

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/44008529>

Logique séquentielle, GRAFCET et automatisme

Article

Source: OAI

CITATIONS

0

READS

1,395

1 author:



Jean-Paul Chemla

University of Tours

18 PUBLICATIONS 86 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Microemboli Detection [View project](#)



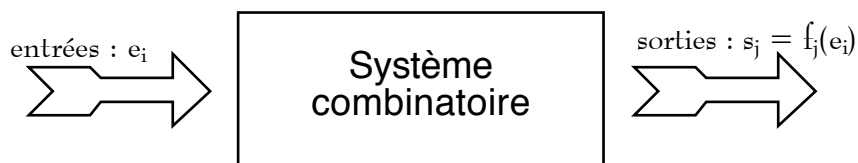
Home IO [View project](#)

Partie A : Introduction à la logique séquentielle

Chapitre I : Rappels sur les systèmes combinatoires

1.1 DÉFINITION

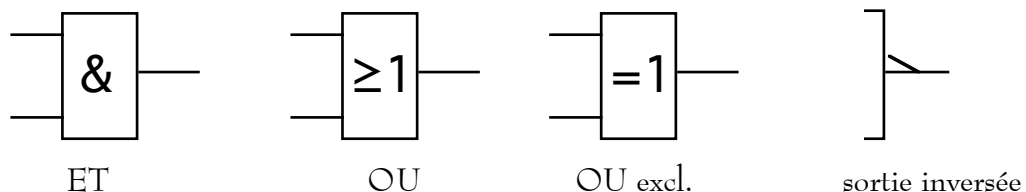
Dans un système logique (les entrées et sorties ne peuvent prendre que 0 ou 1 comme valeur) combinatoire, les sorties ne sont fonctions que des entrées.



L'outil mathématique qui permet de décrire les systèmes combinatoires est l'**algèbre de Bool**. Par la combinaison des trois fonctions de base que sont le *NON*, le *OU* (inclusif) et le *ET*, on va pouvoir décrire chacune des sorties en fonction des entrées.

1.2 Représentation d'une fonction booléenne par schémas à relais

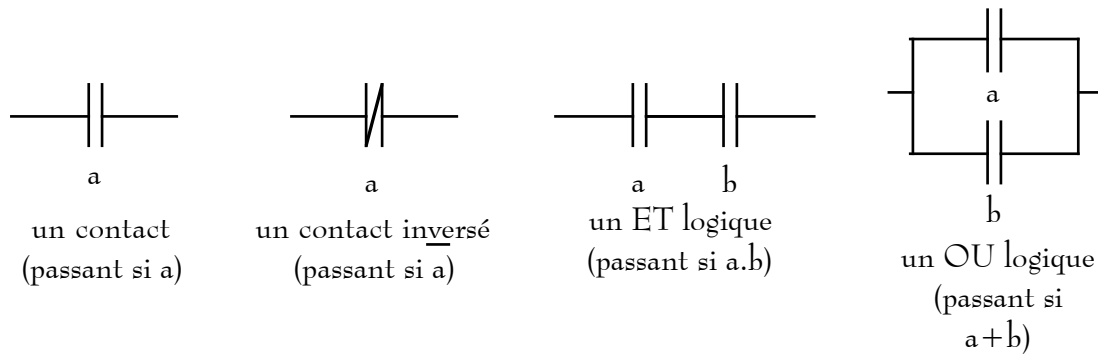
Le lecteur est habitué à représenter des fonctions booléennes par les symboles traditionnels tels que :



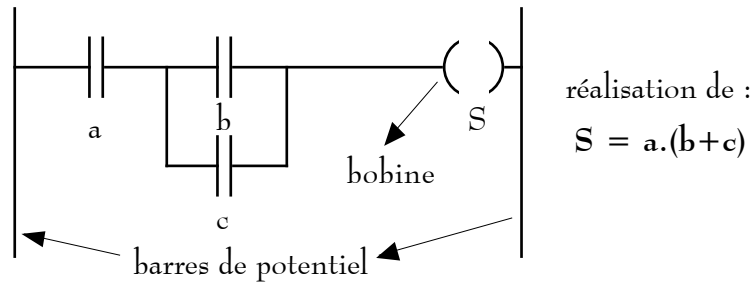
Il existe une autre façon de représenter les fonctions booléennes : les schémas à relais aussi appelé LADDER (vient des USA).

Les éléments de cette représentation sont :

- deux barres de potentiels (une à gauche, une à droite) ;
- des contacts (inversés ou non) portant le nom d'une variable d'entrée ;
- sur la dernière colonne à droite avant la barre de potentiel de droite, des bobines (inversées ou non) portant le nom d'une variable de sortie ;
- la mise en série (resp. en parallèle) de deux contacts représente un ET (resp. un OU).



Exemple de réalisation :

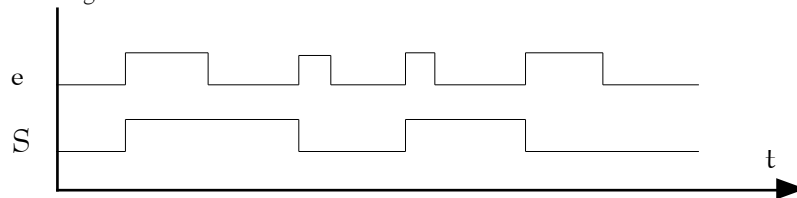


Ce mode de représentation est courant dans les langages d'automate (voir partie B). Cette représentation est plus naturelle pour les électriciens qui, pour comprendre le fonctionnement, mettent mentalement des interrupteurs à la place des contacts et une lampe à la place de la bobine. Si la lampe s'allume, c'est que la variable de sortie vaut 1 et 0 sinon.

Chapitre 2 : Notion de systèmes séquentiels

2.1. NOTION D'ÉTAT

Prenons l'exemple suivant : on considère un système à 1 entrée e et une sortie S. La sortie S du système doit changer de valeur à chaque front montant de l'entrée e. Ce cahier des charges peut être représenté par le chronogramme suivant :



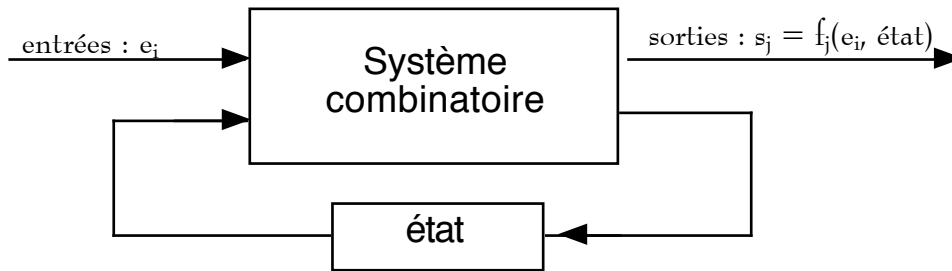
Pour une même valeur de e, S peut prendre deux valeurs 0 ou 1. Ce système n'est pas combinatoire: on ne peut pas définir $S = f(e)$

Par contre la valeur de S peut être déterminée en utilisant ce qui s'est passé auparavant. Le système a en mémoire la valeur de S avant changement. La réalisation de ce système nécessiterait des bascules.

Un **système séquentiel** est un système dont les sorties à l'instant t dépendent à la fois des entrées à cet instant, mais aussi de ce qui s'est passé auparavant : l'histoire du système. Cette histoire sera représentée par une succession d'**états** que prend le système au cours du temps. Le changement d'état sera provoqué par une variation des entrées. Les sorties sont fonction de l'état du système.

Remarque : Quand le nouvel état pourra être déterminé uniquement à partir de l'état immédiatement précédent et des entrées, le système sera dit markovien (on s'intéressera uniquement à ce type de système).

Un système séquentiel pourra être représenté par le schéma suivant:



Exemples de systèmes séquentiels : les montres, les digicodes, les ascenseurs.

Chapitre 3 : Modélisation des systèmes séquentiels

Le cahier des charges est constitué d'une suite de phrases décrivant le fonctionnement désiré du système. C'est la première étape de la conception d'un système. Afin d'analyser et de valider le cahier des charges, on le traduira en un formalisme qui ne permet aucune erreur d'interprétation. On parlera de modélisation. Les modèles obtenus pourront être utilisés aussi pour la synthèse (élaboration matérielle de la commande) :

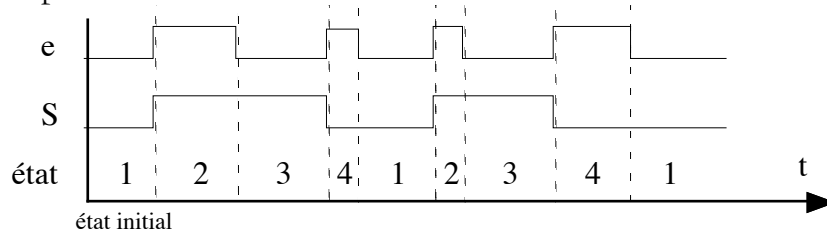
- chronogramme (diagramme des temps)
- graphe de fluence
- tableaux d'état
- graphe d'état
- graphe d'événement
- GRAFCET
- Réseaux de Petri

Dans ce cours, nous nous intéresserons plus particulièrement aux Grafcet (Partie B) qui permettent de représenter le fonctionnement de la partie commande de systèmes automatisés de production et aux Réseaux de Petri qui permettent une modélisation d'un système de production pour en analyser ses performances (En deuxième année).

3.1. CHRONOGRAMME

C'est un modèle graphique qui représente l'évolution au cours du temps de toutes les entrées et sorties du système.

exemple du diviseur par deux :



Cette représentation permet de définir un certain nombre d'états du système. Ils correspondent à une configuration des entrées sorties. Dès que l'on augmente le nombre d'entrées sorties, il existe un risque d'oublier certains états et certaines possibilités d'évolution. Ce mode de représentation n'est pas synthétique. L'état initial est choisi arbitrairement. Le chronogramme servira plutôt pour représenter un exemple concret de fonctionnement.

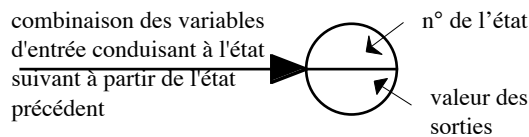
3.2 GRAPHE DE FLUENCE

C'est une traduction graphique du cahier des charges.

définitions préliminaires: **état stable:** état pour lequel les sorties du système restent inchangées, les combinaisons des entrées étant fixes.

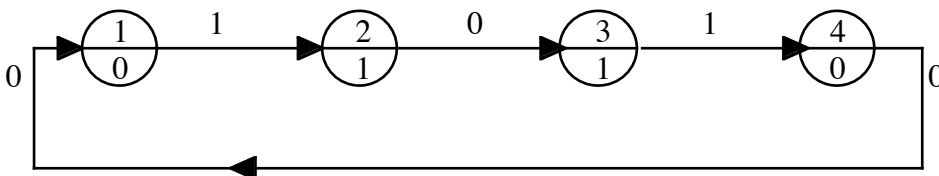
Le graphe de fluence représente tous les états stables du système et l'ordre chronologique dans lequel on atteint chacun des états à partir des autres en fonction des variations des variables d'entrée.

Un état est représenté graphiquement de la manière suivante:



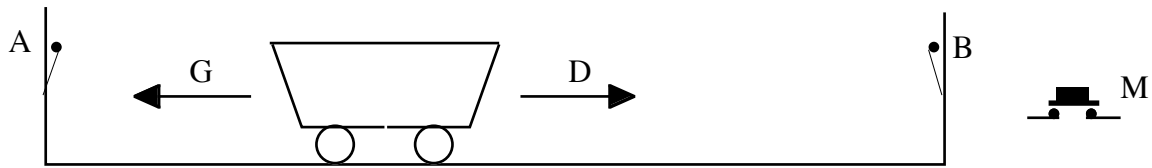
exemple du diviseur par 2 :

On choisit un état initial: c'est l'état à partir duquel on construit le graphe.

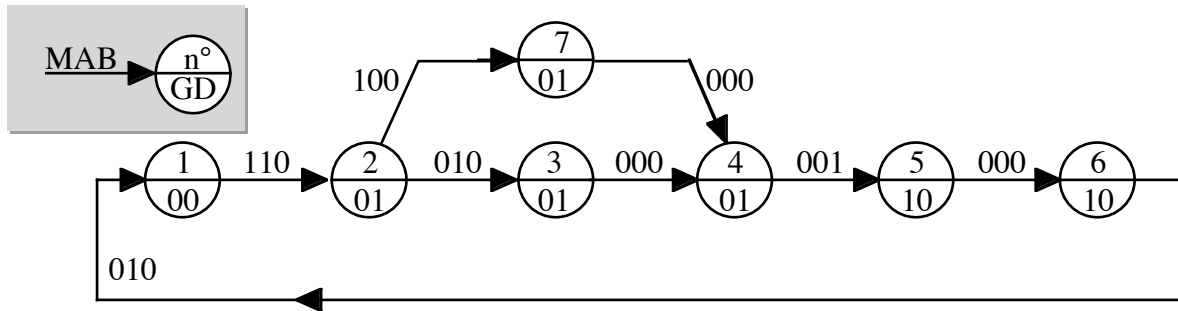


exemple du chariot :

On considère le procédé suivant:



cahier des charges: Si l'on appuie sur le bouton poussoir M lorsque le chariot est au repos en A. ce dernier quitte A, arrive en B et revient en A où il s'arrête.



Remarques:

Il s'agit bien d'un système séquentiel, les états 4 et 6 ont les mêmes entrées et des sorties différentes. Cette méthode de modélisation est systématique : pour chaque état on envisage toutes les variations possibles des entrées. Pour ne pas alourdir la représentation on s'interdit d'appuyer de nouveau sur M lorsque le chariot est parti.

On peut faire du graphe de fluence une représentation tabulaire : le tableau d'état primitif

3.3 TABLEAU D'ETAT

a) Tableau d'état primitif.

