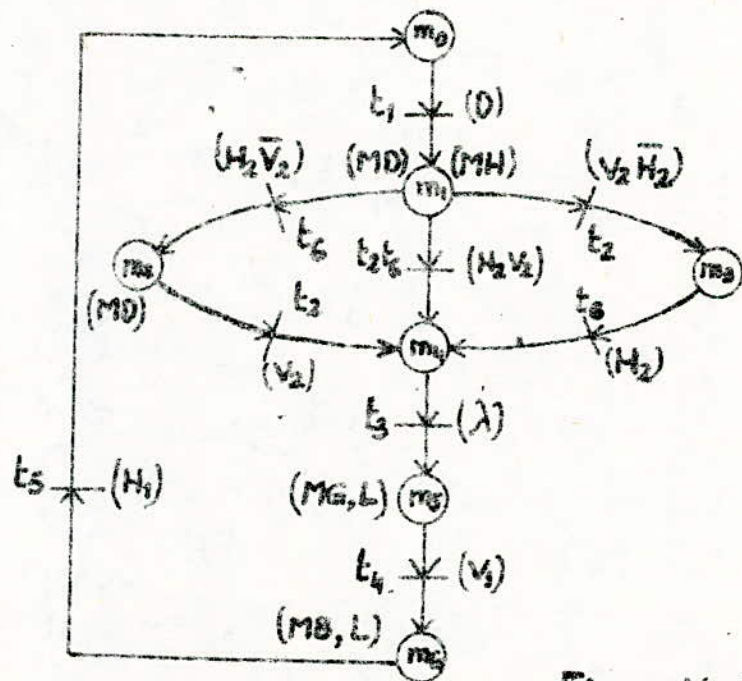


Figure V. 19

d'ores et déjà, la question de savoir s'il est le plus simple possible, structurellement parlant. Par exemple, la place p_8 ne sert qu'à définir la sortie L. En fait, L doit être à 1 tant que MG ou Mb (places p_4 ou p_5) est excité. On pourrait donc songer à supprimer la branche qui contient p_8 , en reportant comme sorties associées respectivement à p_4 et p_5 , MG et L ainsi que MB et L.

V.2.2.C Notion de sous-programme

Lorsque le cahier des charges établi ci-dessus sera présenté à l'utilisateur, il y a de fortes chances qu'il ne lui convienne pas. Et cela d'autant plus qu'il n'avait pas, au départ, une idée précise de ses besoins.

Aussi, nous allons être amenés à introduire - pour utilisation future de la machine - la possibilité d'atteindre B à partir de A ou/et de C (translation du plateau). On doit pouvoir, suivant le cas - et sans grandes modifications - choisir l'un des cycles ACDA, ABACDA, ACBCDA ou ABACBCDA.

Comme il s'agit, à partir de A ou C, d'effectuer ou non la même séquence (aller-retour), on pense immédiatement à l'introduction d'un sous-programme (c'est-à-dire un sous-réseau). Il suffira alors d'en modifier les arguments pour passer d'un cycle à l'autre.

Comment allons-nous le représenter sur le réseau de PETRI ? A partir du réseau de la figure V.16 nous interviendrons aux passages en A (départ) et C (places p_1 et p_6 du réseau de la figure V.20.a) les places d'appel d'un sous-programme (représenté par le réseau banalisé de la figure V.20.b). C'est au niveau de ces places que se feront le passage d'arguments (e_1, e_2, S_1, S_2) entre le réseau principal, et le sous-programme, et la mémorisation de l'adresse de retour. (Ces places peuvent être représentées symboliquement par un double cercle).

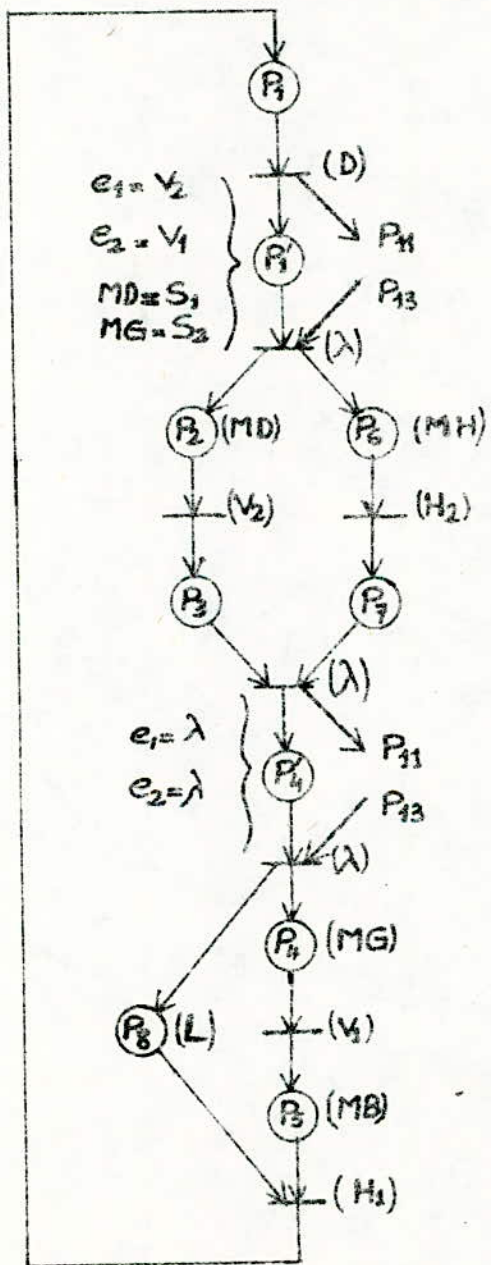
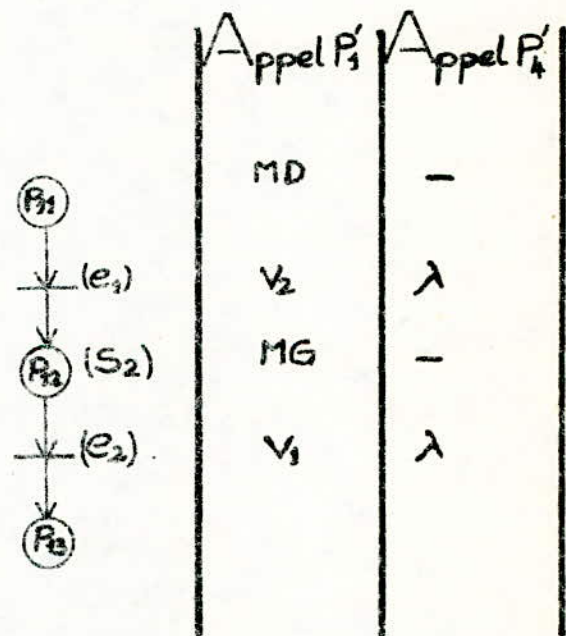


Figure 20.a : Réseau principal



Affectation des arguments

(cycle ABACDA)

Figure 20.b

Les arguments sont faciles à affecter, ceux de la figure V.20 correspondent au cycle programmé ABACDA.

Remarque

La non utilisation du sous-programme en C pour le cycle ABACDA se traduit par un passage transitoire dans le sous-programme ($e_1 = e_2 = \lambda$). Ainsi, la structure de l'ensemble est figée, seules les affectations d'arguments varient suivant le cycle désiré. D'autres solutions sont évidemment possibles (débranchements en place 11 par exemple).

V.3 Conclusion

Il est important de noter :

- La clarté de la représentation : Seules les transitions succédant aux places marquées sont réceptives aux événements qui les définissent; il n'y a donc pas lieu, comme dans la méthode d'Huffman, de se soucier de conséquences que pourraient engendrer les autres événements. Par ailleurs, l'ordre d'arrivée des événements concernant des « sous-machines » indépendantes (branches parallèles simultanément réceptives des réseaux de PETRI) n'a pas à être pris en considération alors qu'avec Huffman l'asynchronisme recherché n'est garanti qu'en envisageant toutes les successions possible de combinaisons. Il s'ensuit qu'un réseau de PETRI est moins encombré donc plus clair qu'un diagramme de fluence;

- la facilité de faire évoluer le réseau pour tenir compte de nouvelles exigences;

- que le manque d'information relatif à la durée des événements (impulsions ou niveaux) n'empêche pas de construire le graphe.

Réduction du nombre d'états

VI.1 Introduction

Ce chapitre est consacré à une approche de réduction - ou simplification - des réseaux de PETRI qui peut être, dans certains cas, intéressante et qu'il est, en tout cas, utile de connaître.

V.2 - Buts et principes de la simplification

Partant du cahier des charges de l'automatisme étudié, on obtient un réseau de PETRI. Plusieurs questions se posent alors : le réseau est-il correctement établi ? Peut-on réduire sa taille ? Comment ? Selon quels critères ?

Répondre à la première question revient à examiner si :

a - le réseau interprété est correct, c'est-à-dire s'il est conforme sans conflit et ne contient ni blocage, ni cycle. La notion de cycle est la même que pour les représentations tabulaires : l'existence d'une configuration des entrées telle que le système évolue, sans se stabiliser sur un ensemble de places (ceci correspond à un cycle de franchissements de transitions).

b - Il représente bien le cahier des charges, c'est-à-dire le fonctionnement désiré de l'automatisme.

VI.2.1 Buts de la simplification

Venons-en, à présent, à la réduction de la taille du réseau, c'est-à-dire du nombre de places ou de marquages ou encore d'états de l'automatisme.

Pourquoi réduire ? Un cahier des charges - réseau de PETRI - se présente sous forme d'une coordination de séquences. Cette représentation ne préjuge en rien de la nature séquentielle du problème qui peut fort bien être combinatoire. Bien que nous n'ayons pas encore parlé de réalisation technologique, on conçoit que la réduction du réseau simplifiera la

réalisation de la partie séquentielle de la machine (nombre de mémoires ou variables internes de codage).

Ceci est, évidemment, encore plus vrai si on envisage de matérialiser chaque place du réseau par une mémoire.

Comment réduire ? Nous avons jusqu'ici affecté, à chaque place une (des) sortie(s), si bien qu'à un marquage donné correspond une configuration des sorties indépendante des variables d'entrée (représentation d'une machine de MOORE).

Tout comme dans les méthodes classiques de réduction, simplifier le réseau nécessite de passer d'une machine de MOORE à une machine MEALY, c'est-à-dire de rendre les sorties associées aux places fonction des variables d'entrée.

Si on compare le processus de synthèse classique avec la synthèse basée sur les réseaux de PETRI, on s'aperçoit que la démarche est différente (figure VI.1).

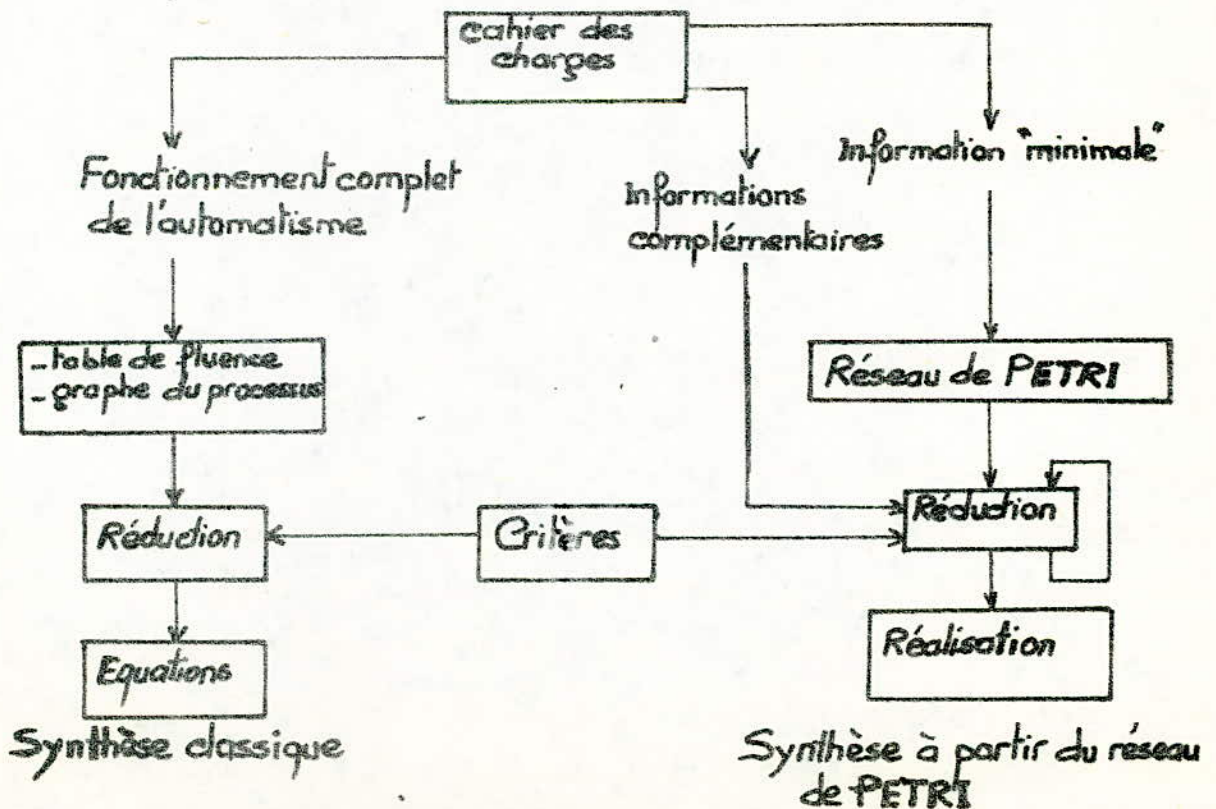


Figure VI.1

- Pour une synthèse classique, le document élaboré à partir du cas des charges (graphe de processus, tables, ...) est difficile à obtenir parce qu'il décrit le fonctionnement spécifique de la machine étudiée et contient toute l'information nécessaire à sa synthèse. Par contre, on peut lui appliquer des méthodes de réduction qui, tout en respectant ce fonctionnement, conduisent à une réalisation dite minimale.

- au contraire, l'établissement du réseau de PETRI ne nécessite cette analyse fine du fonctionnement car il contient l'information minimale pour représenter les spécifications. Mais, dérivant ainsi toute une classe de machines, sa réduction n'est possible que par apport d'informations complémentaires sur le fonctionnement. Plus on veut le réduire, plus il est spécifié jusqu'à aboutir à la transcription du graphe de processus (ou du graphe d'état).

VI.2.2 Principe de la simplification

Les réseaux de PETRI que nous étudierons sont des réseaux concrets et sans conflit. Au départ, la machine considérée est représentée par un réseau de PETRI où une sortie est affectée à une place. La simplification du réseau nous conduira à associer une sortie à un couple événement/marquage.

On peut considérer deux types de simplifications :

- Première simplification

Elle est liée à la structure du réseau de PETRI, quelle que soit son interprétation. Elle a pour but de supprimer les redondances introduites lors de son établissement. On obtient alors un réseau "minimal" contenant la même information que le réseau initial. Cette étape est particulièrement utile dans le cas de grands systèmes où l'on introduit involontairement des redondances d'informations.

Deuxième simplification

Elle est liée à une connaissance plus approfondie de la machine. Des informations complémentaires, n'apparaissant pas sur le réseau, permettent de réduire sa taille en diminuant le nombre de places et transitions (fusion de places).

Remarque

Une machine séquentielle peut être représentée par un réseau de PETRI en affectant les sorties à un état du réseau et à un événement. Dans ce cas, on définit $\mathcal{M}: \langle E, S, M, \rho, \sigma, \mu, \nu \rangle$.

- $E = \{e_1, \dots, e_n\}$ est un ensemble fini, non vide, d'événements (construits sur l'alphabet d'entrée de la machine).

- $S = \{s_1, \dots, s_k\}$ est un ensemble fini, non vide, de sorties.

- $M = \{m_1, \dots, m_l\}$ est un ensemble fini, non vide, de marquages du réseau.

- ρ est une application multivoque de M dans E . À chaque élément de M , elle fait correspondre un sous-ensemble $\rho(m)$ de E auquel la machine est réceptive. ρ est appelée fonction de réceptivité. L'ensemble $\rho(m)$ est composé d'événements qui font évoluer le marquage.

σ est une application multivoque de M dans E . À chaque élément m de M , elle fait correspondre un sous-ensemble $\sigma(m)$ de E auquel la machine est sensible. σ est appelée fonction de sensibilité. L'ensemble $\sigma(m)$ est composé d'événements modifiant la configuration des sorties pour le marquage m .

- μ est la fonction de transition (ou marquage suivant) $\forall m \in M$

$\rho(m) \xrightarrow{\mu} M$: pour chaque élément m de M la fonction μ n'est définie que sur l'ensemble de réceptivité $\rho(m) \subseteq E$. On écrira, par extension: $\mu(m, e) = m'$ ($e \in \rho(m)$)

- ν est une fonction multivoque (fonction de sortie): $\forall m \in M$

$\sigma(m) \rightarrow S$. Pour chaque élément m de M , ν est définie sur l'ensemble de sensibilité $\sigma(m) \subseteq E$. On écrira, par extension :
 $\nu(m, e) = S(e \in \sigma(m))$. Les fonctions ρ, σ, μ et ν sont représentées sur le réseau de PETRI lui-même.

VI.3 Première simplification : Recherche des places implicites

VI.3.1 Définitions

- Soit :
- $\Sigma = \{\sigma_1, \dots, \sigma_j\}$ un ensemble de places du réseau.
 - $M = \{m_0, m_1, \dots, m_k\}$ l'ensemble des marquages du réseau.

La place p partitionne M en deux sous-ensembles :

- M_p tel que $\forall m_i \in M_p \quad m_i(p) = 1$
- M'_p tel que $\forall m_i \in M'_p \quad m_i(p) = 0$

En fait, M_p représente l'ensemble des marquages où apparaît la place p et M'_p l'ensemble de ceux où elle n'apparaît pas.

Définition

- Une place p est « implicite » d'un ensemble Σ de places si :
- a- $\forall m_i \in M_p \exists \sigma_k \in \Sigma$ t.q. $m_i(\sigma_k) = 1$
 - b- $\bigcup_{\sigma_i \in \Sigma} M_{\sigma_i} = M_p$

Ce qui, en clair, veut dire :

- a- On retrouve dans chaque marquage où apparaît p (ensemble M_p) une place σ_i de Σ .
- b- L'ensemble de tous les marquages où apparaît σ_i n'est autre que M_p .

Le marquage de la place p s'exprime alors comme une fonction logique du marquage des places impliquantes Σ . La place p peut être supprimée du réseau.

VI.3.2 Méthode de recherche

Pour rechercher si une place p est implicite, on va d'abord construire deux ensembles de places appelés Compatibles C_p et incompatibles I_p , associés à la place p .

$$- C_p = \{ p_i / \exists m_j \in M_p \text{ t.q. } m_j(p_i) = 1 \}$$

$$- I_p = \{ p_k / \exists m_j \in M'_p \text{ t.q. } m_j(p_k) = 1 \}$$

C_p est l'ensemble des places qui apparaissent dans au moins un marquage de M_p et I_p l'ensemble de celles qui apparaissent au moins une fois dans un marquage de M'_p .

En général, $C_p \cap I_p \neq \emptyset$.

On appelle compatible absolu C'_p l'ensemble de places obtenues en éliminant de C_p les places communes ($C_p \cap I_p$) et la place p .

$$C'_p = C_p - (C_p \cap I_p) - p.$$

Exemple 1

Soit le réseau de la figure V.2.a. Le marquage de p_1 se fera simultanément avec celui de p_2 . L'enlèvement du marqueur de p_1 se fera en même temps que celui de p_3 . D'où on obtient le réseau équivalent (fig. V.2.t).

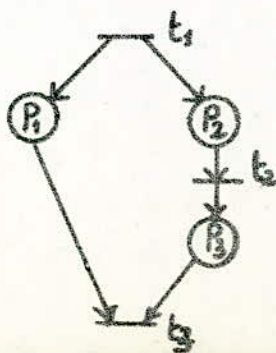


Figure VI.2.a



Figure VI.2.t

Remarque 1

De façon sommaire, l'existence d'une place implicite sera détectée par l'existence, en parallèle sur une branche comportant n places d'une branche ayant une seule place et mêmes transition - origine et transition de fin.

Remarque 2

Dans la mesure où les places sont associées aux sorties du système, simplification par suppression des places implicites revient à "regrouper des sorties".

VI.3.3 Simplification

Une place implicite n'apporte aucune information supplémentaire quant au marquage du réseau, c'est-à-dire la réalisation des conditions. En effet il suffit de savoir qu'une des places implicantes est marquée pour que la place implicite le soit aussi.

On va donc simplifier le réseau en supprimant la place implicite. Elle disparaît donc comme une place d'entrée ou de sortie des transitions q lui étaient associées.

Exemple 2

En suivant ce raisonnement, on peut ramener le réseau de la figure VI.3.a à celui de la figure VI.3.b

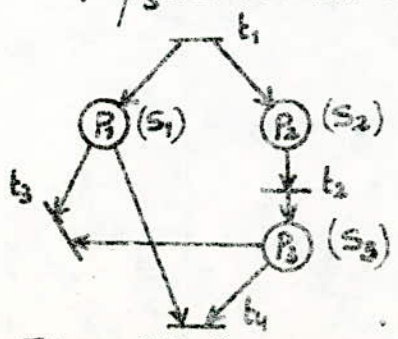


Figure VI.3.a

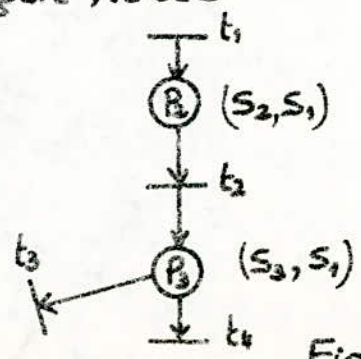


Figure VI.3.b

On a : $C_{P_1} = \{P_1, P_2, P_3\}$ car les marquages possibles sont :
 $\{P_1, P_2\}$ et $\{P_2, P_3\}$.

$$I_{P_1} = \emptyset$$

$$C'_{P_1} = C_{P_1} - (C_{P_1} \cap I_{P_1}) - P_1 = \{P_1, P_2, P_3\} - \{\emptyset\} - P_1 = \{P_2, P_3\}$$

C'_{P_1} contient l'ensemble des places implicantes possibles de la place;

$$M_{P_1} = \{P_1 \cdot P_2, P_1 \cdot P_3\} \text{ et } P_2 \in \{P_1 \cdot P_2\} ; P_3 \in \{P_1 \cdot P_3\}.$$

Les sorties affectées à la place implicite sont reportées sur les places implicantes. Les sorties qui étaient réalisées lorsque la place implicite était marquée sont maintenant réalisées par le marquage d'au moins une place implicante.

Remarque

Il faut noter que cette première simplification ne s'effectue pas en une seule fois, mais d'une façon itérative. Après suppression de la place implicite, le nouveau réseau est examiné et de nouvelles places implicites recherchées.

Exemple 3

Soit le réseau de la figure VI-4-a

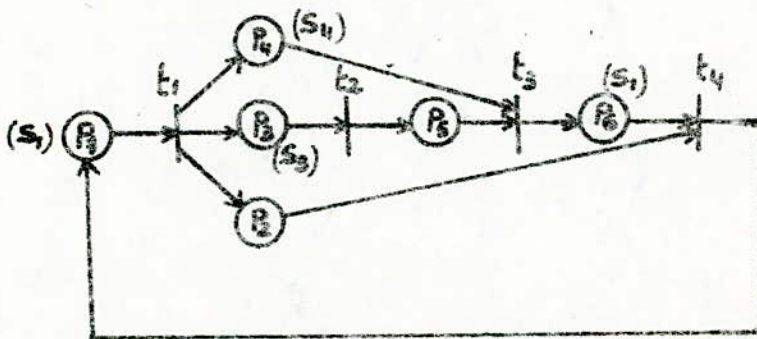


Figure VI-4-a

Soit, $\mathcal{P} = \{P_1, P_2, P_3, P_4, P_5, P_6\}$

• M est l'ensemble des marquages du réseau (cf. Exemple du paragraphe I.6).

La place P_4 partitionne M en deux sous-ensembles :

- $M_{P_4} = \{m_1, m_2\}$

- $M'_{P_4} = \{m_0, m_3\}$

• Construction de deux ensembles : compatibles C_{P_4} et incompatible I_{P_4} , associés à la place P_4 .

- $C_{P_4} = \{P_2, P_3, P_4, P_5\}$

- $I_{P_4} = \{P_1, P_2, P_6\}$

• Compatible absolu C'_{P_4} :

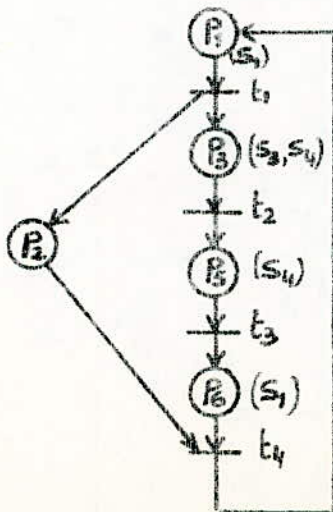
- $C'_{P_4} = C_{P_4} - (C_{P_4} \cap I_{P_4}) - P_4$

or, $C_{P_4} \cap I_{P_4} = \{P_2\}$

Donc, $C'_{P_4} = \{P_2, P_3, P_4, P_5\} - \{P_2\} - P_4 = \{P_3, P_5\}$.

C'_{P_4} contient l'ensemble des places implicites possibles (P_3 et P_5) de la place P_4 .

Le réseau de la fig. VI.4-a se transforme en celui de la fig. VI.4-b.



$m(P_4) = m(P_3) + m(P_5)$
Suppression de la place P_4

Figure VI.4-b

Recherche d'une nouvelle place implicite après l'étude du réseau de la figure VI-4-b

Soit: $\mathcal{P} = \{P_1, P_2, P_3, P_5, P_6\}$

- M l'ensemble des marquages du nouveau réseau.

m_0		m_1		m_2		m_3
1		0		0		0
0		1		1		1
0	,	1	,	0	,	0
0		0		1		0
0		0		0		1

La place P_2 partitionne M en deux sous ensembles :

- $M_{P_2} = \{m_1, m_2, m_3\}$

- $M'_{P_2} = \{m_0\}$

Les ensembles de places : Compatible C_{P_2} et incompatible I_{P_2} , associés à la place P_2 .

- $C_{P_2} = \{P_2, P_3, P_5, P_6\}$

- $I_{P_2} = \{P_1\}$

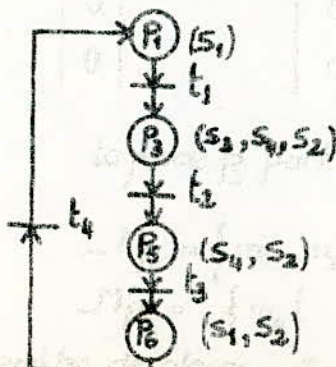
Détermination du compatible absolu C'_{P_2}

- $C'_{P_2} = C_{P_2} - (C_{P_2} \cap I_{P_2}) - P_2$, avec $(C_{P_2} \cap I_{P_2}) = \emptyset$

D'où: $C'_{P_2} = \{P_2, P_3, P_5, P_6\} - \emptyset - P_2 = \{P_3, P_5, P_6\}$

$\{P_3, P_5, P_6\}$ sont les places implicantes possibles de la place P_2 .

Le réseau de la figure VI-4-b se transforme finalement en celui de la figure VI-4-c.



$m(P_2) = m(P_3) + m(P_5) + m(P_6)$
Suppression de la place P_2 .

Figure VI-4-C

VI-4 Deuxième simplification: Fusion des places

VI-4-1 Principe

Le réseau de PETRI utilisé pour représenter le fonctionnement d'une machine particulière décrit, en fait, toute une classe de machines.