Les colonnes de ce tableau correspondent aux combinaisons des variables d'entrée du système. Les lignes correspondent aux différents états. Les valeurs des sorties sont associées à chaque état.

exemple du diviseur par 2 :

e	0	1	S
	1	2	0
	3	2	1
	3	4	1
	1	4	0

Les chiffres en gras correspondent aux états stables du système. Les autres correspondent aux états transitoires, c'est à dire au passage d'un état stable vers l'état stable suivant. Cette transition est provoquée par la variation de l'entrée. L'évolution se fait toujours horizontalement puis verticalement.

exemple du chariot :

On transpose le graphe de fluence en tableau d'état primitif :

M.A.B	0.0.0	0.0.1	0.1.1	0.1.0	1.1.0	1.1.1	1.0.1	1.0.0	G.D
	-	-	-	①	2	-	-	-	0.0
	-	-	-	3	②	-	-	7	0.1
	4	-	-	③	-	-	-	-	0.1
	④	5	-	-	-	-	-	-	0.1
	6	⑤	-	-	-	-	-	-	1.0
	⑥	-	-	1	-	-	-	-	1.0
	4	-	-	-	-	-	-	⑦	0.1

Les traits correspondent aux impossibilités d'évolution du système à partir de l'état stable indiqué sur la ligne.

b) États stables équivalents ou pseudo-équivalents

Il est possible que, au cours de la description du système permettant d'aboutir au tableau d'états primitif, on ait utilisé un ou plusieurs états pour représenter en réalité un seul état stable. On dira alors que deux (ou plusieurs) états stables sont:

équivalents si : ils correspondent aux mêmes entrées,
ils produisent les mêmes sorties,
les séquences qui en sont issues sont identiques.

extrait d'un tableau d'état tels que les états 1 et 5 sont équivalents :

e1.e2	0.0	0.1	1.1	1.0	S
	①	2	-	4	0
	⑤	2	-	4	0

les états 1 et 5 sont équivalents : tous les 5 peuvent être remplacés par des 1.

pseudo-équivalents si : mêmes entrées,
mêmes sorties,
les séquences qui en sont issues existent dans un cas et n'existent pas dans l'autre.

extrait d'un tableau d'état tels que les états 1 et 5 sont pseudo-équivalents :

e1.e2	0.0	0.1	1.1	1.0	S
	①	-	-	4	0
	⑤	2	-	4	0

Dans les deux cas ci dessus on obtiendra:

e1.e2	0.0	0.1	1.1	1.0	S
	①	2	-	4	0

c) États compatibles : obtention du tableau d'état réduit

Les tableaux d'états obtenus jusqu'à maintenant ne comportent qu'un seul état stable par ligne. Certains états pourront être distingués en utilisant les combinaisons des variables d'entrée.

exemple :

e1.e2	0.0	0.1	1.1	1.0	S
	①	2	-	4	0
	1	②	3	-	1

Les états 1 et 2 sont compatibles: on fusionne les deux lignes:

e1.e2	0.0	0.1	1.1	1.0
	1(0)	2(1)	3	4

Les états 1 et 2 seront distingués par la combinaison des variables d'entrée (0.0 pour l'état 1, 0.1 pour l'état 2). Les évolutions du système seront préservées. Chaque valeur de la sortie est indiquée entre parenthèses.

Après fusionnement on obtient le *tableau d'états réduit*. Chaque ligne du tableau d'états réduit sera codée par des variables supplémentaires, les *variables internes*. Les lignes du tableau d'état réduit représentent les *états internes* du système.

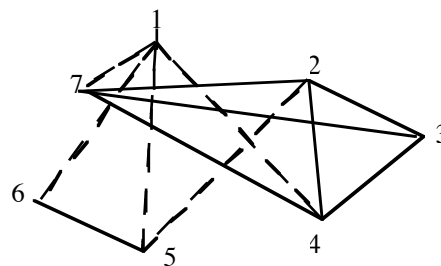
Règles de fusionnement : deux lignes peuvent être fusionnées si dans la même colonne on trouve:

- un état stable i et un état transitoire i,
- des indifférences (-),
- un état (stable ou instable) et une indifférence.

Pour rechercher les états compatibles on établit le polygone des liaisons dans lequel chaque sommet représente une ligne. Lorsque deux lignes sont fusionnables on les relie par un trait plein si les sorties sont identiques, on les relie par un trait pointillé si les sorties sont différentes.

exemple du chariot :

M.A.B	0.0.0	0.0.1	0.1.1	0.1.0	1.1.0	1.1.1	1.0.1	1.0.0	G.D
	-	-	-	①	2	-	-	-	0.0
	-	-	-	3	②	-	-	7	0.1
	4	-	-	③	-	-	-	-	0.1
④	5	-	-	-	-	-	-	-	0.1
6	⑤	-	-	-	-	-	-	-	1.0
⑥	-	-	-	1	-	-	-	-	1.0
	4	-	-	-	-	-	-	⑦	0.1



Polygone des liaisons

Deux choix sont possibles pour le fusionnement:

- fusionner des lignes pour lesquelles les variables de sortie sont identiques : on obtient alors une machine de MOORE.
- fusionner des lignes pour lesquelles les variables de sortie sont différentes: on obtient alors une machine de MEALY.

Pour l'exemple précédent :

- machine de MOORE:

M.A.B	0.0.0	0.0.1	0.1.1	0.1.0	1.1.0	1.1.1	1.0.1	1.0.0	G.D
a	-	-	-	①	2	-	-	-	0.0
b	④	5	-	③	②	-	-	⑦	0.1
c	⑥	⑤	-	1	-	-	-	-	1.0

- machine de MEALY:

M.A.B	0.0.0	0.0.1	0.1.1	0.1.0	1.1.0	1.1.1	1.0.1	1.0.0
a c	6 (1.0)	5 (1.0)	-	1 (0.0)	2	-	-	-
b	4 (0.1)	5	-	3 (0.1)	2 (0.1)	-	-	7 (0.1)

Le nombre de lignes du tableau d'états réduit représente le nombre d'états internes nécessaire pour mémoriser le passé du système (3 pour la machine de Moore, 2 pour la machine de Mealy).

Dans la machine de Moore, les états 2 et 3 seront représentés par le même état interne b et par des entrées différentes. Les états 1 et 3 correspondant aux mêmes entrées externes seront représentés par des états internes différents.

Les états internes seront codés par des variables internes. Ce codage pourra être optimal si l'on utilise le nombre minimum possible de variables internes (2 pour la machine de Moore, 1 pour la machine de Mealy). On peut aussi utiliser une variable par état interne.

3.4. REMARQUES

La description d'un système séquentiel par graphe de fluence ou par tableau d'états primitif est une méthode systématique de modélisation : on examine toutes les combinaisons des variables d'entrée. Ces méthodes ne seront pas applicables pour des systèmes ayant un grand nombre d'entrées et (ou) un grand nombre d'états. On préférera alors employer des méthodes basées implicitement sur la description directe des états internes :

- graphe d'état,
- graphe d'événements,
- GRAFCET,
- Réseaux de Petri.

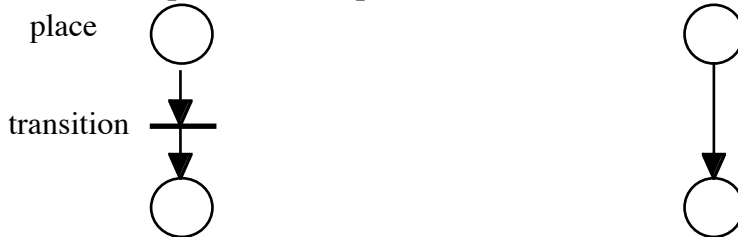
Ces méthodes n'étant plus systématiques, nécessiteront une validation des modèles obtenus.

Nous présenterons le GRAFCET qui sert à la modélisation du fonctionnement de la partie commande des systèmes automatisés de production dans la partie B ; les réseaux de Petri seront vus l'an prochain.

3.5. GRAPHE D'ETAT

a) Le modèle

Les états du système sont représentés par des cercles (que l'on appellera place). Des conditions d'évolution qui sont des combinaisons des variables d'entrées externes permettent l'évolution du système (passage d'un état à un autre état). Elles sont indiquées à coté des transitions représentées par des traits. Un arc orienté relie une place à une transition et une transition à une place. Une transition est franchie quand l'état qui la sensibilise est actif et quand la condition d'évolution qui lui est associée est vraie. L'état initial est indiqué par un astérisque ou un point. Les sorties (ou action) associées aux états sont indiqués à coté des places.

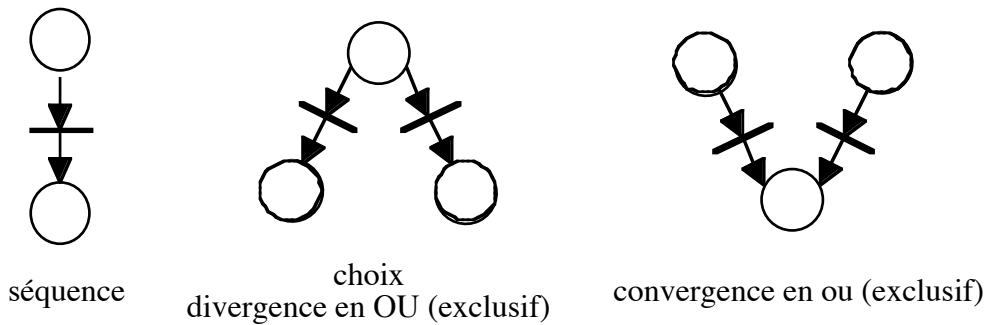


remarque: il existe une autre représentation où l'on ne fait pas figurer les transitions.

N.B.: un système étant dans un seul état à un instant donné, dans un graphe d'état une seule place est active à un instant donné. En conséquence :

- dans un graphe d'état chaque transition a exactement un arc entrant et un arc sortant,
- les conditions d'évolution permettant de quitter un état doivent être exclusives.

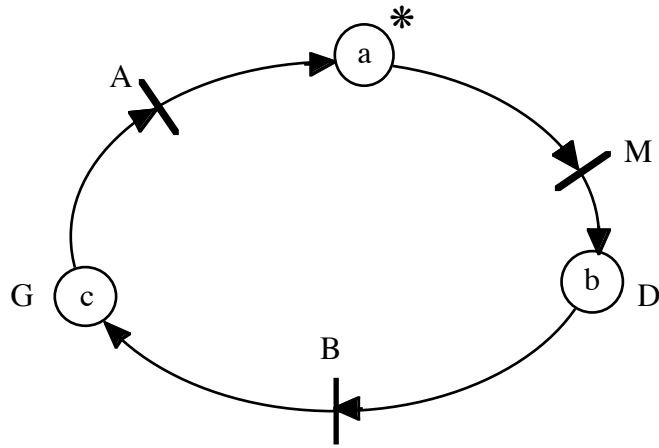
On pourra donc avoir les structures suivantes:



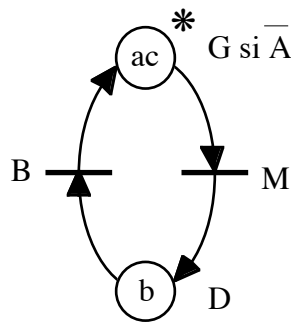
premier exemple : chariot

Démarche:

- choix d'un état initial (*)
- recherche des états successeurs et des conditions d'évolution qui amènent dans ces états
- on associe ensuite les actions aux états.



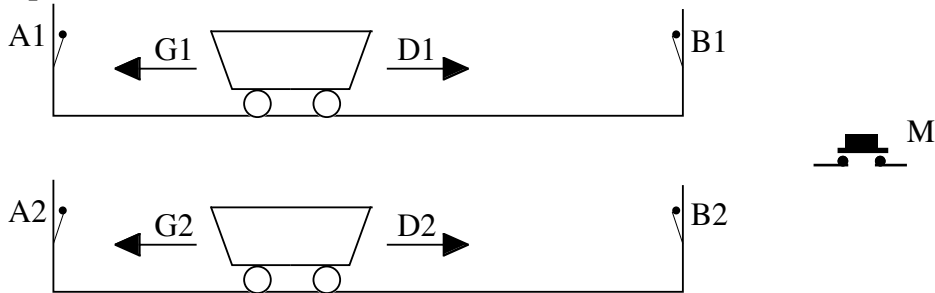
On aurait pu avoir aussi:



Le premier modèle est une machine de Moore, le second une machine de Mealy. On remarque que dans cette dernière les sorties dépendent des états et des entrées externes.

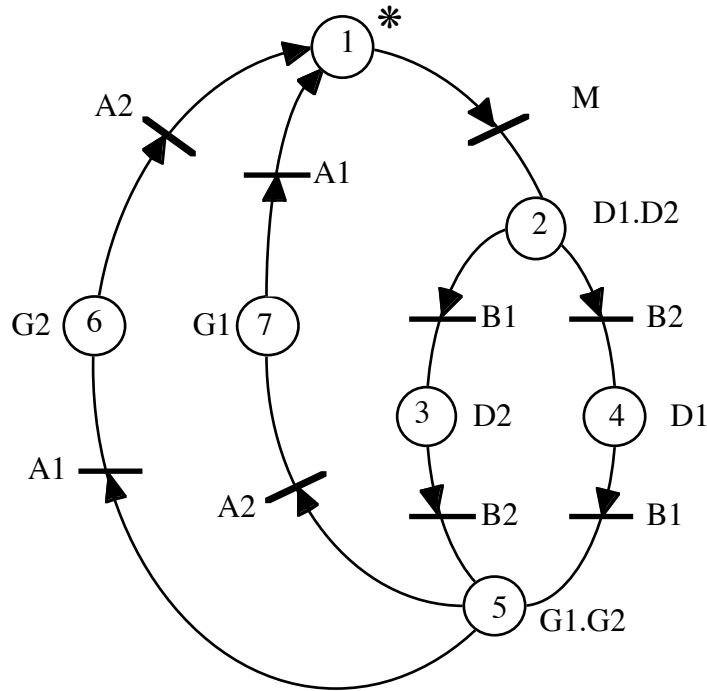
Ces modèles peuvent s'obtenir de façon systématique à partir des tableaux d'états réduits (je vous invite à le faire).

deuxième exemple:



Par action sur M, si le chariot 1 est en A1, si le chariot 2 est en A2, déplacer le chariot 1 vers B1 et le chariot 2 vers B2. Lorsque le chariot 1 arrive en B1 il retourne vers A1 si le chariot 2 est déjà passé en B2. Lorsque le chariot 2 arrive en B2 il retourne vers A2 si le chariot 1 est déjà passé en B1.

Il est très difficile de faire un graphe de fluence ou un tableau d'états primitif : il y a cinq variables d'entrées externes donc 32 combinaisons différentes. On construira directement le graphe d'état à partir de l'état initial que l'on choisit.



Pour construire ce modèle on suppose que les deux événements $\downarrow B1$ et $\downarrow B2$ ne peuvent pas se produire en même temps. Un événement est produit par le changement de niveau d'une variable ou d'une expression booléenne.

Hypothèse fondamentale des systèmes séquentiels : Des événements non corrélés ne se produisent jamais en même temps.