Pour pouvoir simplifier le réseau (diminuer le nombre de places et transitions) il faut le spécifier, c'est-à-dire réduire la classe de machines qu'il représente. On utilisera pour cela des informations sur le fonctionnement de la machine (nature et évolution des signaux d'entrée).

La structure de la machine séquentielle -réalisée par un circuit séquentiel- est représentée par la figure VI-5-a. Le circuit de sortie est relativement simple. Chaque sortie s'écrit :

$$S_i = \sum_j m(p_j)$$

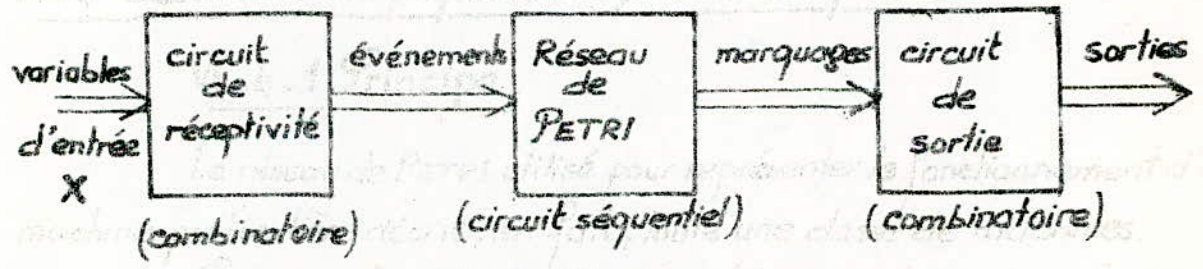


Figure VI-5-a

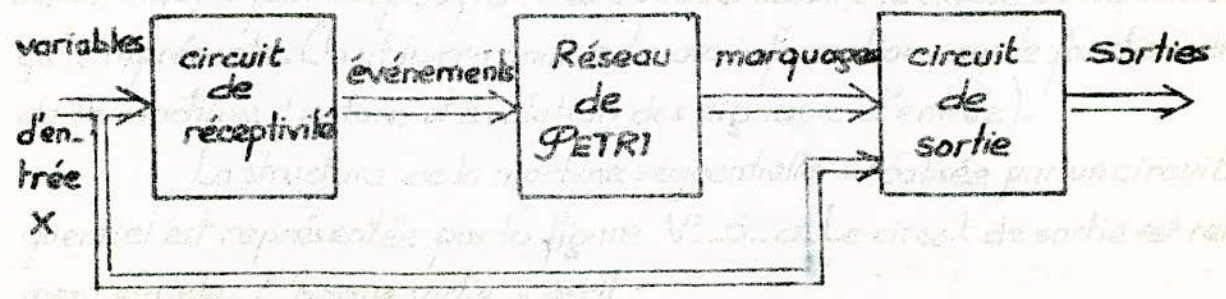


Figure VI-5-b

Σ : Somme booléenne ou somme binaire

$\{p_j\}$ représente l'ensemble des places où est affectée ponctuellement la sortie S_i .

Le schéma obtenu après simplification (figure VI-5-b) est modifié dans le sens suivant :

- le réseau de PETRI est plus simple (la partie séquentielle de la machine est réduite).
- la complexité du circuit de sortie augmente : les sorties sont des fonctions combinatoires de l'état des places et aussi des variables d'entrée.

Chaque sortie s'écrit alors :

$$S_i = \sum_j f(m(p_j), x)$$

Soient p_1 et p_2 deux places reliées par une transition t à laquelle est associé un événement E (figure VI-6-a)

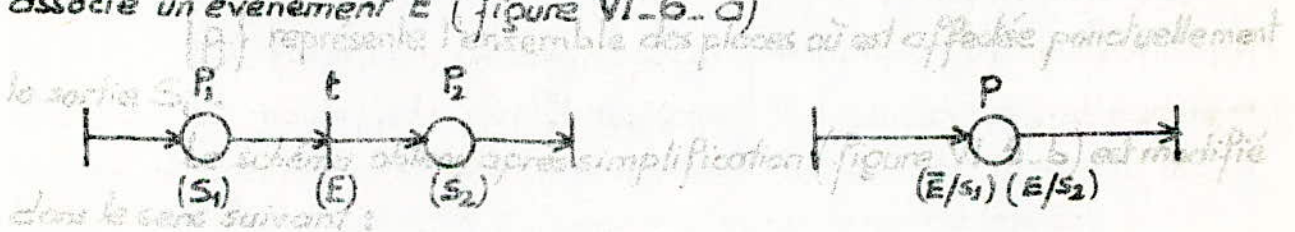


Figure VI-6-a

Figure VI-6-b

Si l'étude approfondie du fonctionnement de l'automatisme nous apprend que l'occurrence de l'événement E n'est pas fugitive, mais se maintient à la place p_2 , il est alors possible de supprimer la transition t . L'évolution séquentielle de la machine qui était assurée par la structure du réseau (Figure VI-6.a) est maintenant réalisée par l'évolution des événements (\bar{E}, E) (Figure VI-6-b).

Les places p_1 et p_2 sont fusionnées en une seule place p . L'état de la sortie à cette place est lié à la présence de l'événement E (c'est-à-dire à la configuration des entrées) qui assure une partition de la place p :

Soient p_1 et p_2 deux places reliées par une transition t à laquelle est associé un événement E (figure VI-6-a)

$$S_1 = m(p). \bar{E}$$

$$S_2 = m(p). E$$

On a donc, pour le réseau simplifié, une représentation de la machine séquentielle telle qu'elle est définie dans la remarque du paragraphe VI.2.2.

Pour le réseau initial, la machine était réceptive à l'événement E uniquement pour passer du marquage p_1 au marquage p_2 . Cet événement était sans influence pour la place p_2 .

Au contraire, dans le réseau simplifié, l'événement E ne fait plus évoluer le marquage du réseau, mais la machine reste sensible à E tant que la place p est marquée.

La simplification consiste donc à déterminer les ensembles de places fusionnables, le nouveau réseau et les nouvelles fonctions de sortie.

Il est évident que, pour un réseau donné, un certain nombre de fusions de places sont possibles. Le choix de la fusion à retenir dépend alors de critères qui restent à définir : Sens physique du réseau simplifié, sécurité de fonctionnement, facilité de réalisation...

VI.4.2 Exemple

Considérons la figure VI.7. Deux cahiers des charges différents sont définis :

- I. Aller-retour du chariot C (Figure VI.7.a). On appuie sur M , le chariot quitte P_1 (moteur droit), arrive en P_2 , puis revient en P_1 (moteur gauche) où il s'arrête jusqu'à ce qu'on appuie sur M à nouveau.
- II. Le chariot effectue sans interruption le cycle aller-retour ($M.P_1.P_2.P_1.M$). Le moteur droit est alimenté entre M et P_2 et le moteur gauche entre P_2 et P_1 , car le chariot rejoint M sous l'action de son propre poids (Figure VI.7.b).

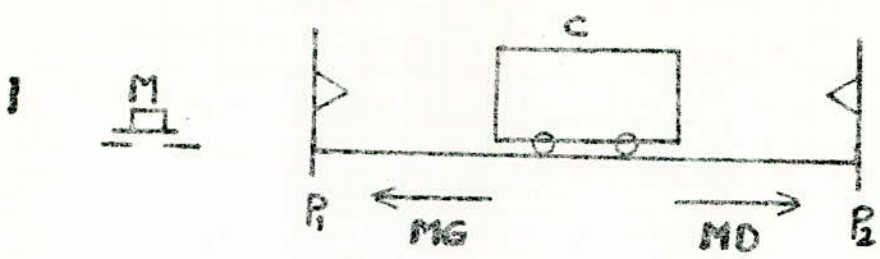


Figure VI-7.a

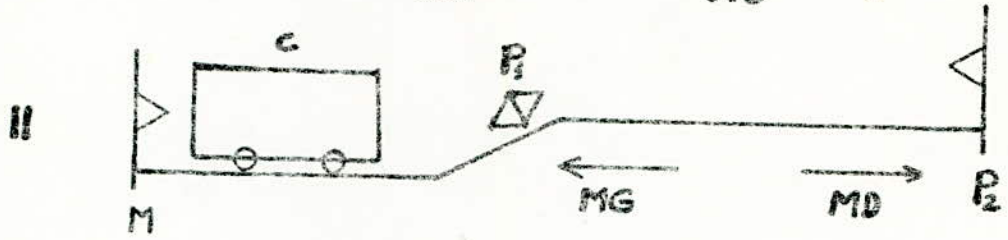


Figure VI-7.b

Ces deux systèmes sont représentés par le même réseau de PETRI (Figure VI-7.c) bien que leurs fonctionnements soient différents.

En effet, un observateur chargé de transcrire les spécifications du système I comme du système II, les traduira par un même réseau de PETRI. Par contre, le concepteur qui veut simplifier la réalisation de l'automatisme, sera amené à connaître des comportements internes non nécessaires à la compréhension du cahier des charges.

Une analyse plus fine des fonctionnements (états des variables internes aux places) montre que seule la machine II est simplifiable (Fig.VI-7.d).

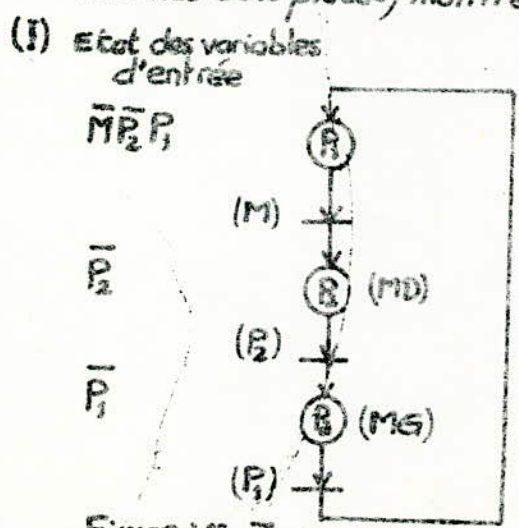


Figure VI-7.c

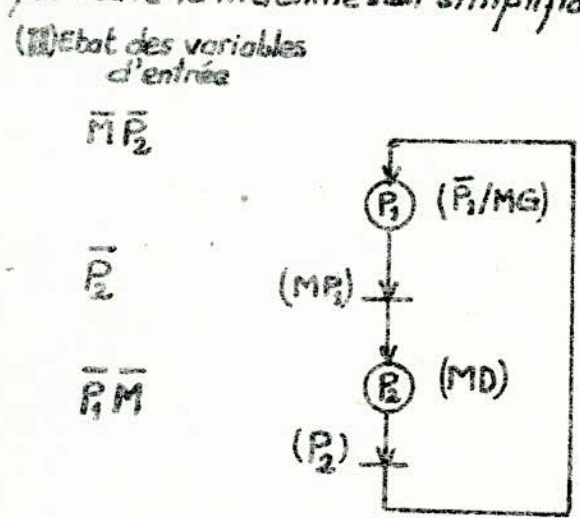


Figure VI-7.d

VI.4.3 Graphes d'état

VI.4.3.a Définition (cf. III.2)

VI.4.3.b Simplifications des sous-graphes d'état

Nous allons simplifier les sous-graphes d'état en recherchant les ensembles maximum de places fusionnables. C'est parmi eux que le concepteur choisira, suivant certains critères, les fusions de places qu'il retient.

Variables d'entrée significatives

Nous savons que, pour simplifier un réseau, des informations complémentaires sont nécessaires. Ces informations, nous allons les introduire sous forme d'état des variables d'entrée aux places.

Une variable du réseau est significative pour un sous-graphe d'état G du réseau si elle prend respectivement les valeurs "0" et "1" en au moins deux places distinctes du sous-graphe G .

Une variable significative permet donc de distinguer au moins deux du sous-graphe. Chaque variable significative va grouper les places du sous-graphe G en deux ensembles en général non disjoints :

$$-P_0(e_k) = \{p_i \in G / e_k = 0 \text{ ou " - " à la place } p_i\}$$

$$-P_1(e_k) = \{p_i \in G / e_k = 1 \text{ ou " - " à la place } p_i\}$$

Avec :

$$P_0 \cap P_1 = \{p_i \in G / e_k \text{ est indifférent " - " à la place } p_i\}$$

P_0 et P_1 représentent deux ensembles de places du sous-graphe d'état G qui ne pourront pas être "distinguées" grâce à l'entrée e_k .

$$\text{On écrit : } G_{e_k} = \{P_0, P_1\}$$

Si on considère deux variables significatives e_1 et e_2 :

$$G_{e_1} = \{P_0, P_1\} \quad \text{et} \quad G_{e_2} = \{P'_0, P'_1\}$$

$$G_{e_1} \cdot G_{e_2} = \{P_0 \wedge P'_0, P_0 \wedge P'_1, P_1 \wedge P'_0, P_1 \wedge P'_1\} = \{P''_0, P''_1, P''_2, P''_3\}$$

P''_0, \dots, P''_3 représentent des ensembles de places non distinguables par les entrées e_1, e_2 .

En procédant ainsi pour toutes les entrées significatives, on obtient :

$$I = G_{e_1} \cdot G_{e_2} \dots G_{e_n} = \{I_1, \dots, I_j\}$$

Chaque ensemble I_i de I (où on a éliminé les ensembles inclus dans d'autres) contient les places du sous-graphe non fusionnables.

VI-4-4 Fusion des places dans un sous-graphe d'état

Fusionner plusieurs places d'un sous-graphe d'état consiste à supprimer ces places du sous-graphe et à les remplacer par une place unique P_i - la place de fusion - à laquelle est associé un événement e_i (construit sur l'alphabet d'entrée) pour chaque place fusionnée p_i .

Les places fusionnées appartiennent au même sous-graphe d'état et sont contenues dans un même compatible. A chaque pas de la fusion, il faut déterminer :

- le nouveau graphe d'état,
- les nouveaux événements associés aux transitions,
- les états des variables d'entrée à la place de fusion,
- les nouvelles fonctions de sortie.

a. Nouveau graphe d'état

Lors de la fusion de plusieurs places, les places fusionnées sont remplacées dans le graphe par une place unique : la place de fusion. Les transitions du réseau (avant fusion) se divisent en :

- Transitions externes :

Elles n'ont pour places de sortie et d'entrée que des places non fusionnées. Ces transitions ne sont pas modifiées par la fusion.

- Transitions internes :

Elles n'ont pour places d'entrée et de sortie que des places fusionnées dans la même place de fusion (et une seule puisque la fusion se fait dans un sous-graphe d'état). Ces transitions n'apparaissent plus dans le nouveau réseau.

- Transitions antécédentes :

Elles ne possèdent aucune place fusionnée parmi les places d'entrée et une seule place fusionnée parmi les places de sortie. Cette place est remplacée par la place de fusion correspondante.

- Transitions subséquentes :

Elles ont comme place d'entrée une place de fusion (qui sera remplacée par la place de fusion) et peuvent posséder une place fusionnée parmi leurs places de sortie (elle sera remplacée par la place de fusion correspondante).

-b. Nouveaux événements associés aux transitions

Les événements associés aux transitions externes et antécédentes des places fusionnées ne sont pas modifiés. Seuls les événements associés aux transitions subséquentes sont affectés par la fusion.

Soit t une transition subséquentes qui avait initialement pour événement associé e et pour place d'entrée p_i .

- Dans le graphe initial, la transition t est franchie (figure VI-8-a)

• Lorsque la place p_i est marquée.

• Dès que l'événement e est vrai.

- Dans le nouveau graphe, après fusion, la transition t est franchie (figure VI-8-b):

- Lorsque la place de fusion P est marquée et l'événement e_i "distinguait" la place p_i (parmi les places fusionnées) est vrai.
- Dès que l'événement e est vrai.

Ce fonctionnement est représenté par un réseau de PETRI où la place p est la place d'entrée de la transition t et où l'événement associé est :

$$E = e.e_i \text{ (Figure VI.8.c)}$$

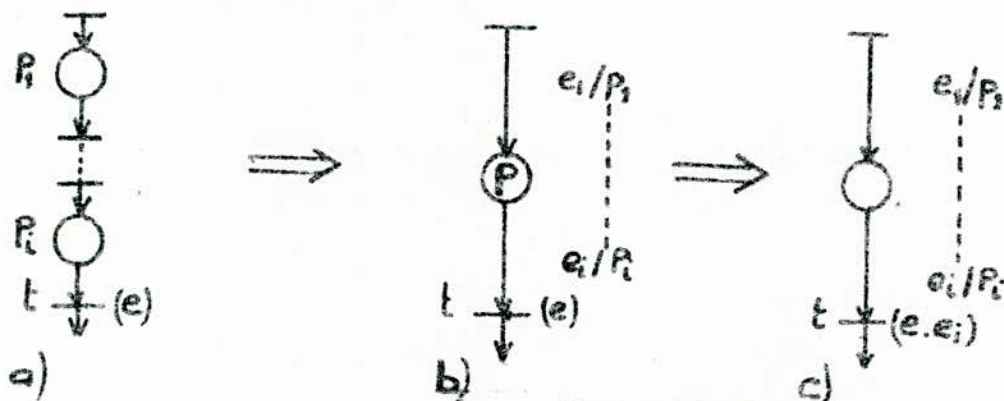


Figure VI.8

En fait, on peut trouver pour E une expression plus simple que le produit booléen $e.e_i$. En effet, les événements e_i sont déterminés de façon à partitionner la place de fusion p en $p_1 \dots p_k \dots p_n$. Ici, pour valider la transition t , il suffit d'un événement e_i distinguant seulement la place p_i des autres places fusionnées.

D'où :

Soit p la place de fusion de n places fusionnées p_1, \dots, p_n . L'événement e associé à une transition subséquente d'une place p_i du graphe initial, sera remplacé dans le nouveau graphe par $E = e.e_i$ où e_i assure à la place p une partition $p_1, p_2, \dots, p_{i-1}, p_{i+1}, \dots, p_n$.

.c. Nouvel état des entrées aux places

L'état des entrées n'est modifié que pour les places fusionnées et la place de fusion. L'état d'une variable d'entrée à la place de fusion sera $O(1)$ si cette variable est à l'état $O(1)$ dans toutes les places fusionnées. Dans tous les autres

cas, la variable devient indifférente.

d. Nouvelles fonctions de sortie

Dans le graphe initial, une sortie était associée à une place (1).

$$S_i = f_i(m) = m(p_{i_1}) + m(p_{i_2}) + \dots + m(p_{i_k}) \quad (1)$$

La fonction de sortie S_i est réalisée pour tout marquage m tel que

$$m(p_i) = 1, \quad i \in \{i_1, \dots, i_k\}$$

Dans le nouveau graphe d'état, les sorties aux places fusionnées sont associées à un couple, place de fusion / événement (2). On écrira :

$$S_i = m(p_1) + \dots + m(p_n) + e_{i_1} \cdot m(p_j) + \dots + e_{i_r} \cdot m(p_r) \quad (2)$$

où :

p_1, \dots, p_n sont des places non fusionnées.

p_1, \dots, p_r sont des places de fusion.

e_{i_1}, \dots, e_{i_r} événement partitionnant les places de fusion p_1, \dots, p_r .

Remarque

Lorsqu'une place de fusion p_j regroupe n places fusionnées $p_{j_1}, p_{j_2}, \dots, p_{j_n}$, nous avons vu qu'il existe un ensemble d'événements $e_{j_1}, e_{j_2}, \dots, e_{j_n}$ partitionnant p_j .

Si une sortie S prend sa valeur en plusieurs places p_{j_1}, \dots, p_{j_k} de p_j , on peut l'écrire :

$$S = m(p_j) (e_{j_1} + e_{j_2} + \dots + e_{j_k})$$

mais on obtiendra une expression plus simple en écrivant :

$$S = m(p_j) \cdot e_{j,1k}$$

où $e_{j,1k}$ est l'événement assurant à la place p_j la partition :

$$p_{j_1}, \dots, p_{j_k} ; p_{j_{k+1}}, \dots, p_{j_n}$$

VI.5. Un exemple d'application

En partant du système étudié au paragraphe V.2.2, nous allons appliquer les différentes notions que nous venons de définir afin de "matérialiser" leur signification.

Le réseau de PETRI de l'automatisme est rappelé sur la figure VI.9.

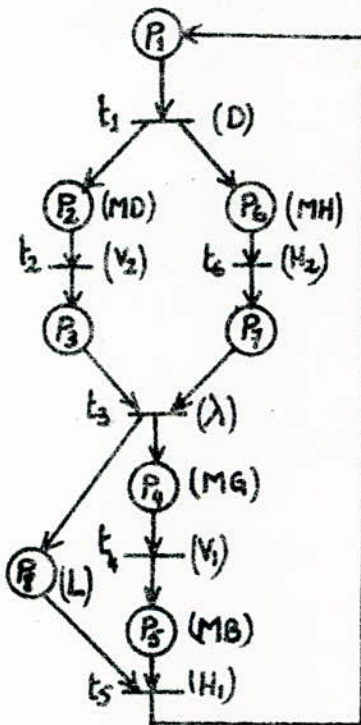


Figure VI.9

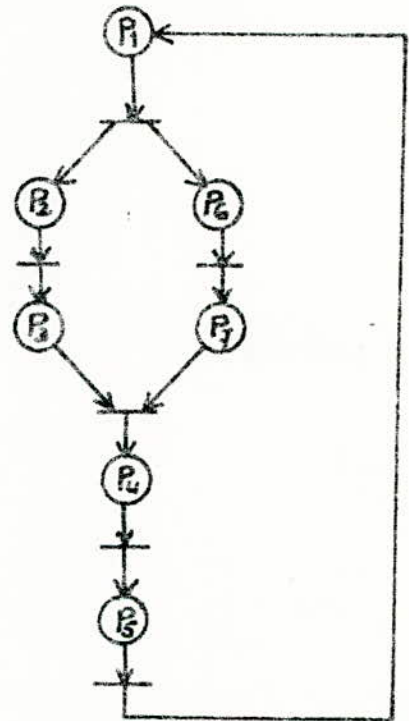


Figure VI.10

VI.5.1. Recherche des places implicites

De façon évidente, seule P_8 l'est (ses places implicantes sont P_4 et P_5). Les sorties associées aux places P_4 et P_5 sont maintenant (MG, L) et (MD, L) respectivement.

VI.5.2. Recherche des sous-graphes d'état

Supprimons dans le réseau de la figure VI.9 tous les arcs entrant et sortant des nœuds ET.

La figure VI.10 met en évidence trois sous-graphes d'état

- (P_2, P_3) : sous-graphe 1
- (P_6, P_7) : sous-graphe 2
- (P_1, P_4, P_5) : sous-graphe 3

VI.5.3. Simplification de ces sous-graphes (deuxième simplification)

Pour cela, il nous faut connaître l'état des entrées aux places. Cet état est donné par le tableau VI-11. Ceci nous permet de représenter les trois sous-graphes comme sur la figure VI-12.

Variable d'entrée	Place
\bar{D}	P_1
\bar{v}_1	P_3, P_6
v_1	P_1, P_5
H_2	P_3, P_6
H_2	P_7, P_4
\bar{v}_2	P_1, P_2
v_2	P_3
\bar{H}_1	P_7, P_4, P_5
H_1	P_1

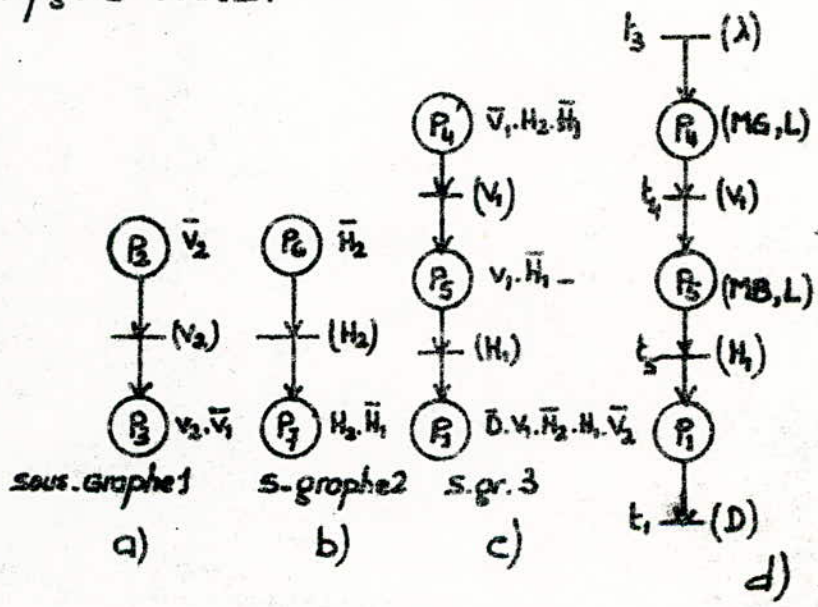


Figure VI-11

Figure VI-12

On voit, pour les sous-graphes d'état 1 et 2 respectivement que v_2 distingue P_2 de P_3 et que H_2 distingue P_6 de P_7 . On peut donc les fusionner.

Cherchons les ensembles I pour le sous-graphe 3 qui présente le plus de difficultés. Il faut d'abord remarquer que les seules variables significatives sont : v_1, H_2 et H_1 .

Chaque variable significative va grouper les places P_4, P_5 et P_1 en deux ensembles P_6 et P_7 .

$$\begin{array}{lll}
 \bullet H_1 & P_1(M) = \{P_1\} & \bullet H_2 & P_2(M_2) = \{P_3, P_4\} & \bullet H_3 & P_3(M_3) = \{P_5\} \\
 & P_1(M) = \{P_1, P_2\} & & P_3(M_3) = \{P_4, P_5\} & & P_1(M_3) = \{P_1\}
 \end{array}$$

On trouve alors :

$$G_{H_1} = (P_1; P_1, P_2)$$

$$G_{H_2} = (P_3, P_4; P_3, P_4)$$

$$G_{H_3} = (P_5; P_5)$$

$$\begin{aligned}
 G_E = G_{H_1} \cdot G_{H_2} &= \{(P_1; P_3, P_4) \cdot (P_3, P_4; P_3, P_4)\} = \\
 &= \{(P_1 \wedge P_3, P_4), (P_4 \wedge P_3, P_3), (P_3, P_4 \wedge P_3, P_4), (P_3, P_4 \wedge P_3, P_4)\} \\
 &= \{P_4; P_3, P_4\}
 \end{aligned}$$

$$\begin{aligned}
 G_E \cdot G_{H_3} &= (P_4; P_3, P_4) \cdot (P_5; P_5) = (P_4 \wedge P_5, P_3, P_4), (P_4 \wedge P_5, P_3), (P_3, P_4 \wedge P_5, P_4), \\
 &\quad (P_3, P_4 \wedge P_5, P_4) = (P_4; P_3, P_4)
 \end{aligned}$$

D'où l'on tire : $I = G_{H_1} \cdot G_{H_2} \cdot G_{H_3} = (P_4; P_3, P_4)$

Donc p_1, p_4 et p_5 sont fusionnables en une seule place.

On remarque alors que :

- les transitions externes de ce sous-graphe sont t_2 et t_6 ,
- les transitions internes t_4 et t_5 ,
- les transitions antécédentes se limitent à t_3 ,
- les transitions subséquentes se limitent à t_1 .

Dans le réseau primitif (Figure VI-10) t_1 était franchie lorsque p_1 étant marquée, D était vrai. Dans le réseau simplifié, t_1 sera franchie lorsque 3 (place de fusion) étant marquée, H₃ étant rencontré (puisque H₃ = t est l'entrée qui distingue p₁), D sera vrai.

Le réseau de la figure VI-10 devient celui de la figure VI-13.