Attention aux "OU" : Cette hypothèse devrait garantir que le système évoluera soit vers l'état 3 soit vers l'état 4 depuis l'état 2. Il faudra être très prudent pour passer du modèle à la réalisation : les choix technologiques peuvent corréler deux événements qui ne le sont pas. Il est préférable d'écrire la condition d'évolution de: 2 vers 3 : $B1.B2'$ et de 2 vers 4 : $B2.B1'$. On a assuré l'exclusion logique des conditions. (Par commodité, je noterai $X' = \text{non } X$)

b) Validation du modèle

La modélisation par graphe d'état n'étant pas une démarche où l'on envisage toutes les combinaisons possibles des variables d'entrée pour décrire le système il est nécessaire de valider le modèle obtenu. On vérifiera que : les conditions d'évolution des divergences en OU sont physiquement ou logiquement exclusives et que le modèle traduit bien le cahier des charges.

c) Conclusion

Un graphe d'état représente tous les états du système (chaque place est un état). Les graphes d'états font clairement apparaître deux mécanismes élémentaires des systèmes séquentiels : la séquence et le choix. Par contre ils ne font pas apparaître clairement le parallélisme et la synchronisation. Ces derniers apparaissent clairement dans un autre type de modèle: les graphes d'événements.

Exemple (traité en cours) :

On considère un digicode à l'entrée d'un immeuble. Il possède les touches a, b, c, d, e. Le code est "a.c.e". Ces touches sont considérées comme des entrées. Elles valent 1 quand la touche est appuyée et 0 quand elle est relâchée.

Faire un graphe d'état représentant le fonctionnement de ce digicode.

Faire ensuite le tableau d'état réduit du système en considérant qu'on n'appuie jamais sur deux touches en même temps.

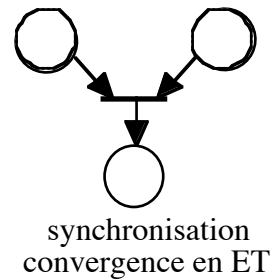
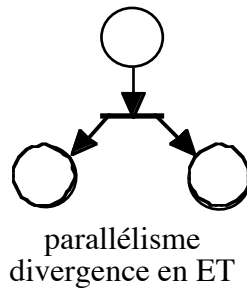
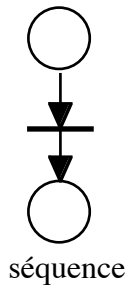
Vérifiez que le graphe d'état et le tableau d'état réduit sont équivalents.

Exercice :

Un chariot fait des aller-retours (sortie D et G) entre deux points (entrées a et b) sans arrêt. Quand un interrupteur c est à 1, le chariot s'arrête. Il reprend sa course quand c repasse à 0. En prenant comme état initial "le chariot se déplace vers la droite", élaborez le graphe d'état qui représentera ce système. (suite de l'exercice à la fin du chapitre 4)

3.6. GRAPHE D'EVENEMENTS

Dans un graphe d'événements chaque place a au plus une transition d'entrée et une transition de sortie. Par conséquent on peut avoir les structures suivantes:

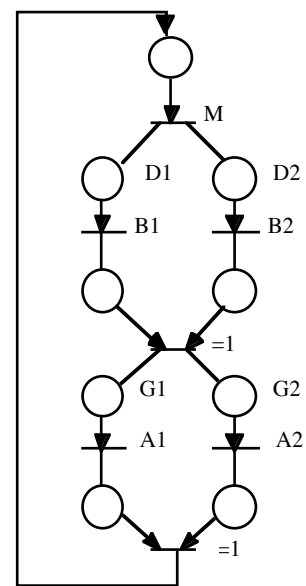


Deux nouveaux mécanismes élémentaires des systèmes séquentiels apparaissent: le parallélisme et la synchronisation. On ne peut plus, en graphe d'événements, représenter les choix.

exemple des deux chariots :

Il apparaît une nouvelle représentation des états du système. Un état est représenté par l'ensemble des places actives à un instant donné. Remarque : Les graphes d'états et les graphes d'événements n'étant pas normalisés, nous avons choisi une représentation graphique proche de celle des réseaux de Petri.

" =1 " marque des conditions de transition toujours vraie: elles seront franchies dès que toutes les places immédiatement précédentes seront actives.



Pour $b+$:

équation d'évolution : on atteint $b+$ à partir de a seulement. En exploitant la ligne a du tableau d'état réduit :

$B \setminus M.A$	0.0	0.1	1.1	1.0
0	x	0	1	x
1	x	x	x	x

$$\text{d'où } b+_{ev} = M.a$$

équation de maintien : on exploite la ligne b du tableau :

$B \setminus M.A$	0.0	0.1	1.1	1.0
0	1	1	1	1
1	0	x	x	x

$$\text{d'où } b+_{maint} = B'.b$$

On obtient finalement : $b+ = M.a + B'.b$

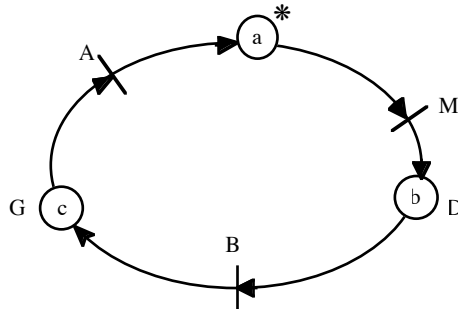
Pour tout le système on a :

$$a+ = A.c + M'.a$$

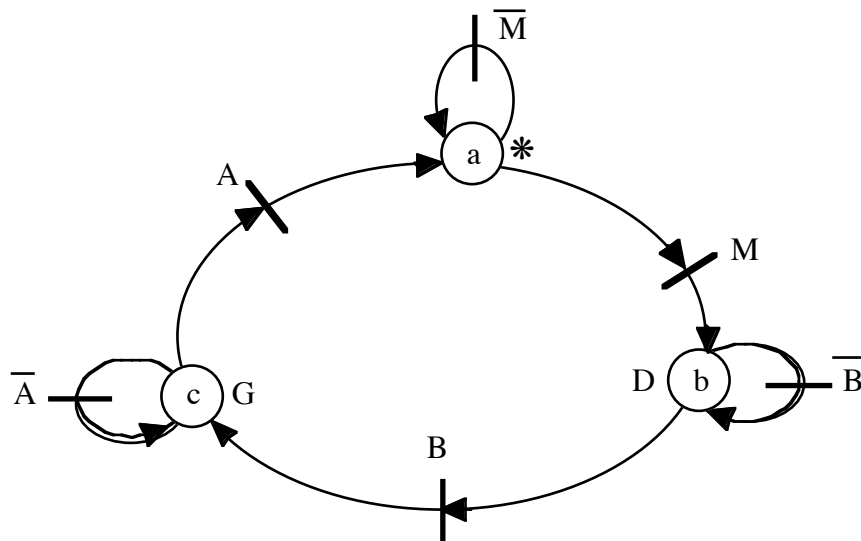
$$b+ = M.a + B'.b$$

$$c+ = B.b + A'.c$$

Reprenons le graphe d'état obtenu au chapitre précédent :



Sur le modèle figurent uniquement les conditions d'évolution. On complète ce graphe pour représenter aussi les conditions de maintien :



explications: le système étant dans l'état a, si la condition M devient vraie il passe dans l'état b. La condition M peut alors disparaître, c'est B' qui maintient le système dans l'état b.

b) Équations des sorties

Pour une machine de Moore les sorties s'écrivent uniquement en fonction des états. Pour cet exemple on obtient :

$$D = b \qquad G = c$$

Pour une machine de Mealy les sorties s'écrivent en fonction des états et des entrées externes. On aurait :

$$G = (ac).A \qquad D = b$$

c) Initialisation

On peut prendre en compte l'initialisation dans les équations du système. On ajoute une entrée externe supplémentaire I:

$$a+ = A.c + M'.a + I \qquad b+ = (M.a + B'.b).I' \qquad c+ = (B.b + A'.c).I'$$

En faisant $I = 1$ on met le système dans son état initial. On autorise le fonctionnement en faisant $I = 0$

Remarque :

- La représentation algébrique s'étend à d'autres modèles que le graphe d'état, au graphe d'événements et au Grafcet,
- La réalisation pratique consiste en une traduction de ces équations algébriques en circuits logiques en utilisant des bascules D ou JK, synchrones ou asynchrones.

Compléter le graphe d'état du chariot (voir exercice précédent) en

- *prenant en compte que les O/N doivent être exclusifs pour éviter tout dysfonctionnement,*
- *représentant les conditions de maintien,*

puis écrire les équations d'évolution et de maintien des états du système. Ecrire enfin les équations des sorties.

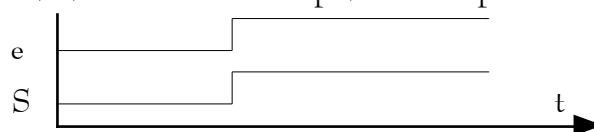
4.2 DIFFÉRENTES CATÉGORIES DE SYSTEMES SÉQUENTIEL

Nous nous contenterons de les énumérer et d'envisager les problèmes qui se posent.

La classification en asynchrone et synchrone dépend uniquement de la réalisation envisagée.

a) Systèmes Séquentiels Asynchrones

Une modification des entrées est immédiatement prise en compte.



Les réalisations asynchrones sont de moins en moins envisagées (sauf quand on a des objectifs de rapidité) car elles conduisent à des aléas de fonctionnement. Pour éviter ces aléas, il faudra s'interdire toute variation simultanée des entrées qui agissent sur le bloc combinatoire.

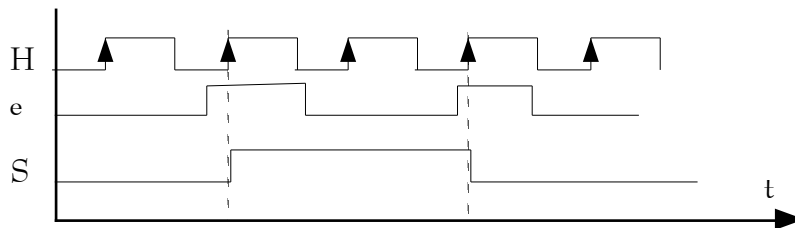
Les problèmes d'aléas seront facilement maîtrisés en effectuant une réalisation synchrone.

b) Systèmes Séquentiels Synchrones

Les changements d'état sont effectués après validation par le front montant (ou descendant) d'un signal supplémentaire appelé horloge. Pratiquement, l'horloge n'intervient que sur la mémorisation des états: elle n'est pas considérée comme une entrée supplémentaire et n'interviendra jamais dans les méthodes de synthèse.

avantages: les entrées externes peuvent commuter n'importe quand sauf sur les fronts actifs de l'horloge. Elles peuvent éventuellement être synchronisées par une deuxième horloge afin d'obtenir des informations stables sur les entrées des organes de mémorisation.

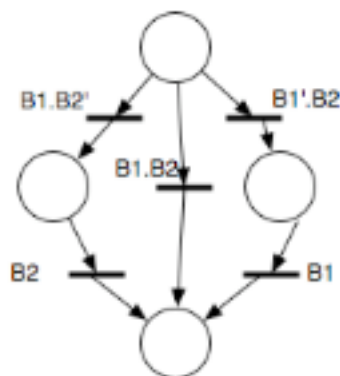
Pour le système précédent on obtiendra:



La plupart des réalisations concrètes utilisant des processeurs (automates, cartes à micro-contrôleurs...) va fonctionner sur ce mode.

Pour ces types de procédés, l'hypothèse "Des événements non corrélés ne se produisent jamais en même temps" n'est plus assurée car dans le "en même temps" il faut cette fois comprendre "entre deux fronts d'horloge". Il va donc falloir bien gérer cette situation.

Prenons l'exemple des deux chariots. Le graphe d'état a été construit en prenant l'hypothèse qu'il est impossible que les deux chariots arrivent en même temps. D'où une divergence B1 OU B2. Rendre le OU exclusif ne résoud pas le problème d'une éventuelle "simultanéité" des deux événements (B1 et B2 dans un même créneau d'horloge). Il faut donc prévoir une branche OU supplémentaire avec B1.B2. Ci-dessous la modification pour la divergence OU en B1 et B2. Il faudrait faire de même avec la divergence A1 OU A2



Fin de la partie A

Partie B : GRAFCET et automatisme

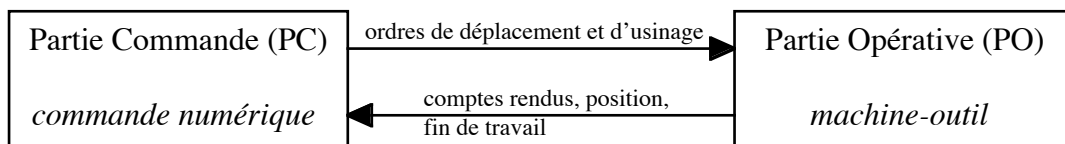
Chapitre I : Description d'un automatisme

1.1. DÉCOMPOSITION PO / PC

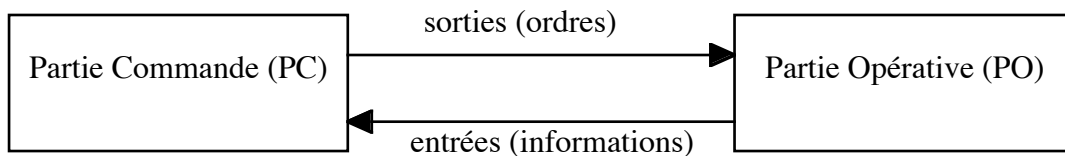
Un système automatisé se décompose en deux parties distinctes :

- LA PARTIE OPÉRATIVE (PO) : c'est la partie puissance, celle qui opère, celle qui agit (le muscle),
- LA PARTIE CONDUITE (PC) : c'est la partie qui commande la partie opérative, celle qui sait ce qu'il faut faire (le cerveau).

Exemple : une machine-outil à commande numérique



En général :



En automatique, on s'intéresse à la partie commande, son élaboration, sa description et sa validation. Dans les milieux industriels, ces parties commandes (de systèmes séquentiels) se concrétisent le plus souvent par un automate programmable et/ou une armoire de commande.

1.2. DISTINCTION ENTRÉE / SORTIE

On se place sur un point de vue 'Partie commande'.

Les **entrées** sont des informations booléennes qui peuvent provenir soit de **capteurs** (A ou B dans l'exemple précédent du chariot) soit de **boutons de commande** qu'un opérateur appuiera (M dans l'exemple précédent).

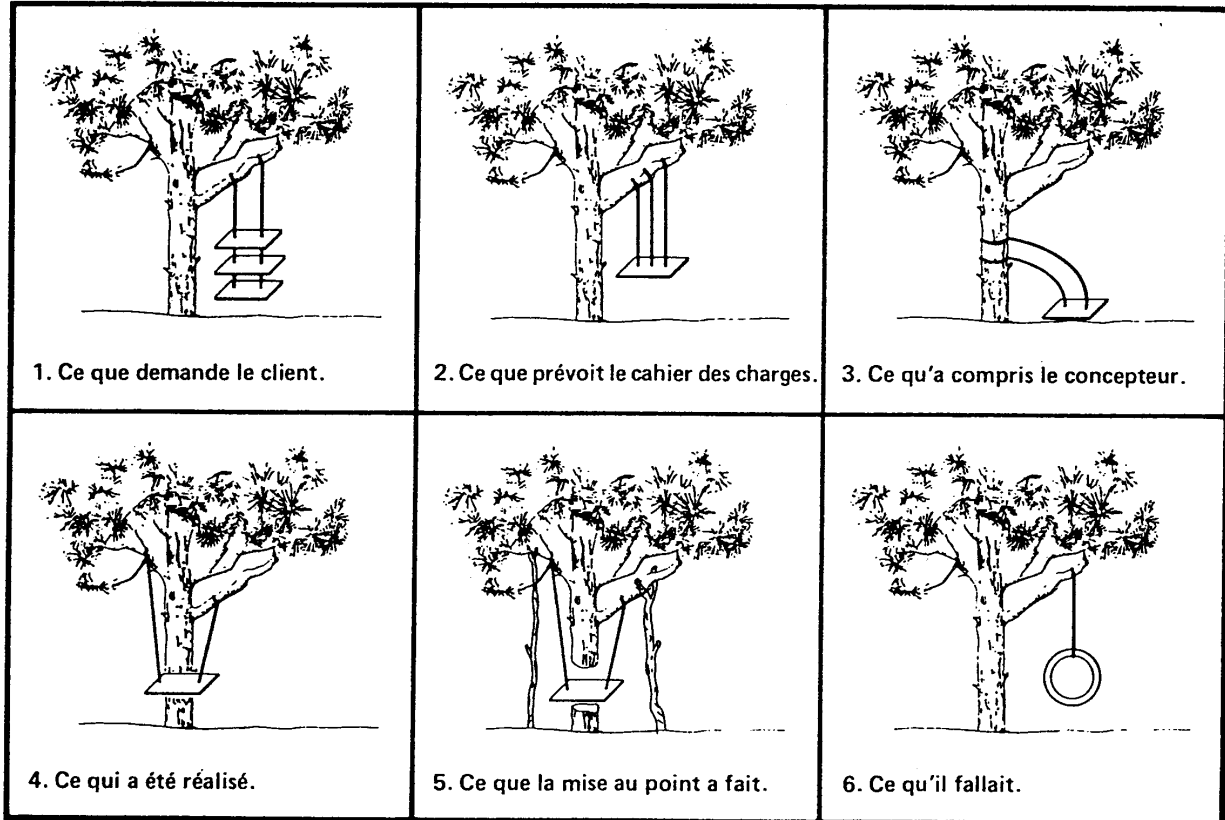
Les **sorties** sont soit les **ordres** donnés à la partie opérative soit des **voyants** du pupitre de commande. Ce sont aussi des variables booléennes.

Donner les entrées et les sorties de la partie commande d'un ascenseur.

Chapitre 2 : Présentation du Grafcet - la norme

2.1 UN PEU D'HISTOIRE

a) Pourquoi une norme pour la description d'automatismes ?



"Le Grafcet devrait non seulement permettre que les 3 premiers dessins soient identiques mais encore qu'ils représentent réellement ce dont a besoin l'utilisateur (dessin 6). Par la synthèse directe, on devrait également supprimer les déformations que la phase de réalisation fait subir au cahier des charges (dessins 4 et 5). Mais tout n'est pas si simple". (M. Blanchard, un des créateurs du Grafcet)

b) Histoire du GRAFCET

1975 : Un groupe de travail de l'AFCET (Association Française pour la Cybernétique Économique et Technique) décide de créer une commission "Normalisation de la représentation du cahier des charges d'un automatisme logique" sous l'impulsion de Michel Blanchard.

1977 : Le Grafcet est une norme française. Il y a 24 signataires. 12 d'entre eux sont universitaires, les 12 autres sont industriels.

1987 : Le Grafcet est une norme internationale.

Le Texte ci-dessous est de Michel Blanchard et date de 1982 :

GRAF CET. Sans doute le lecteur a-t-il eu l'occasion de découvrir récemment ce nom dans une revue spécialisée, la notice d'un industriel ou au détour d'un programme d'enseignement. Mais il est peu probable qu'il connaisse son histoire. Elle est pourtant exemplaire.

Au début des années 70, le bilan de la conception des systèmes logiques n'était pas brillant. Chez les chercheurs, on constatait encore une importante dissipation d'énergie pour perfectionner les méthodes théoriques de synthèse (Huffman, expressions régulières, ...) afin de les rendre plus accessibles, plus efficaces et « optimales ». Dans l'industrie, ces travaux étaient ignorés ou jugés avec quelque sévérité; ils avaient en tous cas peu d'impact sur les méthodes pratiques de synthèse qui restaient dans l'ensemble empiriques et rattachées à la technologie des automatismes à relais.

Cependant, avec l'arrivée des technologies nouvelles et l'accroissement de la complexité des systèmes étudiés, les praticiens prirent conscience que l'approche empirique n'était ni sûre, ni adaptée à leurs besoins. Certaines tentatives de concertation entre chercheurs et industriels pour associer tant bien que mal méthodes théoriques et besoins pratiques échouèrent. Il fallait imaginer une autre voie.

Le premier pas pour sortir de cette impasse fut accompli par P. GIRARD qui très tôt introduisit les notions de réceptivité et d'étape. A la même époque, aux Etats-Unis, des équipes d'informaticiens, totalement étrangers aux problèmes de synthèse des automatismes logiques, découvraient les réseaux de Petri que celui-ci avait définis dans sa thèse en .. 1962.

La voie nouvelle dans laquelle s'engageait la conception des systèmes logiques se précisait alors. Deux mots clefs la définissent: cahier des charges et modélisation.

Partout en France - car curieusement cette approche est typiquement française -, d'importants travaux de recherche, souvent soutenus par des organismes officiels (DGRST, ...), aboutissaient à des propositions de modèles aptes à représenter un cahier des charges. De leur côté les industriels s'orientaient, faute de mieux, vers l'utilisation d'outils dérivés des organigrammes employés en Informatique.

Devant cette trop grande diversité d'outils, pourtant issus d'une même démarche, le risque était grand de laisser passer une occasion inespérée d'aborder sous un angle neuf la synthèse des systèmes logiques.

C'est dans ces conditions que fut créée en 1975 une « commission de normalisation de la représentation du cahier des charges d'un automate logique », dans le cadre du groupe de travail « Systèmes logiques » de l'AF CET (Association Française pour la Cybernétique Économique et Technique), que nous animions à l'époque. L'objectif de la commission était clair: homogénéiser les différentes approches afin de dégager un outil unique de représentation d'un cahier des charges. Était-il réaliste ? Nous le pensions. Difficile à atteindre ? Personne n'en doutait.

Les vingt-quatre membres de la commission, également répartis entre Recherche et Industrie, s'attelaient alors à une tâche ardue puisque quatorze modes de représentations différents étaient sur la ligne de départ. Très vite, nous prîmes conscience qu'avant de rechercher un outil unique de représentation, il fallait s'entendre sur l'objet à représenter, en l'occurrence le contenu de cahier des charges. Rapidement aussi, ce petit groupe se transforma en une équipe soudée où chercheurs et industriels, théoriciens et praticiens, concepteurs et utilisateurs travaillaient ensemble avec une diversité d'approches et de conceptions qui fit précisément la richesse de ces travaux, avec aussi une foi et une persévérance qui seules permirent de surmonter les nombreuses difficultés de parcours.

En 1977, deux ans après sa création, la commission, dans son rapport final, définissait le contenu d'un cahier des charges et ses différents niveaux d'élaboration et proposait un outil pour sa représentation: le GRAFCET (Graphe de Commande Étape - Transition et aussi rappel du rôle important joué par l'AF CET dans cette création); puis elle se séparait. Loin d'un point final, ce rapport marque au contraire le point de départ de la fulgurante carrière du GRAFCET.

En effet, le groupe de travail « Automatisation séquentielle » de l'ADEPA (Agence nationale pour le Développement de la Production Automatisée) dont l'animateur avait participé aux travaux de la commission AF CET, reprit le flambeau pour donner au GRAFCET, sans toucher au fond, une forme normalisée tenant compte des normes existantes et des usages généraux des symboles

normalisés. Aujourd'hui, le rapport de ce groupe de travail constitue une contribution essentielle à un avant-projet de norme française. L'ADEPA, en le diffusant largement, assure la promotion du GRAFCET notamment auprès des Petites et Moyennes Industries.

Ainsi, quatre années se sont écoulées et la percée du GRAFCET est indiscutable. Elle va bien au-delà des espoirs les plus fous que nous nourrissions en créant la commission AFCET.

Pouvait-on imaginer en effet que le GRAFCET deviendrait une proposition française de Norme auprès des organismes internationaux compétents et parviendrait à s'imposer face à ses concurrents étrangers.

Pouvait-on imaginer que l'Enseignement Technique, réagissant avec une belle promptitude, inscrirait le GRAFCET aux programmes de certaines agrégations, le ferait pénétrer jusque dans les Lycées techniques si bien que les techniciens de demain ne tarderont pas à le rencontrer en feuilletant leurs manuels d'élève. Chacun mesure l'importance de l'événement.

Pouvait-on imaginer enfin que les industriels l'introduiraient sans réserve dans leurs bureaux d'étude, leurs notices allant même jusqu'à créer de nouveaux produits adaptés à son implantation et des moyens de C.A.O. pour mieux l'exploiter, que des universitaires se pencheraient sur lui pour en acquérir la connaissance profonde que seule une étude théorique peut apporter.

Le fait est que toutes les forces qui habituellement contribuent à la dispersion et l'éparpillement lorsqu'une approche nouvelle voit le jour, se sont ici rassemblées, avec une unanimité remarquable, dans une même direction. Et cela, grâce à l'apport spécifique de chacun selon ses moyens et sa vocation: l'AFCET, l'ADEPA, l'Enseignement Technique, des Industriels, des Laboratoires Universitaires, etc...

Au vu de cet historique, le lecteur comprendra que nous insistions toujours, en présentant le GRAFCET, pour préciser qu'il n'est la propriété de personne mais le produit d'un travail d'équipe (les membres de la commission AFCET sont répertoriés en fin de cet ouvrage). A ce titre, il est au service de tous.

Pour conclure, nous soumettons au lecteur un sujet de réflexion: n'existe-t-il pas aujourd'hui d'autres « GRAFCET potentiels » ? Si oui, la démarche qui a été suivie reste valable. En confrontant des besoins réels que seuls les praticiens peuvent exprimer et des moyens disponibles que les théoriciens ont dégagés de leurs travaux, la collaboration Recherche-Industrie, loin d'être stérile, devient alors une source féconde d'innovation. Le GRAFCET en est l'illustration.

Michel BLANCHARD

2.2. ÉLÉMENTS DE BASE

Le Grafcet est un outil de modélisation de systèmes séquentiels et plus particulièrement pour les parties commandes d'automatismes. C'est un graphe qui comporte deux types de nœuds : les places et les transitions. Des arcs orientés relient les places aux transitions et les transitions aux places (comme pour les graphes d'états et d'événements).

L'étape :



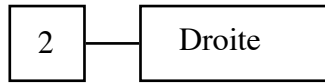
étape



étape initiale

Une étape peut être active ou inactive. Pour représenter une étape active, on peut par exemple

ajouter une marque dans l'étape. On peut associer une action à une étape (qui s'effectuera quand l'étape sera active) :

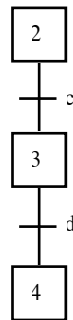


La transition :

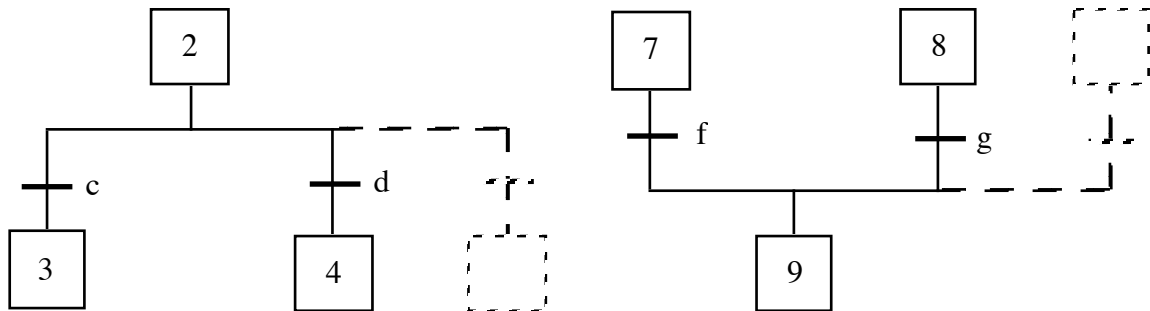


A côté de la transition, on doit indiquer la condition de validation (réceptivité). Si la condition est toujours vraie, on indique "1" ou "=1".

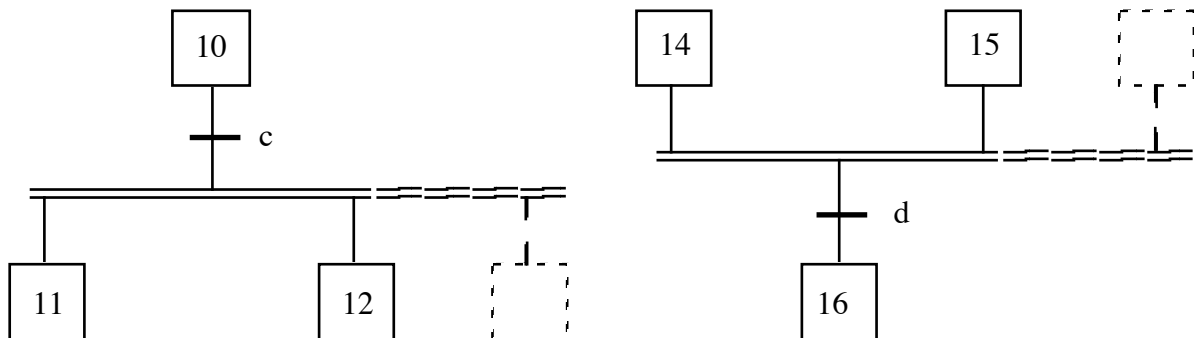
Séquence :



Sélection de séquence et fin de sélection de séquence : branchement OU :



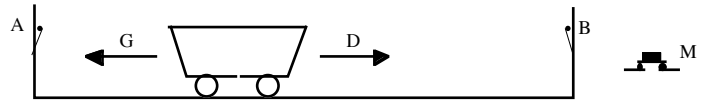
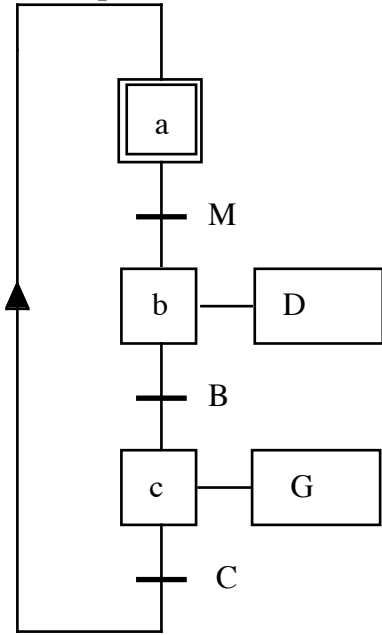
Séquences simultanées : branchement ET :



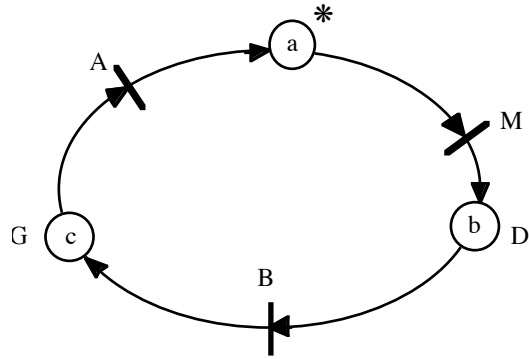
La divergence ET représente du parallélisme. La convergence ET représente une synchronisation.