Quelques remarques :

- L'état des entrées à la place 3 est indifférent;

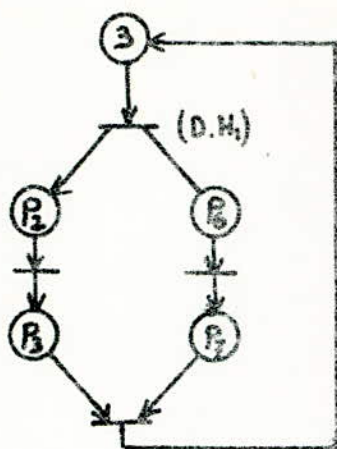


Figure VI.13

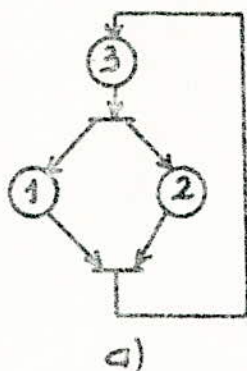
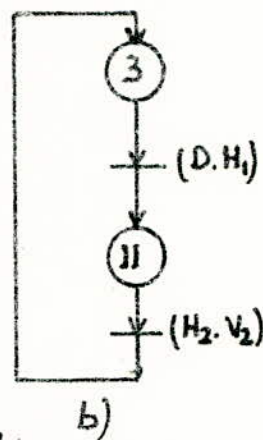


Figure VI.14



b)

• les sorties en 3 deviennent :

- * $MG = m(3) \cdot \bar{V}_1$
- * $MB = m(3) \cdot V_1 \cdot \bar{H}_1$
- * $L = m(3) \cdot \bar{H}_1$

On obtient le résultat simplifié de la figure VI.14 - a sur lequel on remarque que 1 est implicite de 2. Le réseau final est donc celui de la fig.VI.14.b) Les équations des sorties correspondantes sont indiquées ci-dessous :

- * $MH = m(II) \cdot \bar{H}_2$
- * $MD = m(II) \cdot \bar{V}_2$

VI.6 - Processus de simplification

Nous allons résumer le processus de simplification d'un réseau de Petri

1. Apport d'informations complémentaires (états des variables d'entrée aux places).

2. Recherches :

- des places implicites : suppression de ces places et report des sorties sur les places implicantes.

- des transitions implicites : suppression de ces transitions

(ou regroupement des transitions identiques).

3. Décomposition du réseau en sous-graphe d'état.

4. Pour chaque sous-graphe d'état :

- construction des ensembles maximum de places fusionnables,
- choix d'une fusion,
- détermination du nouveau réseau de PETRI. Retour en 2.

VI.7. Conclusion

On peut connaître systématiquement toutes les possibilités de réduction d'un réseau. Quant à les utiliser toutes, le concepteur est seul juge de l'intérêt et du choix des fusions.

La recherche d'un réseau réduit - comme d'une table - repose sur l'idée que simplifier une réalisation, c'est la rendre la moins séquentielle possible (d'où la réduction du nombre d'états).

Matérialisation de systèmes décrits par réseau de PETRI

VII.1. Introduction

On cherche dans les méthodes de synthèse des systèmes séquentiels asynchrones à se ramener à un circuit combinatoire avec boucles de réaction (mémorisation) selon le schéma classique de la figure VII.1. De plus, la réalisation technologique utilisant des composants discrets, le critère de réduction du coût de l'équipement est - ou plutôt était - la minimisation du nombre de variables internes (mémoires).

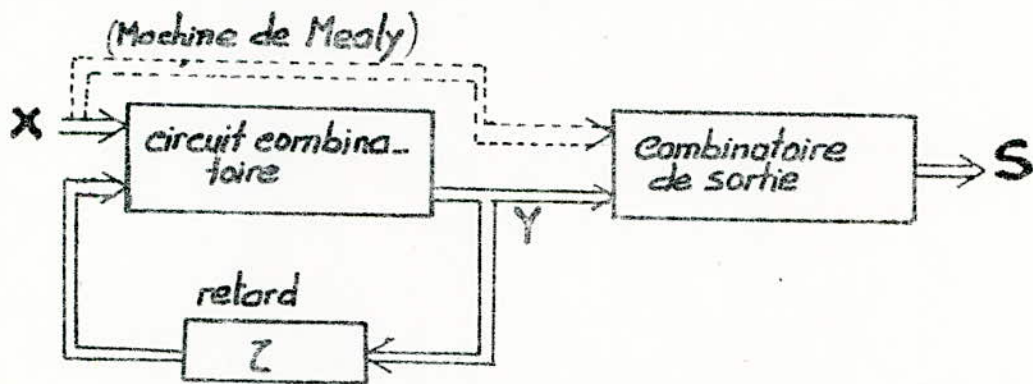


Figure VII.1

Les derniers progrès de la technologie (coût, performances) permettent en effet, d'utiliser un composant de base séquentiel : la « mémoire », que celle-ci soit composant discret, intégrée dans un module ou dans un programme. La structure d'une telle réalisation - organisée autour d'un circuit séquentiel - est représentée figure VII.2.

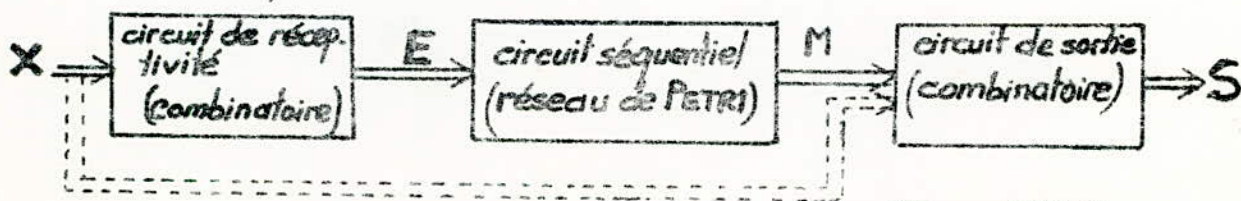


Figure VII.2

Le réseau de PETRI est au circuit séquentiel ce que le schéma à relais est au circuit combinatoire. Dans ce sens, on peut affirmer - et nous insistons sur ce point - que le réseau de PETRI constitue le document unique pour la synthèse, du cahier des charges à la réalisation technologique (séquentielle, évidemment).

L'approche que nous avons choisi d'exposer ici est résolument modulaire et ceci pour deux raisons :

- le fait que la modularité permet d'assurer une grande répétitivité et un coût assez faible,
- elle seule garantit une approche séquentielle de problèmes séquentiels, telle que nous venons de la définir.

VII.2 Les principaux modules

La matérialisation modulaire d'un système décrit par un réseau de PETRI est basée sur deux idées assez différentes des autres.

- **La première**, c'est que les modules utilisés ne seront pas identiques (comme c'est le cas pour la plupart des autres méthodes) : on va les définir par des fonctionnements correspondant à des structures élémentaires du réseau de PETRI. Nous allons définir cinq modules correspondant aux cinq sous-graphes élémentaires de la figure VII-3 et qui sont suffisants pour reconstruire n'importe quel réseau séquentiel. Ainsi, le réseau de la figure VII-4-a peut être "décomposé" suivant la figure VII-4-b.

On voit donc apparaître trois modules "transfert" et un de chacun des autres types.

- **La deuxième** idée sous-jacente à ces modules est celle d'appel-réponse. Le principe en est que chaque module comportera un ensemble de voies de commande (voies d'entrée et de sortie) à deux lignes : une ligne d'appel et

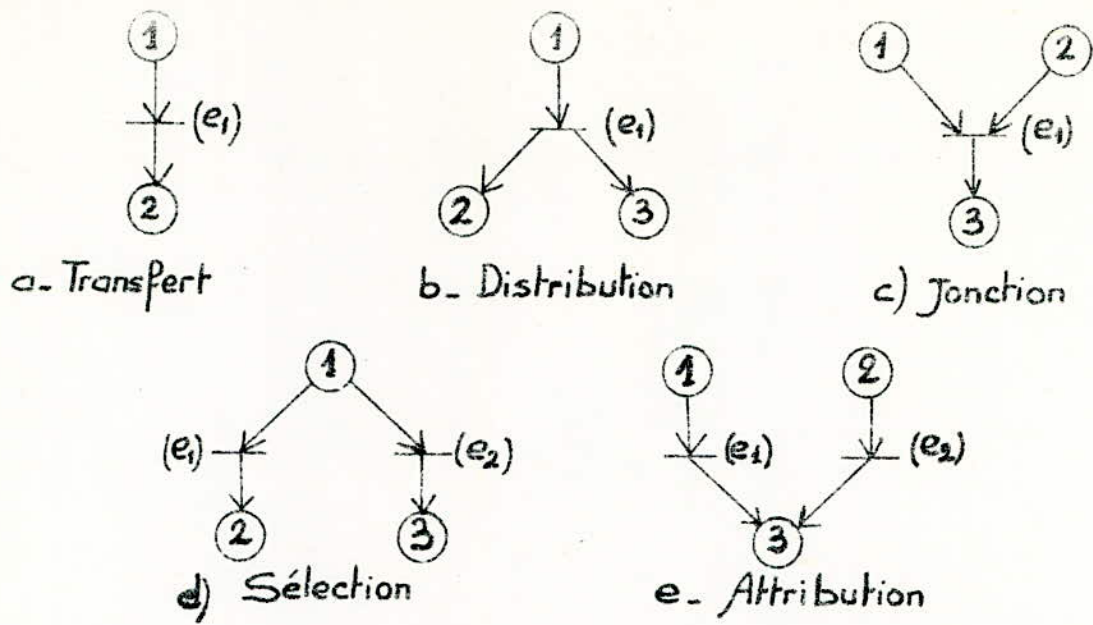


Figure VII.3

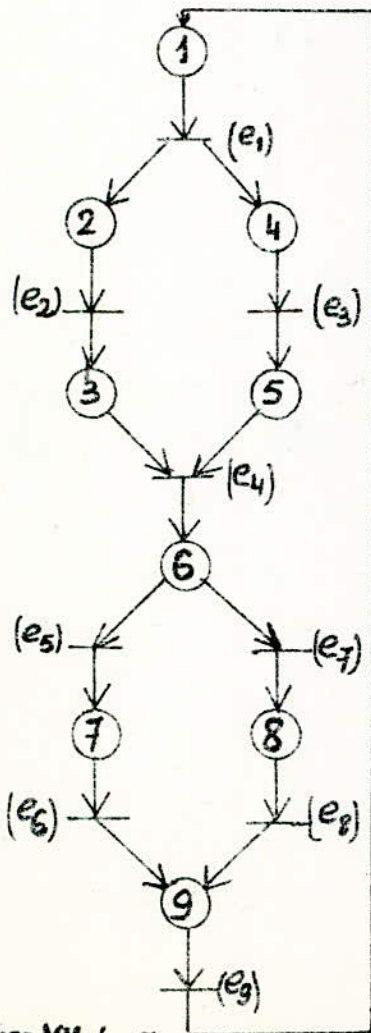


Figure VII.4.a

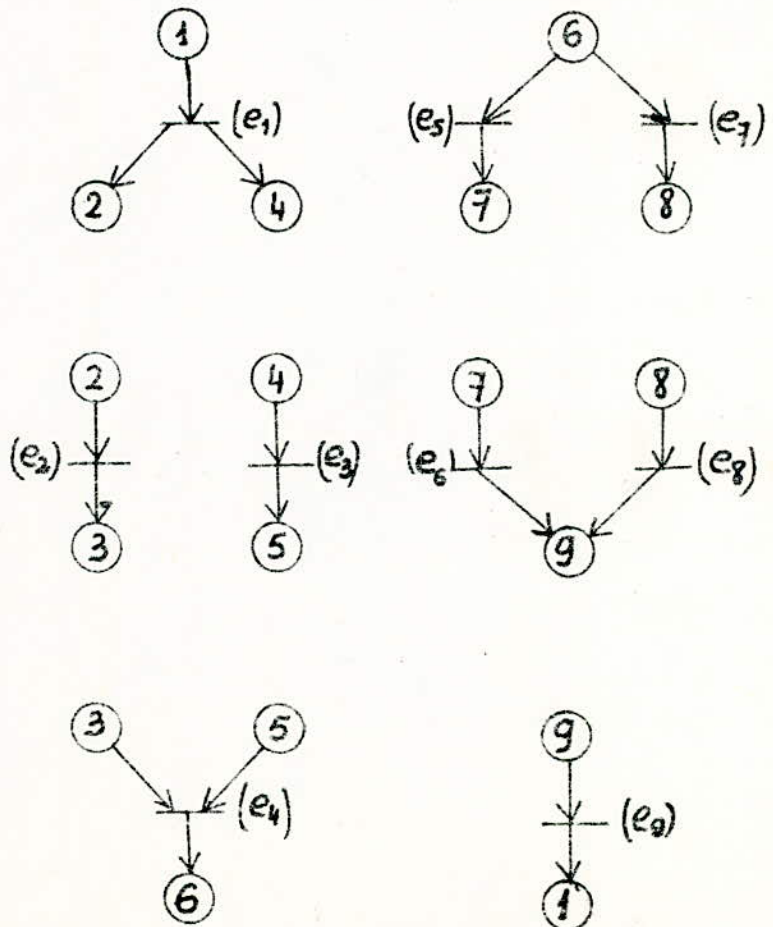


Figure VII.4.b

une ligne de réponse. Chaque voie peut prendre deux états : actif et inactif.

Un module est initialisé lorsque sa fonction de commande est "réalisable" - c'est-à-dire un ensemble de voie(s) d'entrée active(s) -, il effectue alors un "travail" qui consiste à rendre actif un ensemble de voie(s) de sortie. (ces deux ensembles de voies - entrée et sortie - caractérisent la fonction de commande du module).

Le retour à l'inactivité pourra se faire suivant des règles que nous expliciterons plus bas. Il faut noter que ces voies permettent le transfert d'information entre modules. Par contre, il existera des entrées "conditionnelles" qui auront comme fonction d'autoriser ce transfert et d'exciter ou d'inhiber le module lui-même auquel elles sont appliquées.

Nous allons donc examiner d'abord une définition structurelle plus poussée des modules (en voyant, en particulier, les problèmes de retour à l'état inactif). Nous nous occuperons ensuite de leur matérialisation et ceci nous conduira à envisager, de façon plus générale, le problème de la structure d'un automatisme séquentiel.