Exemples de modélisation :

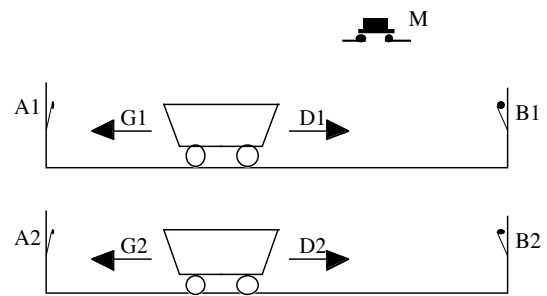
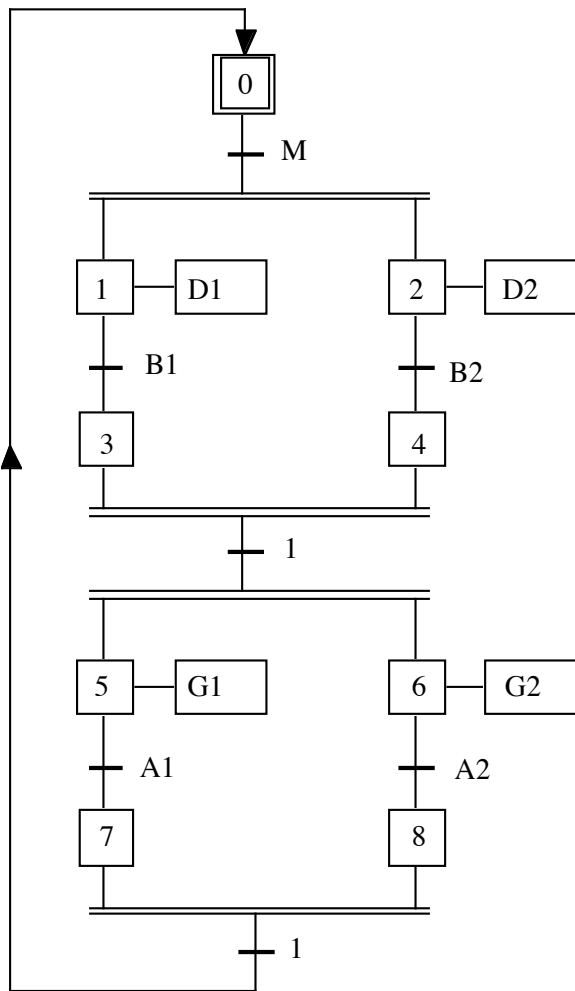
exemple du chariot :



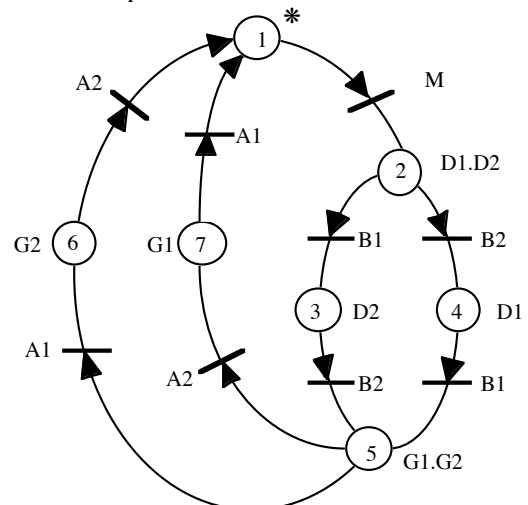
graphe d'état équivalent



exemple des deux chariots :



graphe d'état équivalent



Les numéros des états de ce graphe n'a rien à voir avec les numéros des étape du grafcet

2.3. RÈGLES D'ÉVOLUTION

On appelle situation du grafcet (à un instant donné) l'ensemble des étapes actives à cet instant. Elle correspond à un état du système. L'évolution de la situation se fait par franchissement de transitions.

Une transition est franchissable si et seulement si

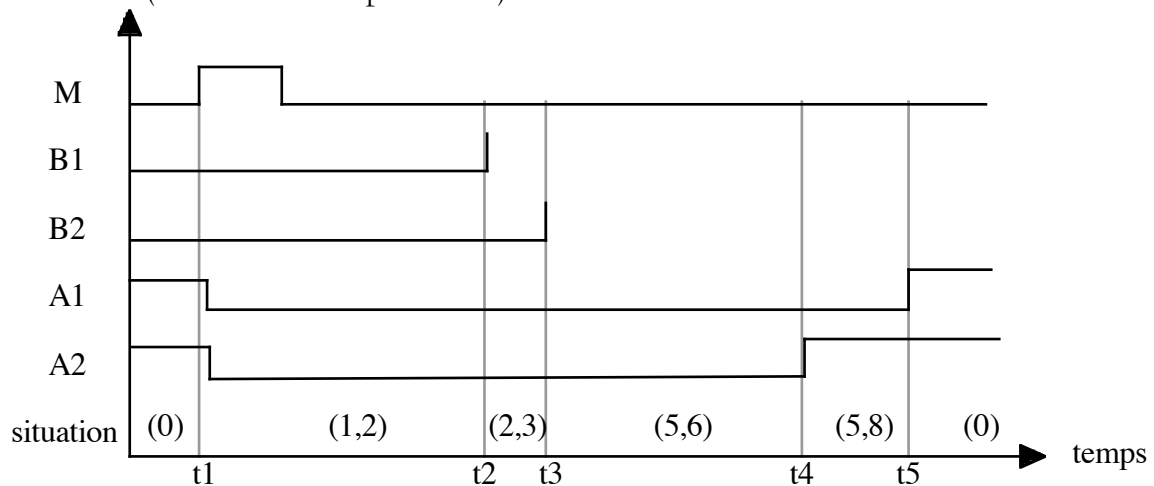
- les étapes en amont de la transition sont actives (on dit que la transition est alors validée)
- la réceptivité de la transition est vraie.

On franchit alors la transition en

- désactivant toutes les étapes précédant la transition,
- activant toutes les étapes suivant la transition,

simultanément.

Exemple de fonctionnement : le grafcet des deux chariots. A l'aide du grafcet précédent et du chronogramme ci-dessous, nous montrons l'évolution du grafcet en précisant à chaque instant quelle est sa situation (ensemble des étapes actives).



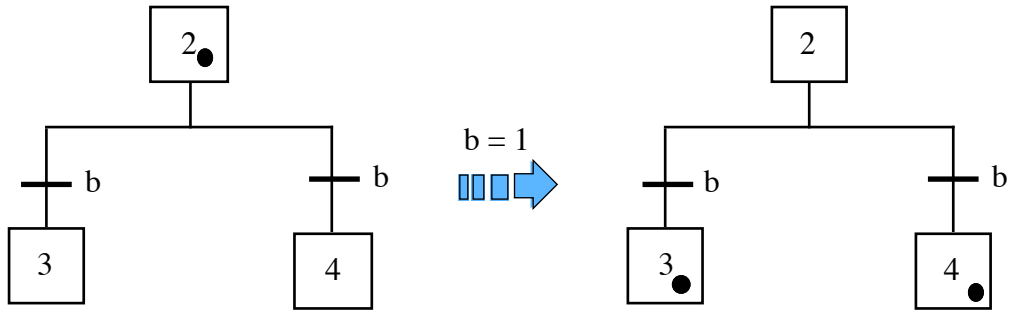
On note X_i la variable booléenne qui vaut 1 si l'étape i est active, 0 sinon. Par exemple, entre la date t_4 et t_5 , on a : $X_1 = X_2 = X_3 = X_4 = X_6 = X_7 = 0$ et $X_5 = X_8 = 1$.

Complétez le chronogramme ci-dessus et représenter les sorties ainsi que X_3 et X_5 .

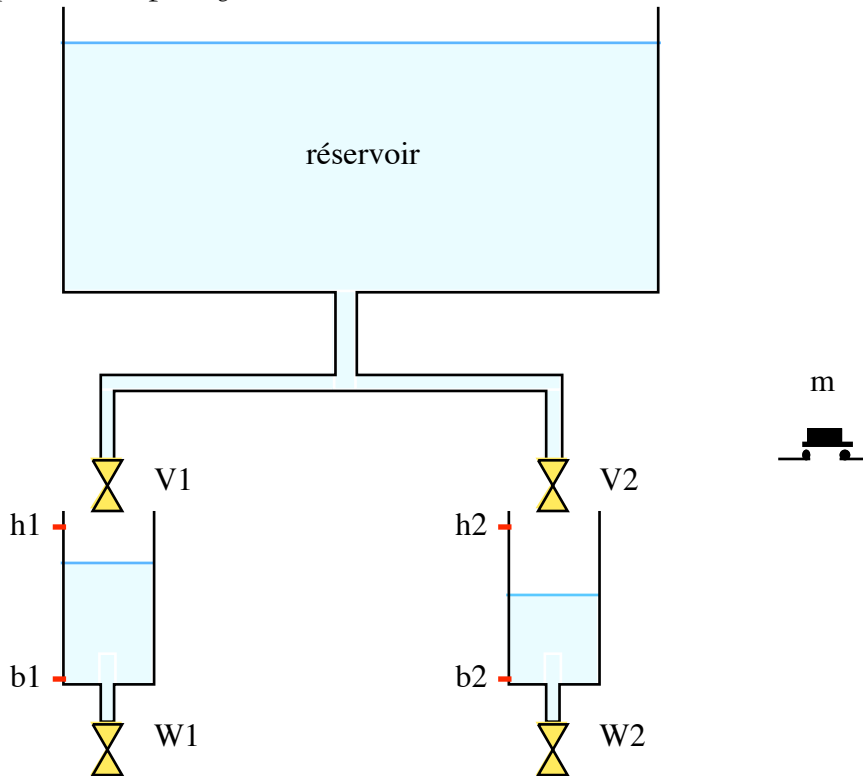
Règles de franchissement :

- Règle 1 : Toute transition franchissable est immédiatement franchie.
- Règle 2 : Plusieurs transitions simultanément franchissables sont simultanément franchies.
- Règle 3 : Lorsqu'une étape doit être simultanément activée et désactivée, elle reste active.

Illustration de la règle 2 dans le cas d'une divergence OU non exclusive :



Exemple de dispositif : remplissage de bacs.



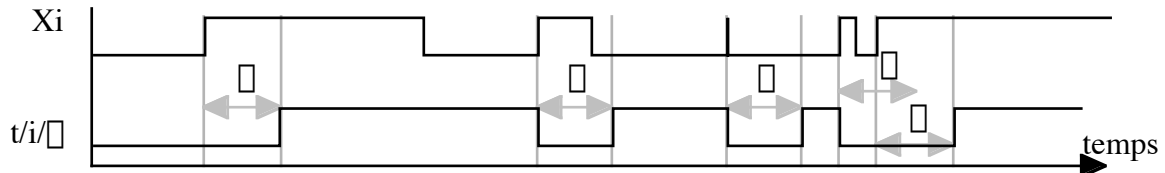
Le dispositif concerné est représenté sur la figure ci-dessus. Les deux bacs sont utilisés de façon similaire. Le bac 1 est vide lorsque le niveau est au-dessous de b_1 , c'est-à-dire $b_1 = 0$. Il est plein lorsque le niveau est au-dessus de h_1 c'est-à-dire $h_1 = 1$. A l'état initial, les deux bacs sont vides. Au moment où on appuie sur le bouton poussoir m , les deux bacs se remplissent grâce à l'ouverture des vannes V_1 et V_2 . Dès qu'un bac est plein, par exemple le bac 1, on arrête son remplissage (fermeture de la vanne V_1) et l'on commence à utiliser son contenu (ouverture de la vanne W_1). Lorsque le bac 1 est vide, on ferme la vanne W_1 . Le remplissage ne pourra recommencer que lorsque les deux bacs seront vides. Ce remplissage sera déclenché par appui sur le bouton poussoir m .

Faire un grafcet qui corresponde au fonctionnement décrit.

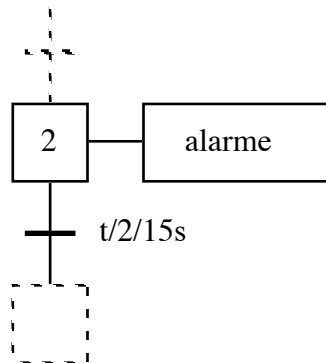
2.4. ACTIONS, SORTIES ET RÉCEPTIVITÉS

a) Le temps

On note $t/i/\Delta$ ou Δ/X_i la variable booléenne qui vaut 1 si et seulement si il s'est écoulé un temps au moins égal à Δ depuis la dernière fois que l'étape i est passée de l'état inactif à l'état actif. En d'autres termes, $t/i/\Delta$ vaut 0 pendant un temps Δ quand l'étape i devient active puis passe et reste à 1. Sur un chronogramme, cela donne :



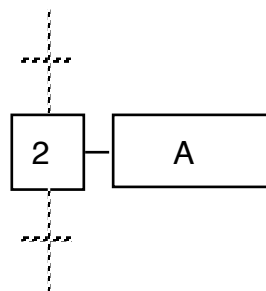
Exemple : pour maintenir une action pendant 15 secondes :



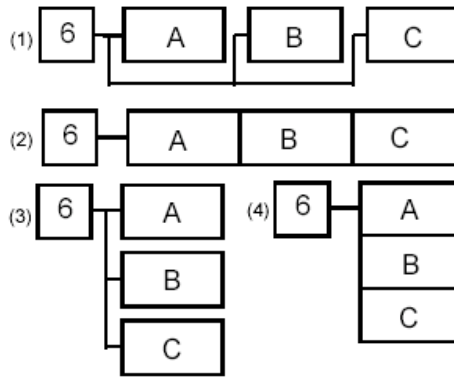
b) Les actions

Les actions associées à une étape sont inscrites dans un rectangle d'action de façon à mettre en évidence ce qui s'exécute lorsque cette étape est active. Souvent, il s'agira de commande d'actionneurs (vérins, moteurs, ...). Ce peut être aussi des commandes de fonctions auxiliaires d'automates (compteur, tempos, ...). Elles peuvent aussi décrire des liens avec d'autres systèmes logiques ou analogiques (changement de vitesse moteur par exemple).

Représentation normalisée :



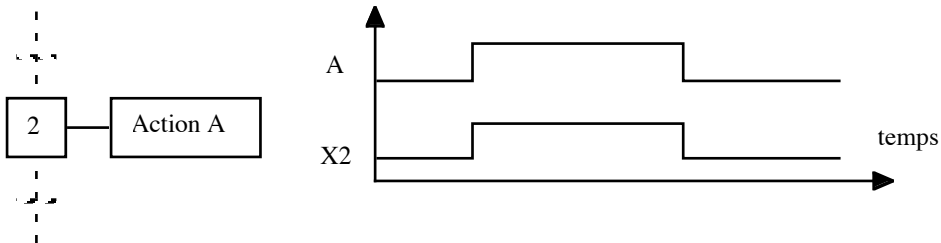
Au cas où l'on veut associer plusieurs actions à une étape, les représentations possibles sont décrites ci-dessous :



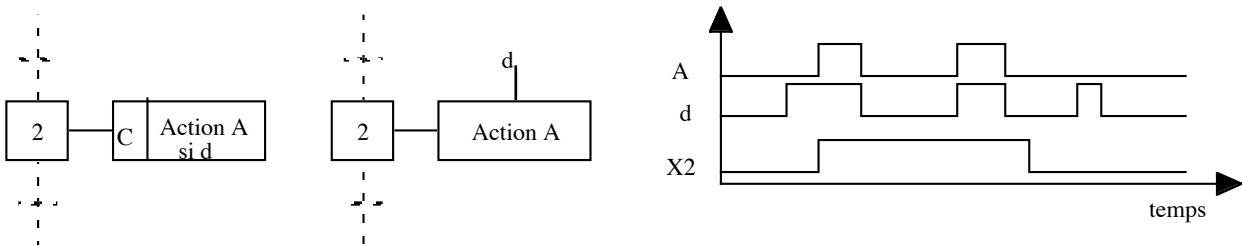
Exemples :

Action continue :

c'est la plus classique. L'action n'est effectuée que pendant l'activation de l'étape

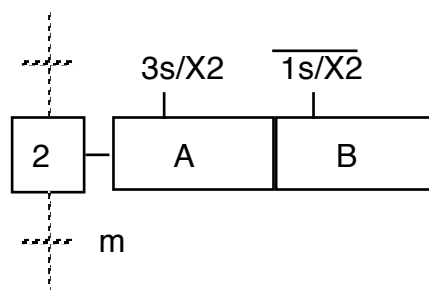


Action conditionnelle

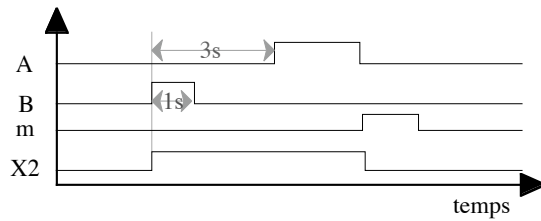


Actions temporisée ou retardée

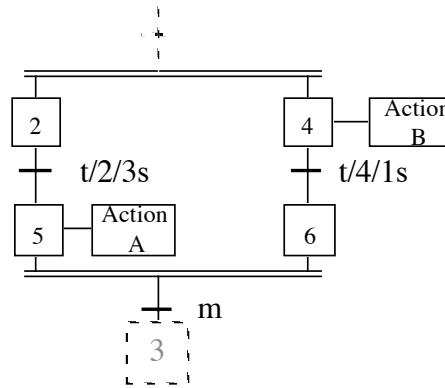
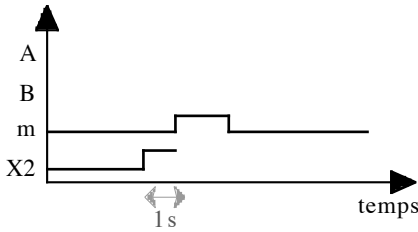
L'action A est retardée de 3s, l'action B dure 1s au plus (si l'étape 2 est active plus d'une seconde)



cela correspond au chronogramme :

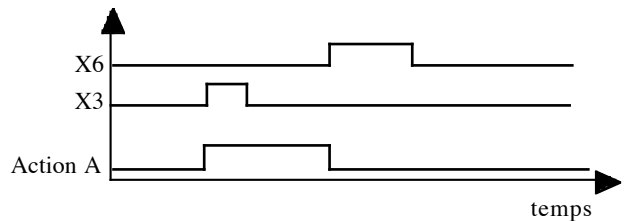
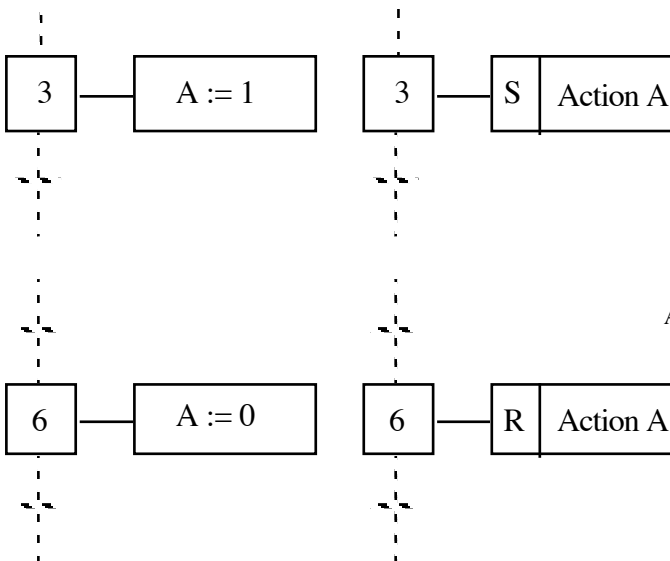


Montrez que le grafecet ci-dessous n'est pas équivalent aux grafecets ci-dessus en vous aidant du chronogramme ci-dessous.



Actions mémorisées.:

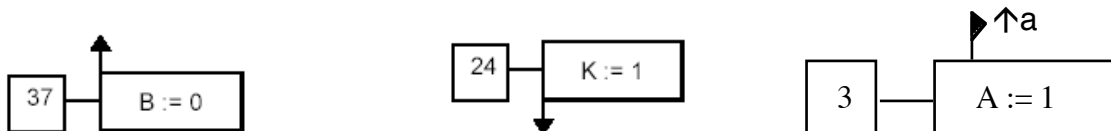
On peut pour cela utiliser des affectations ($A:=1$) (correspond à la nouvelle norme). On rencontre fréquemment l'utilisation de Set/Reset. Les affectations peuvent aussi s'appliquer à des variables (booléennes ou numériques) et à des compteurs (initialisation, incrémentation,...).



S = Set ; R = Reset

Actions impulsives à l'activation ou à la désactivation

Les affectations sont considérées impulsives et ont une durée théorique nulle. Ces actions peuvent être effectuées à l'activation, à la désactivation d'une étape ou sur l'occurrence d'un événement. Notations :



c) Réceptivités

Une réceptivité est une condition logique et/ou un événement. Une condition logique est une fonction booléenne de variables externes et de variables internes. Une variable interne est un Xi ou un bit contenant le résultat de la comparaison d'un compteur et d'une valeur. Une variable externe est l'information venant d'un capteur, d'un bouton ou d'un système extérieur. (On considère les variables $t/i/\square$ comme variable externe car la fin de tempo arrive de façon asynchrone par rapport à l'évolution du grafcet).

Exemples de conditions logiques : $a' + b.X4$, $(t/4/10s) + b.m$
 où a et b sont des capteurs et m un bouton.

Un événement est un front montant ou descendant d'une variable externe. Exemples :

$$\uparrow a, \quad \uparrow (t/4/10s), \quad \downarrow (a + b)$$

Propriétés :

$$\uparrow a = \downarrow a' \qquad \uparrow a.a = \uparrow a \qquad \uparrow a.a' = 0 \qquad \downarrow a.a' = \downarrow a$$

$$\downarrow a.a = 0 \qquad \uparrow a.\uparrow a = \uparrow a \qquad \uparrow a.\uparrow a' = 0$$

Si a et b et c sont indépendants :

$$\uparrow (a.b) = \uparrow a.b + \uparrow b.a \qquad \uparrow (a + b) = \uparrow a.b' + \uparrow b.a'$$

$$\uparrow (a.b).\uparrow (a.c) = \uparrow a.b.c$$

$$\text{Par hypothèse : } \uparrow a.\uparrow b = 0$$

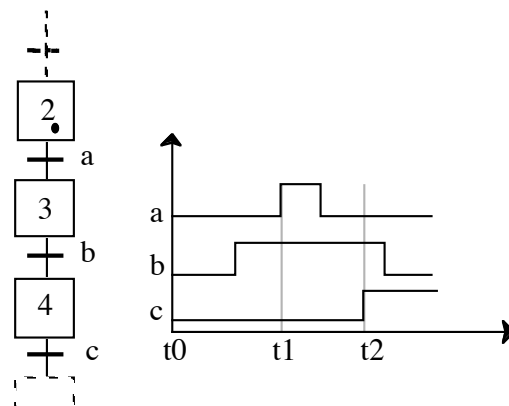
2.5 ALGORITHME D'INTERPRÉTATION DU GRAFCET

A partir d'un chronogramme quelconque des entrées, le Grafcet permet de connaître le chronogramme des sorties correspondantes. L'interprétation doit être sans ambiguïté. C'est l'objectif de l'algorithme d'interprétation. Nous formulons deux hypothèses :

- Deux événements externes non corrélés ne peuvent pas être simultanés.
- Un grafcet a le temps d'atteindre un état stable entre deux occurrences distinctes d'événements externes. (Le passage d'un état stable à un autre a une durée nulle).

Définition d'**état stable** et d'**état instable** :

Entre la date t_0 et la date t_1 , le grafcet est dans la situation $\{X_2\}$. C'est un état stable. A la date t_1 , le grafcet passe par l'étape 3 puis, immédiatement après (car $b=1$) il se retrouve dans la situation $\{X_4\}$. Cette situation correspond à un état stable jusqu'à la date t_2 . La situation $\{X_3\}$ à la date t_1 correspond à un état instable du grafcet. La durée de cet état est nulle.



L'algorithme d'interprétation

1. **Pas 1.** Initialisation : activation des étapes initiales et exécution des actions impulsionnelles qui y sont

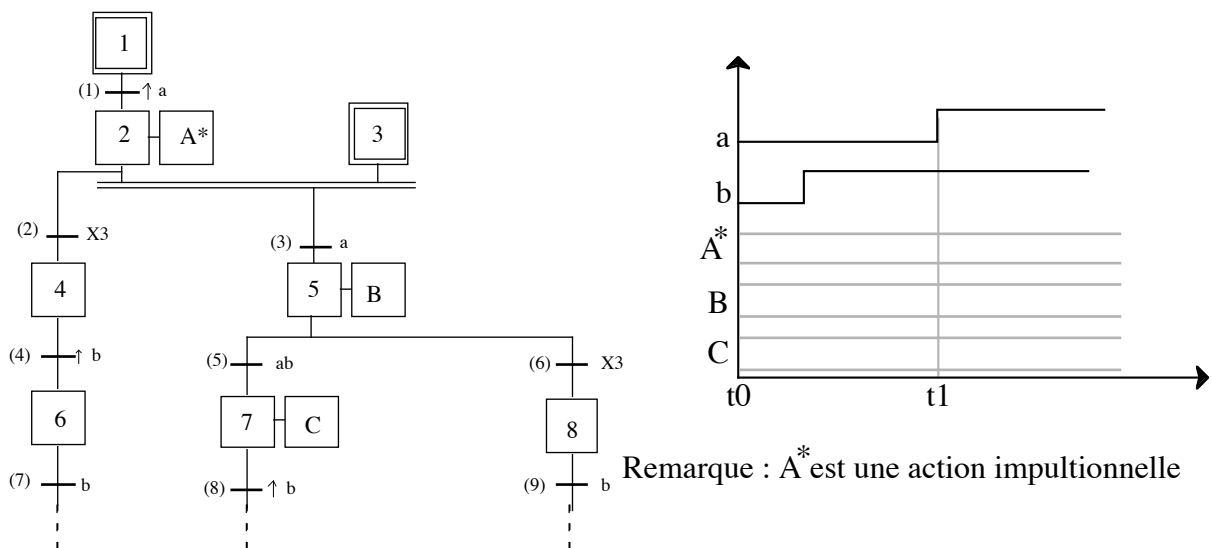
associées. *Aller au pas 5.*

2. **Pas 2.** Quand un nouvel événement externe se produit, déterminer l'ensemble T_1 des transitions franchissables sur occurrence de cet événement. Si T_1 n'est pas vide, aller au pas 3. Sinon, modifier éventuellement l'état des actions conditionnelles associées aux étapes actives (en effet, certaines actions peuvent dépendre de conditions dont les valeurs peuvent avoir changé). Attendre un nouvel événement externe au pas 2.
3. **Pas 3.** Franchir toutes les transitions franchissables. Si la situation est inchangée après ce franchissement simultané, aller au pas 6.
4. **Pas 4.** Exécuter toutes les actions impulsives associées aux étapes devenues actives au pas 3 (y compris l'initialisation des temporisations).
5. **Pas 5.** Déterminer l'ensemble T_2 des transitions franchissables sur occurrence de l'événement e (toujours occurs). Si T_2 n'est pas vide aller au pas 3.
6. **Pas 6.** Une situation stable est atteinte.
 - 6.1. *Pas 6.1.* Déterminer l'ensemble A_0 des actions à niveau qui doivent être désactivées (actions associées aux étapes qui étaient actives au pas 2 et qui sont inactives maintenant, et actions conditionnelles associées aux étapes restées actives pour lesquelles les conditions ne sont plus vérifiées).
 - 6.2. *Pas 6.2.* Déterminer l'ensemble A_1 des actions à niveau qui doivent être activées (actions associées aux étapes qui étaient inactives au pas 2 et qui sont actives maintenant éventuellement sous réserve de conditions, et actions conditionnelles associées aux étapes restées actives pour lesquelles les conditions sont vérifiées alors qu'elles ne l'étaient pas au pas 2).
 - 6.3. *Pas 6.3.* Mettre à 0 toutes les actions qui appartiennent à A_0 et qui n'appartiennent pas à A_1 . Mettre à 1 toutes les actions qui appartiennent à A_1 . *Aller au pas 2.*

Remarques :

- a. La boucle 3 -> 4 -> 5 -> 3 permet d'évoluer jusqu'à une nouvelle situation stable.
- b. Une action impulsive ou mémorisée (S et R) est exécutée même si la situation n'est pas stable. Une action à niveau n'est pas modifiée entre 2 situations stables. Le pas 6.3 assure la continuité des actions à niveau à 1.
- c. Grossièrement, on peut dire que, quand un événement externe se produit, le temps "s'arrête" et l'algorithme se déroule. Dès que l'on revient au pas 2 le temps repart jusqu'au prochain événement externe. (attention aux cycles instables : ça ne s'arrêterait plus).
- d. Cet algorithme peut être utilisé pour la mise en œuvre du grafcet.