VII.3 Définition structurelle des modules

Dire qu'un module réalise l'opérateur "jonction" ou "transfert" comme cela est indiqué sur la figure VII.3, c'est donner, si l'on ose dire, une représentation "macroscopique" de la réalité. Cette définition n'est, malheureusement, pas opérationnelle au niveau des composants de base dont nous pouvons avoir besoin pour matérialiser le module correspondant. On doit donc s'intéresser à une représentation plus fine de la fonction de ces modules. D'autre part, si nous avons défini le passage de l'inactivité à l'activité pour les modules nous n'avons pas explicité le passage inverse. C'est ce point qui sera traité le premier dans ce paragraphe.

VII. 3.1. Les deux types de modules

C'est sur l'exemple du module "jonction" que nous allons expliquer les deux façons essentielles dont on peut concevoir le retour à l'inactivité d'un module. Convenons qu'un module sera symbolisé par le schéma de la figure VII.5. a : Une "boîte noire" porte l'initiale du module; les voies d'entrée

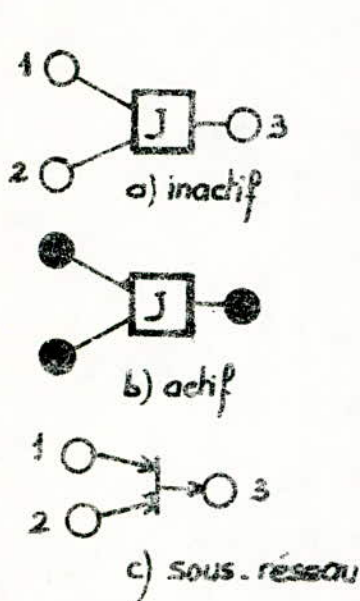


Figure VII.5

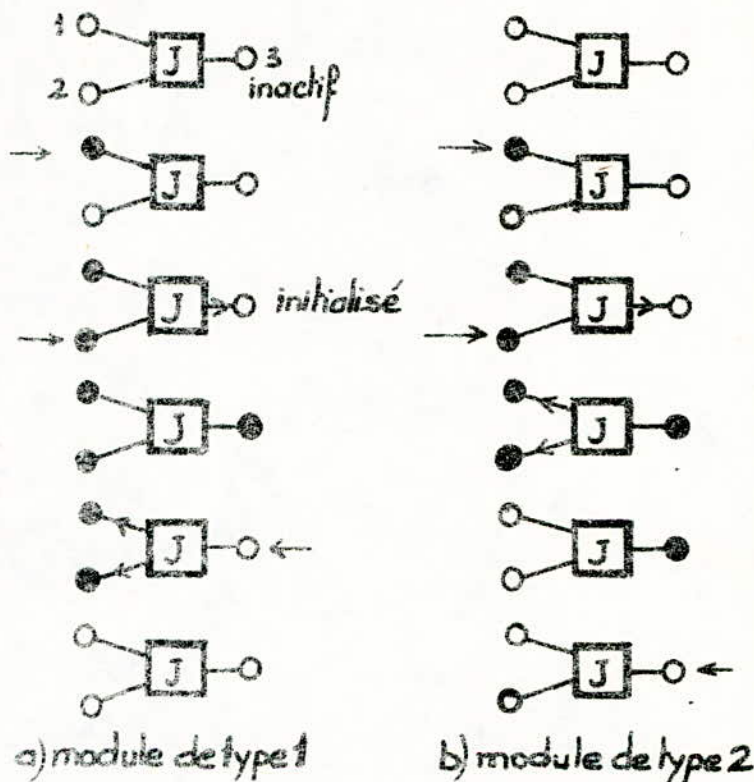


Figure VII.6

et de sortie sont symbolisées par des cercles blancs si le module est inactif, noirs si le module est actif. Une flèche dirigée vers une voie de commande indique l'action d'un agent extérieur.

On a représenté, sur la figure VII.6 le fonctionnement de deux types de modules.

Au départ, le module jonction est inactif. Ceci revient à dire que les places d'entrée (associées aux voies) du sous-réseau de la figure VII.5.c

ne sont pas marquées. Puis le franchissement d'une transition amène au marquage de la place 1 - c'est-à-dire à l'activation de la voie 1. L'activation de la voie 2 (= le marquage de la place 2) provoque l'initialisation du module (la validation de la transition).

À partir de là, le fonctionnement des deux types de module diffère.

- Pour le module de type 1, il est activé et le restera jusqu'à ce qu'une action extérieure désactive la voie de sortie (3) ce qui provoquera la désactivation des voies d'entrée et le retour à l'état initial.

- Pour le module de type 2 c'est l'activité de la voie de sortie (3) qui provoque l'inactivité des voies d'entrée (1 et 2). Il faut alors l'action d'un agent extérieur pour désactiver la voie de sortie et ramener le module à son état initial.

D'après ce que nous venons de dire, le fonctionnement d'une structure modulaire de type 1 se fait par "remplissage et vidage" de chemins - ce qui a l'inconvénient d'introduire des retards de propagation.

Au contraire, le fonctionnement d'une structure modulaire de type 2 se fait par "transfert d'information" entre modules et la structure bouclée reste très proche du réseau de PETRI qu'elle matérialise.

Ce sont les modules de type 2 que nous choisissons et étudierons par la suite. Cela n'a rien d'étonnant dans la mesure où ils constituent une adaptation aux automatismes séquentiels des modules de type 1 conçus, eux, pour une application "informatique" de réseaux de PETRI.

VII.3.2. Fonctionnement détaillé

Si nous reprenons le fonctionnement détaillé des modules de façon analogue à ce que nous venons de faire, ceci met en évidence que le module peut déjà être décrit par un "micro-réseau" de PETRI. Ainsi, au module

"jonction" de type 2 on peut associer la représentation de la figure VII.7. A l'état de chaque voie (1,2,3) est associée une paire de places (a, r) correspondant à l'état (actif ou inactif) de la ligne d'appel et de la ligne de réponse. On remarquera que les seules relations de ce sous-réseau avec l'extérieur se situent au niveau du marquage de ces six places. Par exemple, a_1 et a_2 étant marquées, le marquage de a_3, r_1 et r_2 se fera sans action extérieure (e_1 ne fait qu'autoriser ce marquage). Le retour en configuration initiale (celle de la figure VII.7) inactive se fera grâce au marquage de r_3 (en supposant que a_1 et a_2 ne soient point marquées entre temps, bien sûr...)

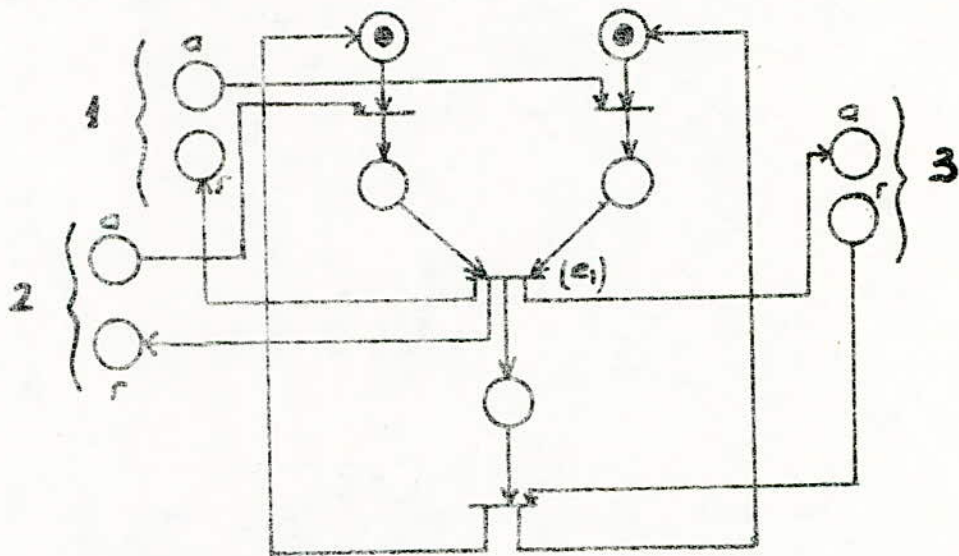


Figure VII.7

De façon analogue, le module "distribution" sera rendu actif (c'est à dire que ses places de sortie a_2 et a_3 seront marquées) lorsque la ligne d'appel de la voie d'entrée a_1 sera activée (place a_1 marquée). Le marquage de a_2 et a_3 provoquera alors le marquage de r_1 (activation de la ligne de réponse de la voie d'entrée) et il faudra que les places r_3 et r_2 (correspondant aux lignes de réponse de la voie de sortie) soient marquées pour permettre le retour à l'état inactif.

VII-3.3 Les possibilités de réalisation

On a deux modes de réalisation :

a. le premier mode consiste à représenter l'état d'une voie de commande par un codage des lignes d'appel-réponse. Ainsi, une voie de commande sera inactive si ces deux lignes portent le même signal binaire (soit 00, soit 11) et active si elles portent des signaux différents.

b. le deuxième mode de fonctionnement consiste à matérialiser l'état de la voie par celui d'une mémoire Y_i dont l'excitation est a_i et la remise à zéro r_i . Ainsi, considérons le module "jonction" dont le graphe est rappelé sur la figure VII-8. a et le schéma bloc donné sur la figure VII-8. b.

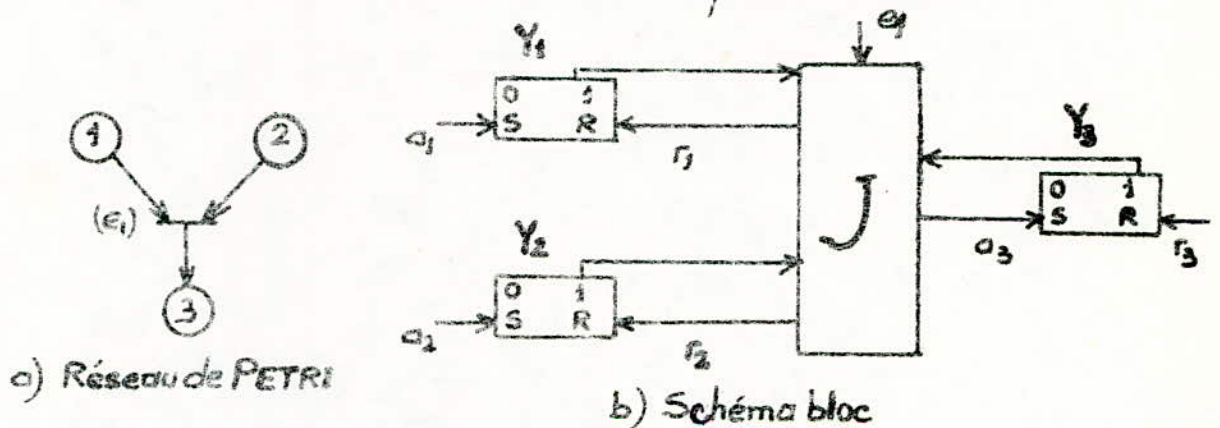


Figure VII.8

La mémoire Y_3 devra être mise à 1 lorsque les mémoires Y_1 et Y_2 seront actives et que sera vraie l'entrée e_1 (en terme de réseau de PETRI: on marquera la place 3 lorsque, 1 et 2 étant marquées, e_1 sera vrai). Autrement dit

$$a_3 = e_1 \cdot Y_1 \cdot Y_2$$

De même Y_1 et Y_2 seront remis à zéro par:

$$r_1 = r_2 = e_1 \cdot Y_3$$

Ceci étant vu, nous allons maintenant nous préoccuper de la réalisation des 5 modules déjà définis. On a indiqué, sur le tableau de la figure VII-9, le réseau de PETRI de chacun des 5 modules choisis, le schéma bloc correspondant et les équations qu'on peut en tirer.

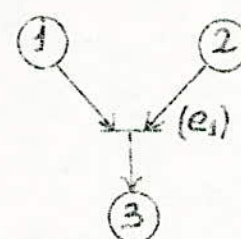
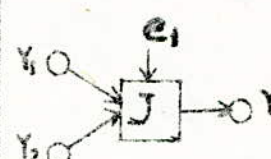
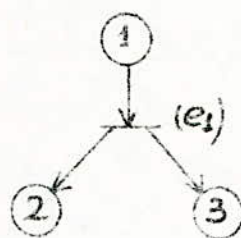
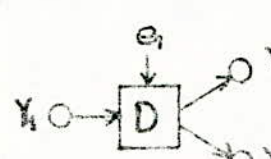
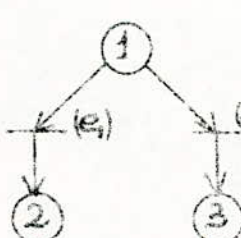
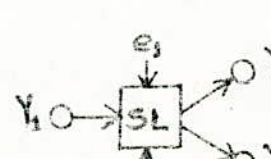
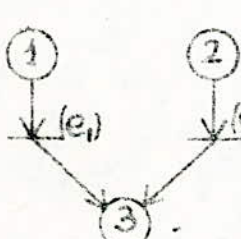

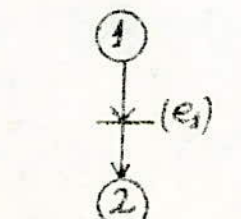
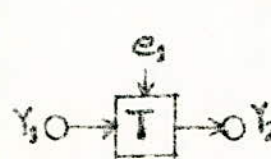
Nom du module	Réseau de Petri	Schéma bloc symbolique	Equations
JONCTION			$a_3 = e_1 \cdot Y_2 \cdot Y_1$ $r_1 = r_2 = e_1 \cdot Y_3$
DISTRIBUTION			$a_2 = a_3 = e_1 \cdot Y_1$ $r_1 = e_1 \cdot Y_2 \cdot Y_3$
SELECTION			$a_2 = e_1 \cdot Y_1$ $a_3 = e_2 \cdot Y_1$ $r_1 = e_1 \cdot Y_2 + e_2 \cdot Y_3$
ATTRIBUTION			$a_3 = e_1 \cdot Y_1 + e_2 \cdot Y_2$ $r_1 = e_1 \cdot Y_3$ $r_2 = e_2 \cdot Y_3$
TRANSFERT			$a_2 = e_1 \cdot Y_1$ $r_1 = e_1 \cdot Y_2$

Figure VII.9 : Récapitulation des modules

Réalisations « Logiques programmées » : Une première approche

VIII.1 Introduction

Il était prévisible que le dernier chapitre serait consacré aux réalisations logiques programmées, d'abord parce qu'il n'est plus possible de les ignorer davantage et ensuite parce qu'elles viennent plaider pour l'utilisation d'un outil séquentiel de représentation (Réseau de PETRI).