Exemple de fonctionnement :

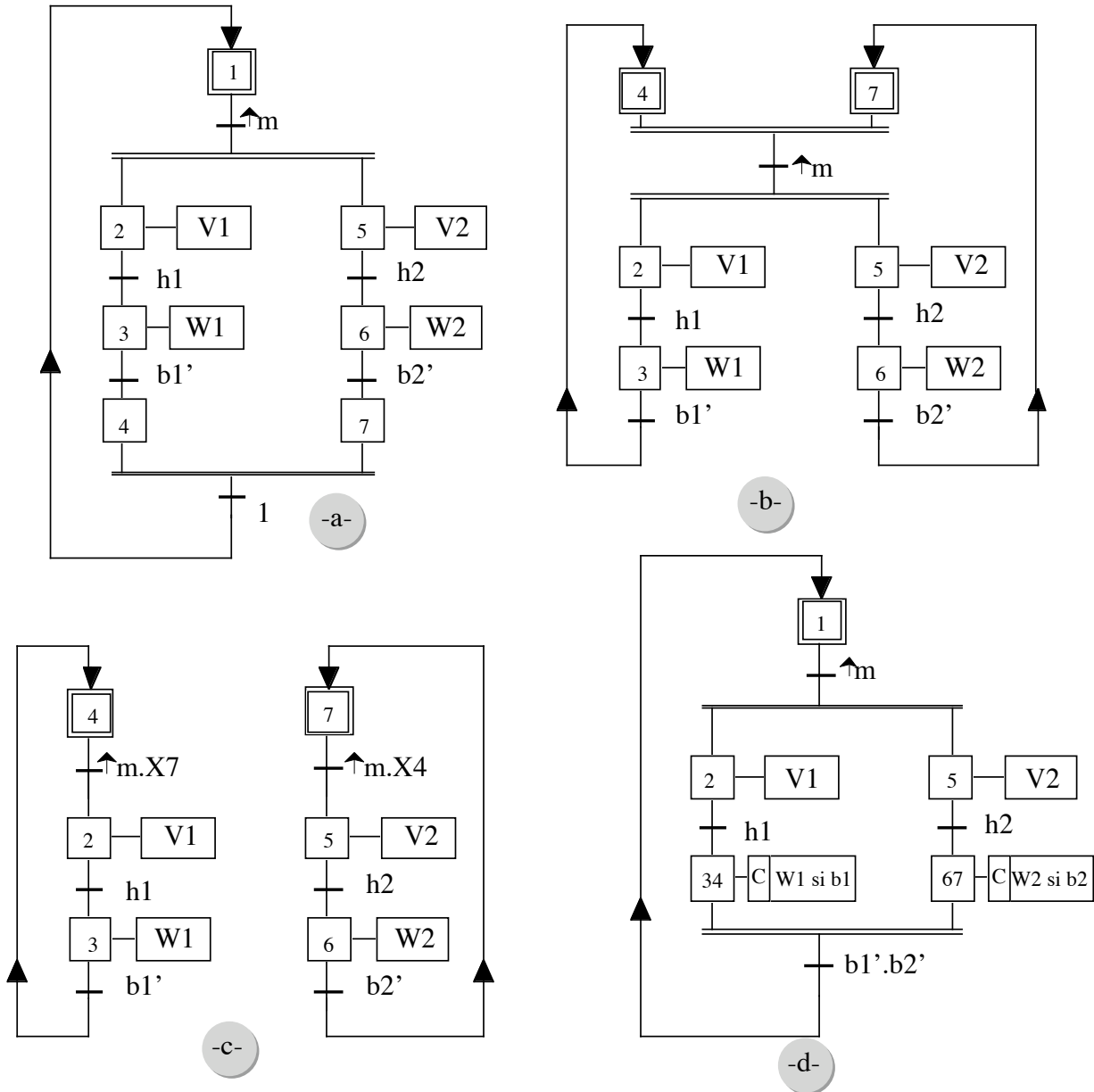


Complétez le chronogramme ci-dessus en tenant compte de l'algorithme d'interprétation.

Chapitre 3 : Compléments sur le grafcet

3.1 GRAFCET ÉQUIVALENTS

La façon de décrire par Grafcet un système n'est pas unique. L'exemple du remplissage des bacs (voir chapitre précédent) illustre bien le problème. Voici 4 grafcets décrivant le même système :

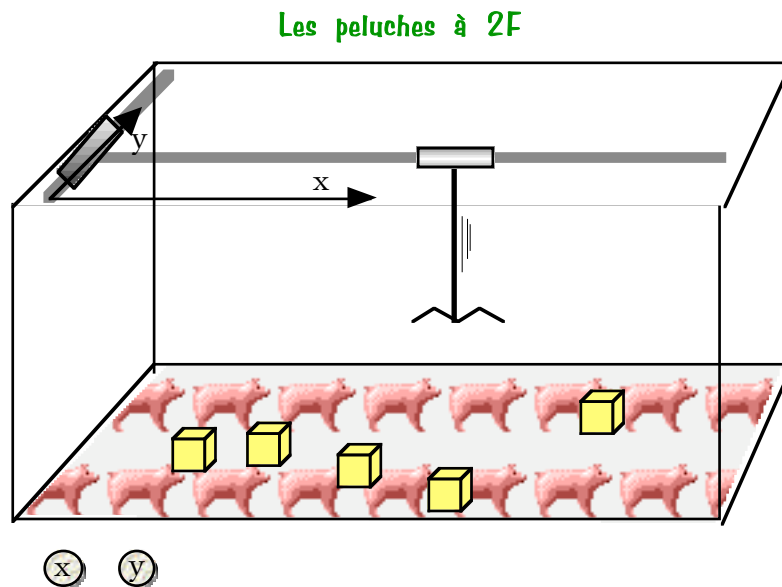


La figure -a- représente le grafcet construit 'intuitivement'. On remarque que quand l'étape 4 est active, cela signifie que le bac 1 est vide. Lorsque l'étape 7 est active, cela signifie que le bac 2 est vide. Lorsque l'étape 1 est active, les deux bacs sont vides. On peut donc se dire que l'étape 1 'est en trop'. On peut alors comprendre l'intérêt du grafcet -b-, où chaque étape a une signification simple. Le grafcet -c- est assez proche du grafcet -b- mais on 'remplacé' la synchronisation "ET" par une synchronisation s'effectuant à l'aide de variables internes X_i au niveau des transitions. On préfère,

lorsque cela n'alourdit pas trop le grafcet, ne pas utiliser de variables internes pour visualiser schématiquement les synchronisations. Le reproche que l'on pourrait faire au grafcet -d- est que les actions conditionnelles masquent l'aspect séquentiel des actions. En conclusion, c'est le grafcet -b- qui semble le mieux adapté.

3.2 ERREURS A NE PAS FAIRE

Dans l'exemple qui suit (et qui sera traité en cours), nous mettrons en avant les erreurs classiques et les "pièges" qui attendent les concepteurs.



Une machine permet de déplacer suivant les trois axes une pince dans un caisson fermé et vitré. Des peluches sont disposées au fond du caisson. Au départ, la pince est en haut et à l'origine des axes x et y . L'utilisateur choisit un des axes de déplacement (soit x soit y) en appuyant sur le bouton approprié. La pince se déplace jusqu'à ce que le bouton soit relâché ou que la valeur maximale soit atteinte. L'utilisateur peut alors faire de même avec l'autre axe (respectivement y ou x). Ce deuxième déplacement n'est pas effectué si l'utilisateur n'appuie pas sur l'autre bouton avant 10 secondes. Ensuite, la pince descend, se ferme, remonte, puis revient en position initiale et s'ouvre après 2 secondes. Si la pince avait attrapé une des peluches, l'utilisateur peut la récupérer.

Les entrées et sorties de la partie commande sont :

entrées		sorties	
1F	présence de 1F	X+	déplacement vers la droite
bpx	bouton déplacement sur x	X-	déplacement vers la gauche
bpy	bouton déplacement sur y	Y+	déplacement vers le fond
xo	position initiale sur x	Y-	déplacement vers le devant
yo	position initiale sur y	Z+	déplacement vers le haut
xmax	position maximale sur x	Z-	déplacement vers le bas
ymax	position maximale sur y	F	Fermer pince
zb	pince en bas		
zh	pince en haut		

Dans l'élaboration du grafcet de commande de cette machine, nous mettrons en avant les point

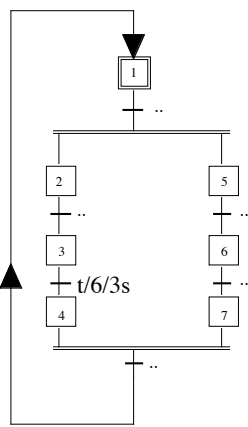
suivants :

a) Un OU n'est pas forcément exclusif

Comme nous l'avons vu dans les règles de fonctionnement du Grafcet, si deux transitions sont franchissables (même si ce sont deux transitions en 'concurrence' dans une disjonction OU) en même temps, elles SONT FRANCHIES TOUTES LES DEUX ! Si cela ne correspond pas à un fonctionnement que vous avez envisagé, des dysfonctionnements peuvent survenir, mettant éventuellement en péril la partie opérative et même l'utilisateur. En règle générale, il sera souvent souhaitable de

rendre les transitions d'une disjonction OU exclusives.

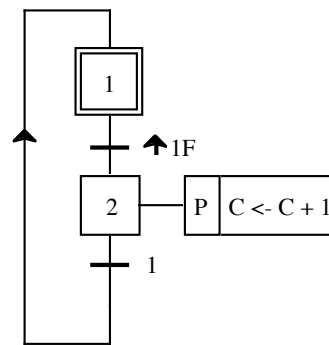
b) Les tempos restent à 1 jusqu'à la ré-activation de l'étape la concernant



Dans le grafcet ci-contre, on veut que la transition entre l'étape 3 et l'étape 4 ne doit s'effectuer que 3s après que l'étape 6 ait commencé à être active. Au premier tour, tout se passe bien et la variable $t/6/3s$ reste à 1 tant que l'étape 6 n'est pas active. Supposons qu'au deuxième tour on active l'étape 3 avant l'étape 6. La variable $t/6/3s$ est toujours à 1 et on passe directement à l'étape 4.

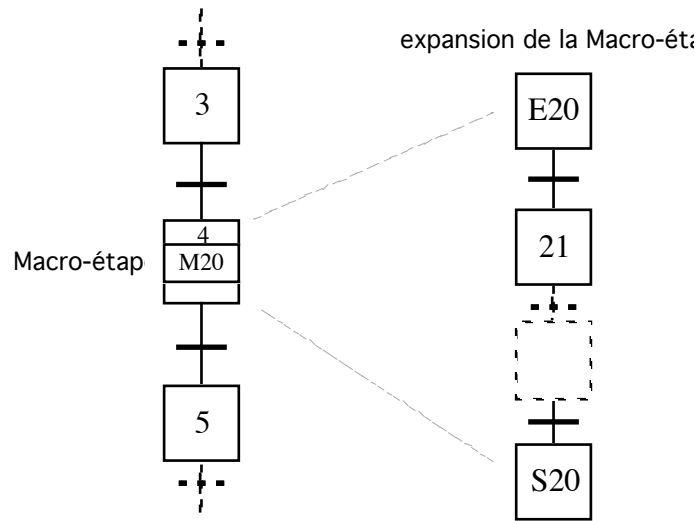
c) Le déroulement du Grafcet est, en théorie, à temps nul

Prenons à présent l'exemple d'un compteur qui sert à calculer le nombre de pièces de 1F qui passe devant un capteur optique. Le grafcet ci-contre réalise bien cela. Il faut souligner la présence du front montant dans l'expression de la réceptivité. En effet sans celui-ci, on risque de faire plusieurs fois la boucle étape1-étape2-étape1 pour la même pièce de 1F car l'automate est a priori très rapide. Remarque : on considère les actions de type compteur comme impulsionnelles.



3.3 STRUCTURATION PAR MACRO-ÉTAPES

Elles ont pour but de simplifier et de faciliter la description de systèmes complexes en allégeant le graphisme d'un grafcet et en détaillant séparément certaines parties. (à rapprocher des procédures en Pascal)



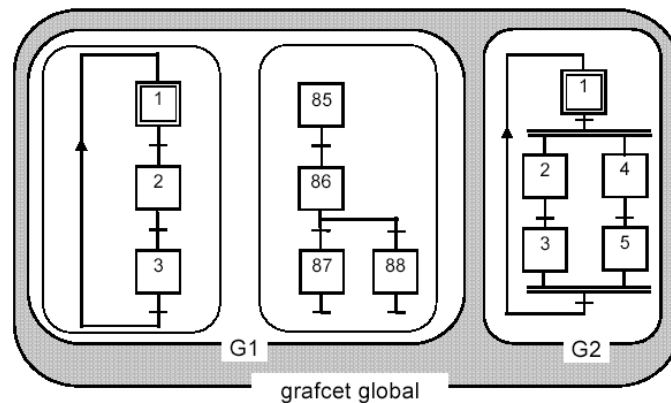
Norme :

1. Une expansion de macro-étape n'a qu'une étape d'entrée (notée E) et qu'une étape de sortie (notée S)
2. Tout franchissement d'une transition en amont de la macro-étape active l'étape d'entrée de son expansion.
3. L'étape de sortie de l'expansion participe à la validation des transitions en aval.
4. Il n'existe aucune liaison orientée qui arrive sur l'expansion de la macro-étape ou qui en parte.