Dans le domaine des réalisations technologiques on constate une progression des « logiques programmées » aux dépens des « logiques câblées ».

L'apparition des calculateurs industriels avait laissé penser qu'on disposait là d'un outil souple, puisque programmable, capable de concurrencer valablement pour réaliser des automates logiques, les systèmes câblés, handicapés par leur trop grande rigidité. Les problèmes posés par l'étude purement logique des systèmes n'étaient pas résolus mais il semblerait plus simple d'introduire un ruban perforé dans un lecteur que de prendre le fer à souder pour modifier un câblage.

Nous allons donc examiner tout d'abord la réalisation des réseaux PETRI sans fonctionnements parallèles (graphes d'état) puis nous envisagerons les possibilités de réalisation des réseaux avec évolutions simultanées. Nous verrons également un produit industriel caractéristique de l'état actuel des techniques : l'automate programmable.

Nous allons rapidement parler de deux types de circuits, essentiels pour les réalisations programmées et où les progrès ont été spectaculaires : les mémoires et les microprocesseurs.

VIII.2 Les mémoires à circuits intégrés

La mémoire est le composant de base qui permet l'enregistrement d'un programme.

On distingue les mémoires suivantes :

- **R.A.M.** (Random Access Memories) : accès aléatoire, à lecture-écriture (mémoire vive), statiques (flip-flop) ou dynamiques (charge d'une capacité en technologie MOS). Elles sont l'équivalent, du point de vue fonction, des mémoires à tores de ferrites mais présentent l'inconvénient d'être volatiles (perte de l'information mémorisée en cas de coupure d'alimentation).

- **R.O.M.** (Read Only Memories) : accès aléatoire, à lecture seulement (mémoire morte), statiques, non volatiles (à diodes ou transistors). Elles sont programmées une fois pour toutes par le fabricant (elles sont l'équivalent des mémoires câblées).

- **P.R.O.M.** (programmable R.O.M) : ce sont des R.O.M programmables une fois pour toutes par l'utilisateur.

- **R.E.P.R.O.M.** (reprogrammable R.O.M)

- **C.A.M.** (content adressable Memories) ou mémoires associatives : ce sont des mémoires R.A.M adressables par leur contenu.

- **P.L.A.** (Programmable Logic Array) : ce sont des R.O.M où la mémoire est programmable.

- **Piles** : empilage de registres, accès séquentiel. On distingue les piles **F.I.F.O.** (first in - first out) où la première information introduite est la première extraite et **L.I.F.O.** (Last in - first out) où la dernière information introduite est la dernière extraite.

VIII. 3. Les microprocesseurs^(*)

Bien qu'apparus récemment, les microprocesseurs en sont déjà à leur troisième génération. Après une première génération travaillant en multiplexage (addition en 10 μ s) sont arrivés des microprocesseurs comportant davantage de fonctions intégrées - dont les interruptions - (INTEL 8080, MOTOROLA 6800) et avec de meilleures performances (addition en 2 μ s); enfin, on

trouve aujourd'hui des circuits où l'on cherche à diminuer la consommation ou bien à augmenter la densité d'intégration ou encore la rapidité, tel que le **TEXAS TMS 9900** - 16 bits - ou la série **INTEL 3000** - par tranches de 2 bits - (addition en moins de $0,5 \mu s$). Et on annonce des microprocesseurs^(*) encore plus rapides avec encore plus de fonctions intégrées.

VIII.4 Réalisations «logiques programmées» des réseaux de PETRI (RdP) sans fonctionnements parallèles (graphes d'état)

VIII.4.1. Réalisations à l'aide de mémoires mortes

Un graphe d'état ne possède qu'une place marquée à la fois. On peut donc, en codant chaque place, caractériser l'état de la machine à un instant donné par le code de la place marquée à cet instant. L'état initial de la machine est repéré par le code de la place initialement marquée.

La structure de la réalisation programmée (figure VIII.1) comporte :

- Un registre contenant l'état présent de la machine (place marquée)
- Une matrice des réceptivités est un circuit qui détecte l'occurrence d'un événement auquel le système est réceptif dans l'état présent. Elle détermine également, en cas de sélection, quel est l'état suivant correct (correspondant)
- des mémoires mortes (R.O.M) permettant de connaître le code des états suivants et l'état des sorties.
- une logique interne qui assure les échanges d'information entre les divers blocs.

(*) Un excellent résumé des caractéristiques des microprocesseurs et des problèmes posés par leur utilisation peut être trouvé dans **ELECTRONICS** vol. 49, n° 8, April 1976

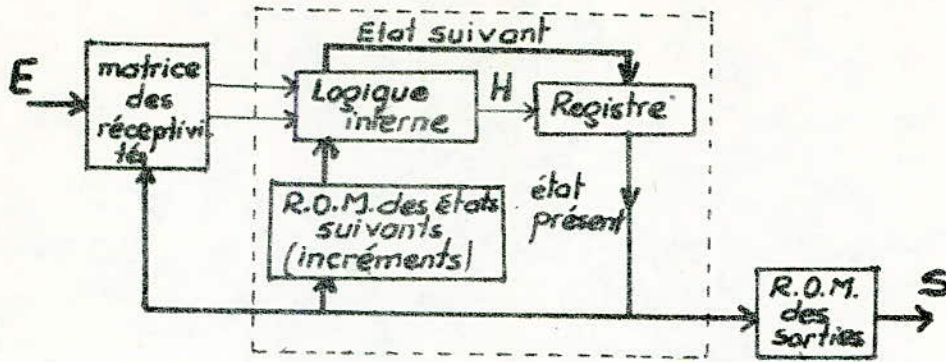
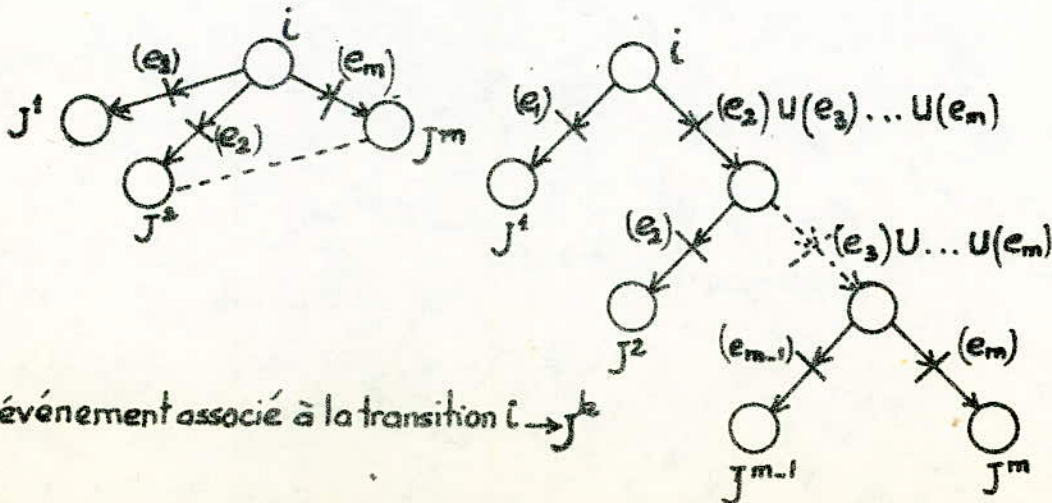


Figure VIII.1

Afin de faciliter les réalisations et de réduire la taille des R.O.M. On peut simplifier le schéma précédent. Nous adoptons pour le GTE les règles suivantes :

- a. dans un graphe d'état (G.E), un état a au maximum deux états suivants (c'est-à-dire chaque place a au maximum deux transitions de sortie).
- b. On code chaque état (place) i par un entier $[i]$ tel que :
 - $0 \leq [i] < N$ ($N =$ nombre d'états (de places du réseau)).
 - Si i a deux états suivants, l'un d'eux est codé $[i] + 1$.

La règle a. exige des modifications du graphe, faciles à mettre en œuvre comme par exemple celles indiquées sur la figure VIII.2



(e_k) = événement associé à la transition $i \rightarrow j^k$

Figure VIII.2

La règle b. nécessite l'application d'un algorithme de codage

Dans la matrice des réceptivités, sortent deux variables binaires $VAL1$ et $VAL2$ qui indiquent l'occurrence d'un des deux événements attendus ($val1 \cup val2 = 1$):

- Si $VAL1 = 1$ code de l'état suivant = code de l'état présent + 1

- Si $VAL2 = 1$ code de l'état suivant = code de l'état présent + incrément.

La R.O.M. des états suivants est alors remplacée par celle des incréments. Pour chaque état présent, elle indique la quantité (incrément) qu'il faut ajouter au code de cet état pour obtenir le code de l'état suivant.

VIII.4.2. Réalisation programmée sur microprocesseur

Dans une réalisation « logicielle » où le temps intervient au niveau de l'exécution du programme, on traduira le réseau de PETRI en un programme - que nous appellerons programme « utilisateur » - dont, à un instant donné, on n'exécutera que la partie spécifique à l'état du réseau à cet instant.

Il s'agit donc, d'une part, de préciser la façon de traduire le réseau de PETRI en un programme, c'est-à-dire la structure et le langage du programme « utilisateur » et, d'autre part, de concevoir un programme « interpréteur » chargé d'interpréter et d'exécuter les instructions du programme « utilisateur ».

VIII.5. Réalisations « logiques programmées » des RdP avec évolutions simultanées

VIII.5.1. Réalisation à partir de mémoires mortes.

Si on considère à présent des réseaux de PETRI conformes, avec évolutions simultanées, la méthode va consister :

- à recouvrir le RdP par des graphes d'état (GE) de façon à ce que la coordination des événements et des conditions associées au réseau de PETRI (RdP) initial soit inchangée.

— Une fois cette décomposition faite, à synthétiser séparément chaque graphe d'état à l'aide de mémoires mortes.

VIII.5.2 Réalisation programmée sur microprocesseur^(*)

VIII.6 Automate programmable

Nous avons tout d'abord utilisé les réseaux de PETRI, en leur donnant une certaine interprétation, pour formaliser le cahier des charges d'un automatisme logique, puis envisager des réalisations modulaires, programmées — où le composant de base n'est plus un circuit combinatoire mais séquentiel (la mémoire) — Le réseau de PETRI devient ainsi le document unique de la synthèse, des spécifications jusqu'à la matérialisation technologique finale.

Nous allons présenter ici un automate programmable (encore appelé contrôleur programmable, séquenceur programmable ...) qui par son principe et sa réalisation est fondamentalement différent des automates programmables existants.

Qu'est-ce qu'un automate programmable ? Pour reprendre la définition de P. ROMAND : "c'est une machine capable de recevoir les informations d'état d'un processus, caractérisé par des signaux "tout ou rien" et d'en générer, par un programme défini les informations de commande, également par signaux tout ou rien. Le programme décrit les relations booléennes et temporelles reliant les informations d'entrée et celles de sortie. Cette description s'effectue par des moyens non technologiques, à caractère général, susceptibles d'être modifiés et mémorisés."

VIII.6.1 Un automate programmable par réseau de PETRI : COLERES (Coordinateur Logique d'Évolutions représentées Séquentiellement)

Il peut être caractérisé en trois points :

(*) Revue générale de l'Électricité. T. 85 n° 3, Mars 1976.

1. Il constitue un outil d'aide à la conception d'un système logique dans la mesure où :
 - le niveau des données à fournir se situe le plus en amont possible dans le processus de la conception, c'est-à-dire les spécifications mises sous forme de réseau de PETRI (ou autre outil séquentiel).
 - la phase de synthèse est purement supprimée. Le concepteur peut apporter des spécifications évidentes (en utilisant des formulations combinatoires générales ou locales).

2. L'exécution du programme est "sélective" c'est-à-dire au lieu d'une exploration séquentielle systématique des programmes, seuls les points sensibles (ceux susceptibles d'évolution) sont examinés. Ceci entraîne une structure particulière à deux niveaux :

- un niveau de recherche des points sensibles, c'est-à-dire des places marquées (il peut y en avoir plusieurs dans le cas de fonctionnements parallèles ou de plusieurs programmes implantés).
- un niveau d'exécution des instructions sélectionnées.

3. Cet automate programmable a un rôle de coordination de tâches. Ses entrées - sorties sont binaires et les opérations qu'il effectue des opérations logiques

VIII-7 Conclusion

On a exposé au cours de ce chapitre, une approche des réalisations par « logiques programmées ».

Cette approche nous a permis de concilier les avantages des représentations par réseau de PETRI

L'emploi des mémoires mortes réduit au maximum les problèmes de câblage : toutes les réalisations sont faites avec les mêmes circuits, seules les mémoires sont à programmer différemment.

Conclusion

Les exemples que nous avons traités mettent en évidence la richesse d'utilisation des réseaux de PETRI pour la synthèse des systèmes séquentiels asynchrones et sa supériorité par rapport à la représentation de type d'Huffman.

En conclusion, nous allons énumérer quelques uns des avantages que l'on peut attribuer à l'emploi des réseaux de PETRI pour représenter le fonctionnement d'un automatisme séquentiel.

D'abord par sa simplicité et sa facilité d'interprétation, cet outil, très proche du cahier des charges (aussi proche qu'un organigramme), est accessible et c'est là l'important à tous techniciens ou non techniciens. D'une grande souplesse, il permet des modifications locales sans remettre en cause l'ensemble du réseau déjà défini. Ceci est très intéressant puisqu'un cahier des charges est sujet à de nombreuses modifications.

De plus, sa capacité à représenter des fonctionnements parallèles et des sous-programmes en fait un outil puissant. Et il peut, de plus, suivant la façon de le matérialiser, devenir un document pour la réalisation technologique.

La formulation d'un problème, à l'aide des réseaux de PETRI, correspond bien à la démarche naturelle du concepteur qui, à chaque pas, cherche à savoir quels événements sont susceptibles de faire évoluer le système.

La réduction se fait par étapes. Ainsi, il est plus simple de réduire un réseau de PETRI par apport d'informations complémentaires que d'établir le diagramme de fluence. Le concepteur bénéficiera alors d'une réduction partielle.

Enfin, nous avons cherché, tout au long de ce projet à élaborer une approche uniforme de la synthèse des systèmes logiques, que la réalisation en soit câblée ou programmée.

De nombreux travaux se poursuivent et restent à poursuivre sur les réseaux de PETRI, mais on peut penser qu'une brèche est ouverte et que la

complexité des problèmes traités s'accroissant sans cesse, le logicien sera amené à adopter un outil nouveau (tel que le réseau de PETRI) et naturellement à bouleverser les idées qu'il avait quant à la réalisation des systèmes logiques.

Cet outil de description à la fois clair et rigoureux serait donc le bienvenu.

T.P. n°1

Mémoire à niveau

On désire commander l'allumage ou l'extinction d'une lampe L à l'aide de deux boutons poussoirs m et a. Initialement la lampe L est éteinte et les deux boutons poussoirs relâchés. Son allumage est obtenu par action fugitive sur le bouton poussoir m, le bouton poussoir a étant relâché. Son extinction est obtenue par action sur le bouton poussoir a. Effectuer la synthèse en utilisant des portes NAND.

Solution

- Le schéma fonctionnel est représenté par la figure 1.
- Nous adoptons les conventions suivantes:

Entrée: l'état appuyé des boutons poussoirs est repéré par $m=1, a=1$, l'état relâché des boutons poussoirs par $m=0, a=0$.

Sortie: l'état de la grandeur de commande L correspond à $L=1$ allumage $L=0$ extinction.

Alphabet d'entrée $\{ \bar{m}\bar{a}, \bar{m}a, m\bar{a}, ma \}$

Alphabet de sortie $\{ L, \bar{L} \}$

- L'état initial : boutons poussoirs relâchés, lampe éteinte ; est représenté par la place $p_1(\bar{L})$ marquée (figure 2).

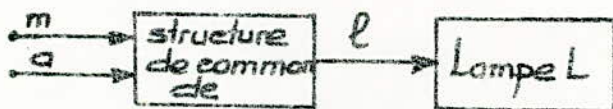


Figure 1



Figure 2

- L'événement d'entrée susceptible de faire évoluer le marquage initial M_0 correspond à $m\bar{a}$ (action sur le bouton poussoir m, a relâché). La production de cet événement permet d'obtenir le marquage M_1 correspondant à la place p_2 marquée (L). Le passage du marquage M_1 au marquage M_0 résulte de

l'événement $\bar{m}a + ma$ soit a . Le réseau de PETRI conforme obtenu est représenté par la figure 3. Ce graphe résume le cahier des charges. L'ensemble M des marquages $[M] = \{M_0, M_1\}$ est caractérisé par le tableau matriciel fig. 4.

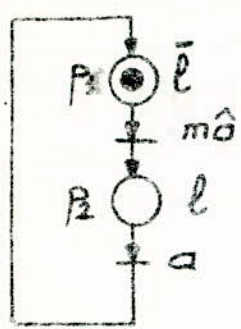


Figure 3

M_0	M_1	
1	0	P_1
0	1	P_2

Figure 4

L'association de deux états q_0, q_1 aux marquages M_0, M_1 permet la représentation de la matrice des états (figure 5) et de la matrice de sortie figure 6.

La matérialisation de la structure de commande à l'aide d'opérateurs NAND se déduit du développement par rapport au 1 du diagramme de Karnaugh (figure 7) et conduit au schéma technologique de la figure 8.

	$\bar{m}\bar{a}$	$\bar{m}a$	$m\bar{a}$	ma
q_0	1	1	1	1
q_1	1	1	1	1

Figure 5

q_0	\bar{l}
q_1	l

Figure 6

l	$\bar{m}a$	00	01	11	10
0	0	0	0	0	1
1	1	1	0	0	1

Figure 7

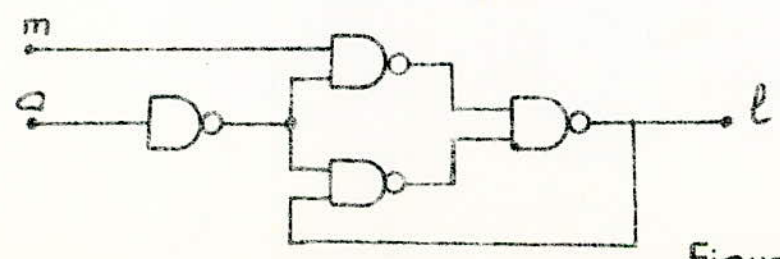


Figure 8

T.P. n° 2Mémoire R.S. à variation de niveau

Matérialiser en utilisant des opérateurs **NAND** un circuit de mémorisation à deux entrées **R** et **S** et une sortie **Y** répondant au cahier des charges suivant :

Les deux entrées **R** et **S** ne varient jamais simultanément et ne prennent jamais simultanément la valeur 1.

Quelle que soit la valeur de l'entrée **R** maintenue constante, la variation de 1 à 0 de **S** entraîne une variation de 0 à 1 ou un maintien à 1 de la sortie **Y**.

Quelle que soit la valeur de l'entrée **S** maintenue constante, la variation de 1 à 0 de **R** entraîne une variation de 1 à 0 ou un maintien à 0 de la sortie **Y**.

Toutes les variations de 0 à 1 des grandeurs d'entrée **R** et **S** ne modifient pas la valeur de la sortie **Y**.

Solution

Le schéma fonctionnel est représenté par la figure 1.

- Alphabet d'entrée $\{RS, R\bar{S}, \bar{R}S, \bar{R}\bar{S}\}$. L'événement RS et les séquences de deux événements successifs $\{R\bar{S}, \bar{R}S\}$, $\{\bar{R}S, R\bar{S}\}$ ne se produisent jamais par définition du cahier des charges.

Alphabet de sortie $\{Y, \bar{Y}\}$.

- L'état initial $R=0, S=0, Y=0$ est représenté par la place $p_1(\bar{Y})$ marquée (figure 2).

- Une variation de 0 à 1 de la grandeur de sortie n'est pas obtenue à partir de la production d'un événement mais de la séquence de deux événements

correspondant à la variation de 1 à 0 de la grandeur d'entrée S . Cette séquence $S\bar{S}$ permettant le passage de 0 à 1 de la sortie Y est résumée par le marquage de la place p_2 (\bar{Y} , marquage M_1) obtenu à partir du marquage initial M_0 par la production de S , puis par le marquage de la place p_3 (Y , marquage M_2) obtenu à partir du marquage M_1 par la production \bar{S} .

De manière identique à partir du marquage M_2 (Y) une variation de 1 à 0 de la grandeur de sortie n'est pas obtenue à partir de la production d'un seul événement mais de la séquence $R\bar{R}$ des deux événements correspondant à la variation de 1 à 0 de la grandeur d'entrée R . Le réseau de PETRI complété est représenté par la figure 2. L'ensemble des marquages $[M] = \{M_0, M_1, M_2, M_3\}$ est caractérisé par le tableau matriciel (figure 3).

Figure 1



	M_0	M_1	M_2	M_3	
p_1	1	0	0	0	P_1
p_2	0	1	0	0	P_2
p_3	0	0	1	0	P_3
p_4	0	0	0	1	P_4

Figure 3

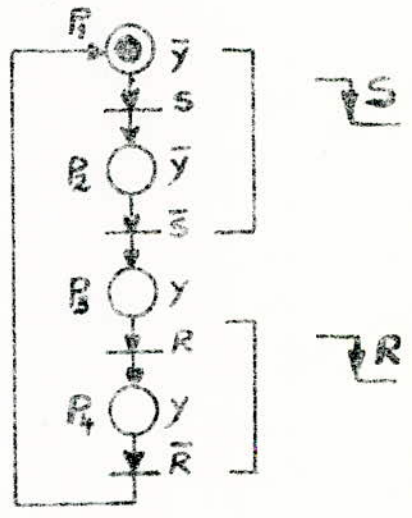


Figure 2

- L'association des quatre états $\{q_0, q_1, q_2, q_3\}$ aux marquages $\{M_0, M_1, M_2, M_3\}$ permet la représentation de la matrice des états et de la matrice de sortie (figure 4) deux variables secondaires V, W sont suffisantes pour coder les quatre états en respectant les adjacences.

La variable W est identique à la sortie Y . La matrice codée et la matrice de sortie sont données par la figure 5.

	$\bar{A}S$	$\bar{R}S$	RS	$R\bar{S}$	
q_0	q_0	q_1	-	q_0	\bar{Y}
q_1	q_2	q_1	-	-	\bar{Y}
q_2	q_2	q_2	-	q_3	Y
q_3	q_0	-	-	q_3	Y

Figure 4

wv	RS				y
	00	01	11	10	
00	00	01	-	00	0
01	11	01	-	-	0
11	11	11	-	10	1
10	00	-	-	10	1

Figure 5

wv	RS			
	00	01	11	10
00	00	01	-	00
01	11	01	-	10
11	11	11	-	10
10	00	01	-	10

Figure 6

En tenant compte des cases indifférentes (figure 6) on obtient:

$$V(t+Z) = S + V\bar{R}$$

$$W(t+Z) = \bar{S}V\bar{W} + W(\bar{V}R) = \bar{S}V + VW + WR$$

$$Y = W$$

Le schéma technologique correspondant à ces deux relations est représenté par la figure 7.

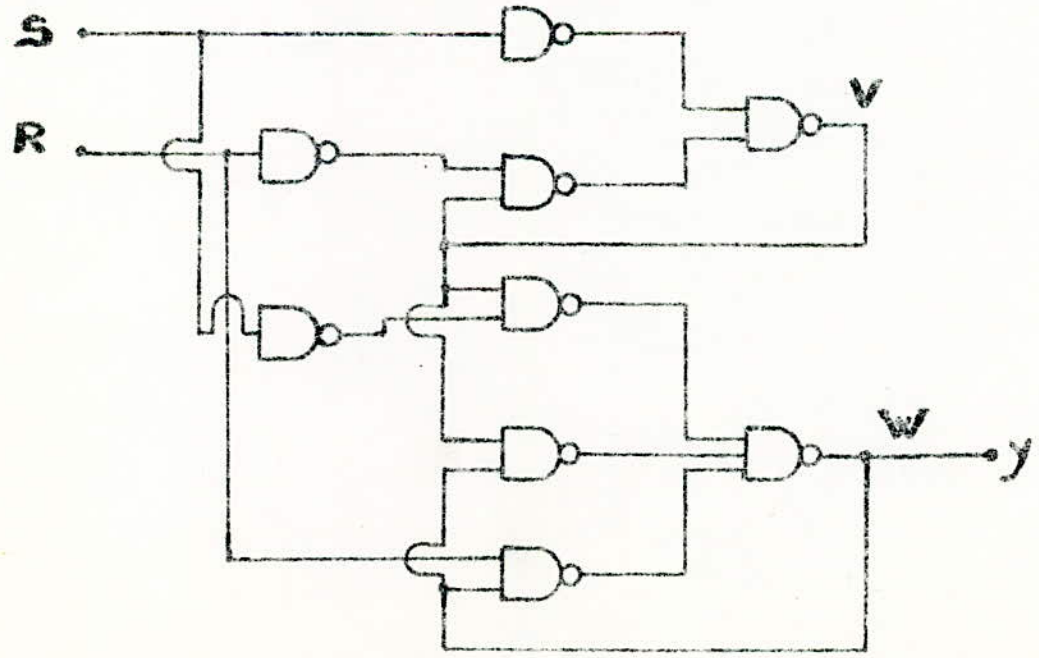


Figure 7.

Notre projet a été la synthèse de cette

Bibliographie

André C., Boeri F., Marin J.

"Synthèse et réalisation des systèmes logiques à évolutions simultanées".
Revue Française d'Automatique, Informatique Opérationnelle
vol. 10, n° 4, Avril 1976, p. 67 à 86.

"Conception de systèmes séquentiels totalement autotestables
partir des réseaux de PETRI. (RAIRO), vol 10, n° 11,
novembre 1976, p. 5 à 22.

Blanchard M., Daclin E.

"Synthèse des systèmes logiques" CEPADUES. Toulouse 1976.

Colloque. AFCET/ADEPA

"Automatismes logiques : recherche et applications industrielles".
Paris, décembre 1976.

Courvoisier M.

"Etude des systèmes logiques de commande asynchrones à évolutions simultanées". Thèse de Doctorat ès Sciences, Toulouse, Février 1975.

"Description et réalisation des systèmes de commandes asynchrones à évolutions simultanées". RAIRO, J₁, Février 1975, p. 5 à 35.

Foussard Y.

"Automates décrits par réseau de PETRI réalisés avec un seul type de module". Automatismes. Tome XXI, n° 4.2, Janvier-Février 1975.

Girard P., Naslin P.

"Construction des machines séquentielles industrielles" Dunod, 1973.

Misunas D.

"PETRI nets and speed independent design". Communication of the ACM, vol. 16, n°8, August 1973.

Thelliez S.

"Pratique séquentielle et réseaux de PETRI". Eyrolles 1978.

Tourres L.

"Une méthode nouvelle d'étude des systèmes logiques et son application à la réalisation d'automatismes programmés". Revue générale de l'électricité. Tome 85 n°3, mars 1976.

Valette R., Courvoisier M.

"Recherche d'un modèle adapté aux systèmes de commande de processus à évolutions parallèles". RAIRO, vol. 11, n°1, 1977.