Remarque : Une expansion ne peut être commune à plusieurs macro-étape (d'après la norme, mais certains disent le contraire)

3.4 STRUCTURATION DES GRAFCETS PARTIELS

L'ensemble des étapes et transition d'une commande peut être divisée en plusieurs grafkets connexes. Ces grafkets peuvent être regroupés en sous ensembles (souvent d'un seul élément) que l'on appelle grafket partiel. La réunion de ces grafkets partiel décrit la commande du système considéré. Chacun de ces grafkets partiel peut avoir un nom ou un numéro.



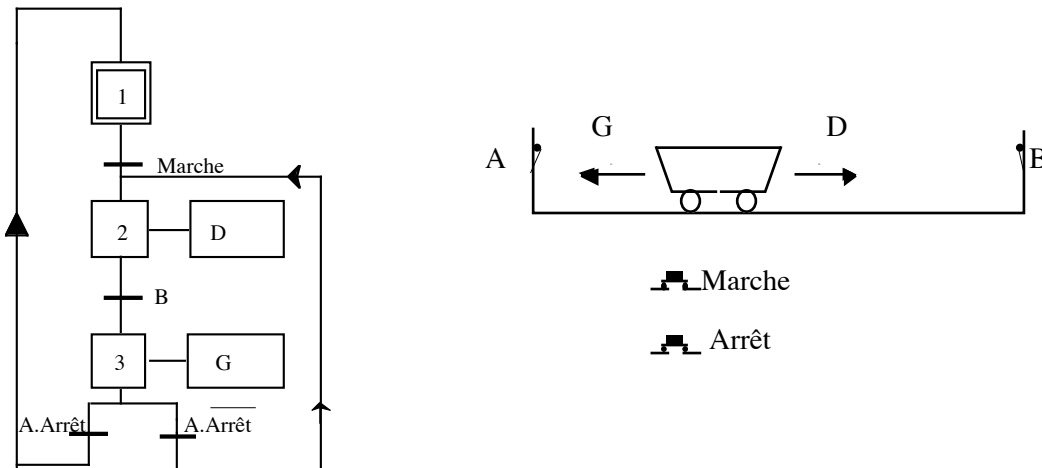
Notations :

- $G\#$ désigne le grafcet partiel # (ex : $G1, G2\dots$)
- $XG\#$ est une variable booléenne. Elle est active (vaut 1) si une étape au moins du grafcet partiel $G\#$ est active.
- $G\#\{ \dots, \dots \}$ désigne la situation d'un grafcet partiel. Exemple : $G1:\{1,86\}$.
- $G\#\{*\}$ désigne la situation courante du grafcet #.
- $G\#\{ \}$ désigne la situation vide du grafcet partiel #.
- $G\#\{init\}$ désigne la situation initiale du grafcet partiel #

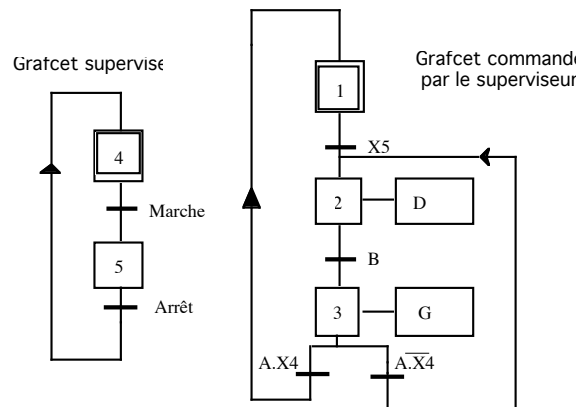
3.5 STRUCTURATION PAR SYNCHRONISATION ENTRE GRAFCETS PARTIELS

On peut structurer les grafquets partiels pour que l'un d'eux puisse jouer un rôle de Grafcet superviseur par rapport à d'autres.

Exemple : Un système automatisé fonctionne selon un cycle 'autonome' : Il s'agit d'un chariot qui fait des navettes entre un point A et un point B. On veut tout de même le commander grâce à deux boutons : Marche et Arrêt. C'est à dire que le chariot commence à effectuer ses cycles qu'une fois que le bouton Marche aura été appuyé et il arrêtera une fois son cycle terminé si l'on a appuyé sur le bouton Arrêt. Proposition de représentation :



Cette solution a un inconvénient : pour arrêter le chariot il faut appuyer sur Arrêt au moment où le chariot arrive en A. Ce n'est pas satisfaisant. L'idée est d'avoir un grafcet qui 'écoute' les boutons et qui commande l'autre. Ce sera un grafcet superviseur :



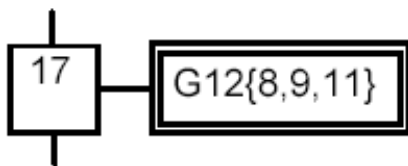
Dans les cas plus généraux et plus complexes, le grafcet superviseur s'occupe du démarrage, de la séquence, de la synchronisation et de l'arrêt de différentes tâches. Chaque tâche est décrite par un grafcet commandé par le superviseur (comme celui ci-dessus à droite).

Remarques : On organise ces grafcets de façon à ce que seul, le grafcet superviseur s'occupe des boutons de commande.

3.6 STRUCTURATION PAR FORÇAGE D'UN GRAFCET PARTIEL

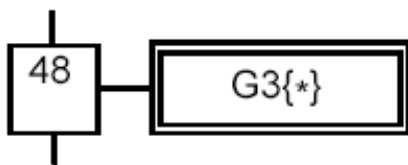
Lorsque l'on veut prendre en compte la sécurité, on risque d'avoir un accroissement important de la complexité d'un grafcet. Au lieu de cela, on peut imaginer qu'un grafcet peut avoir une influence globale sur un autre grafcet par l'utilisation d'actions spéciales : les macro-actions.

Remarques : à un ensemble de grafcets avec des macro-actions, on peut toujours faire correspondre un ensemble de grafcets sans macro-actions.



EXEMPLE 1: Forçage d'un grafcet partiel à une situation déterminée.

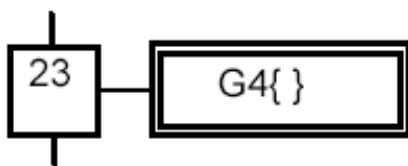
Lorsque l'étape 17 est active, le grafcet partiel 12 est forcé dans la situation caractérisée par l'activité des étapes 8, 9, 11.



EXEMPLE 2: Forçage d'un grafcet partiel à la situation courante.

Lorsque l'étape 48 est active, le grafcet partiel 3 est forcé dans la situation où il se trouve à l'instant du forçage.

NOTE 1 On appelle également cet ordre «figeage».



EXEMPLE 3: Forçage d'un grafcet partiel à la situation vide.

Lorsque l'étape 23 est active, le grafcet partiel 4 est forcé dans la situation vide (voir symbole 36).

NOTE 2 Dans ce cas aucune des étapes de G4 n'est active.

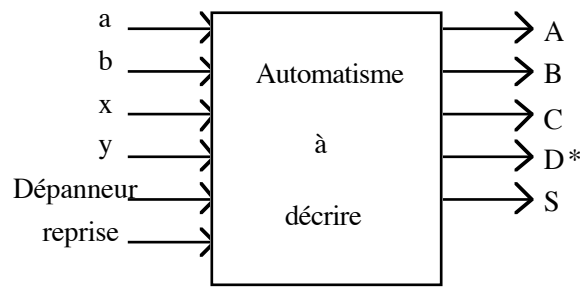


EXEMPLE 4: Forçage d'un grafcet partiel à la situation initiale.

Lorsque l'étape 63 est active, le grafcet partiel 8 est forcé dans la situation dans laquelle seule ses étapes initiales sont actives.

Exemple d'application :

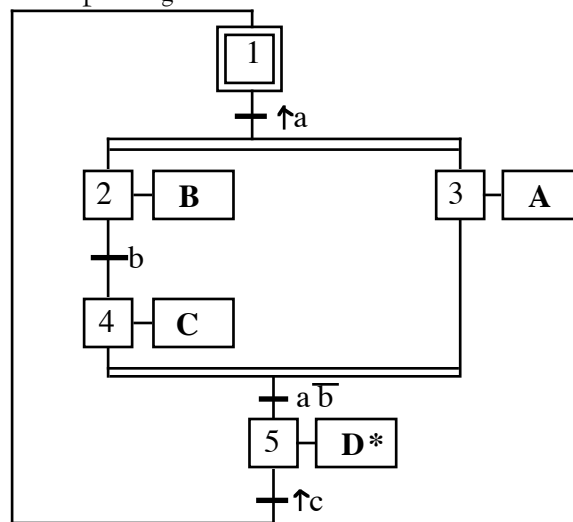
Voici le schéma de l'automatisme à décrire :



a, b, A, B, C, D* sont les Entrées-Sorties en fonctionnement normal (voir grafcet G1).
x et y sont deux défauts possibles.

S est l'alarme.

L'automate sans défaut est décrit par le grafcet G1 ci-dessous :



Si l'un des défauts apparaît, le grafcet G1 est ré-initialisé et l'alarme sonne. Si les défauts disparaissent, le fonctionnement peut reprendre.

(grafcet G2)

à faire

Lorsqu'un dépanneur arrive, il peut arrêter l'alarme (même si le défaut n'a pas disparu) en appuyant sur le bouton "dépanneur arrivé".

(grafcet G3)

à faire

Si le 1er défaut qui est apparu se produit 2 fois en moins d'une minute, il faudra une intervention manuelle (bouton reprise) pour que le fonctionnement reprenne.

(grafcet G4)

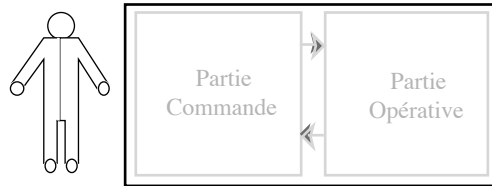
à faire

On peut tenter de décrire 2°, 3° et 4° en complétant le grafcet du 1° mais on préfère décrire un grafcet pour chaque partie.

3.4 NOTION DE POINT DE VUE

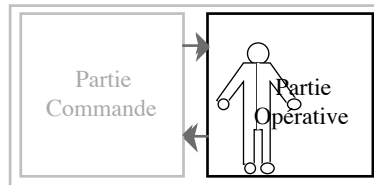
Le Grafcet permet de représenter le fonctionnement d'un automatisme séquentiel. Toute description dépend du point de vue que l'on a. Il existe classiquement 3 points de vue pour l'automatique séquentielle correspondant à une méthodologie classique par affinements successifs permettant de passer "du problème posé" au choix des procédés opératifs et au choix des équipements de la partie commande.

Le point de vue système ou procédé :



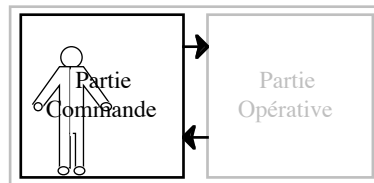
description du fonctionnement vue de l'extérieur

Le point de vue Partie Opérative :



L'observateur connaît la partie opérative, le choix des actionneurs et des capteurs

Le point de vue Partie Commande :

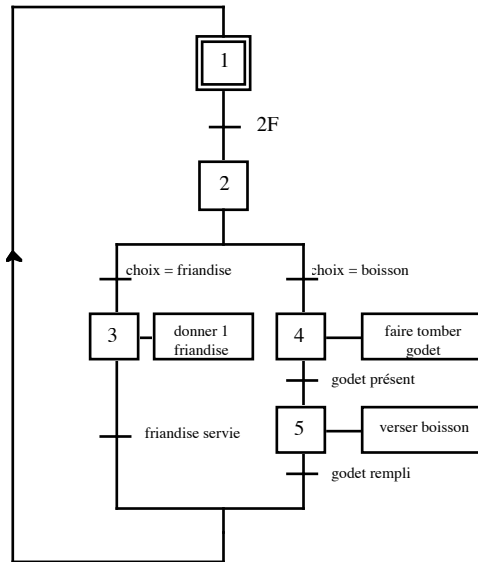


Grafcet vu selon le point de vue réalisateur

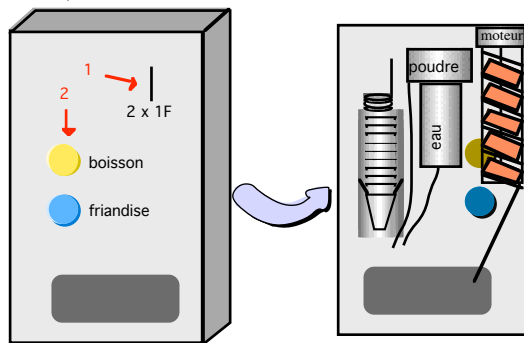
Exemple :

Un distributeur de boissons et de friandises. Un distributeur automatique permet d'obtenir pour la somme de 2 francs soit une boisson (toujours la même) soit une friandise (toujours la même). Le client doit introduire l'argent dans une fente puis choisir "boisson" ou "friandise". Le distributeur lui sert aussitôt ce qu'il a commandé.

Grafcet point de vue système :



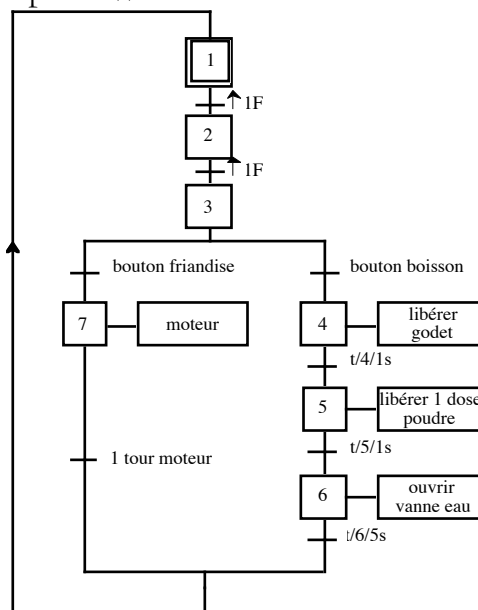
Vous êtes maintenant au courant de ce qui a été choisi comme partie opérative. Vous connaissez la liste des capteurs et des actionneurs.



capteurs : 1f, Bouton boisson, Bouton friandise, 1 tour moteur (pour libérer 1 friandise)

actionneurs : libérer godet, libérer 1 dose poudre, ouvrir vanne eau, moteur.

Le grafecet point de vue opérative peut ressembler à :



Le grafecet point de vue Partie Commande ressemble au grafecet point de vue Partie Opérative à ceci

près que les entrées et sorties sont écrites sous forme de mnémoniques et font référence aux entrées et sorties de l'automate. (En fait, il y a d'autres différences mais nous n'en parlerons pas ici).

3.5 APPROCHE FONCTIONNELLE

Pour concevoir un grafset, il convient d'utiliser une méthode intuitive consistant à :

- tracer l'étape initiale,
- tracer la première action
- prévoir les évolutions possibles
- compléter en détaillant tous les cas.
- vérifier que l'on a rien oublié, corriger, simplifier.

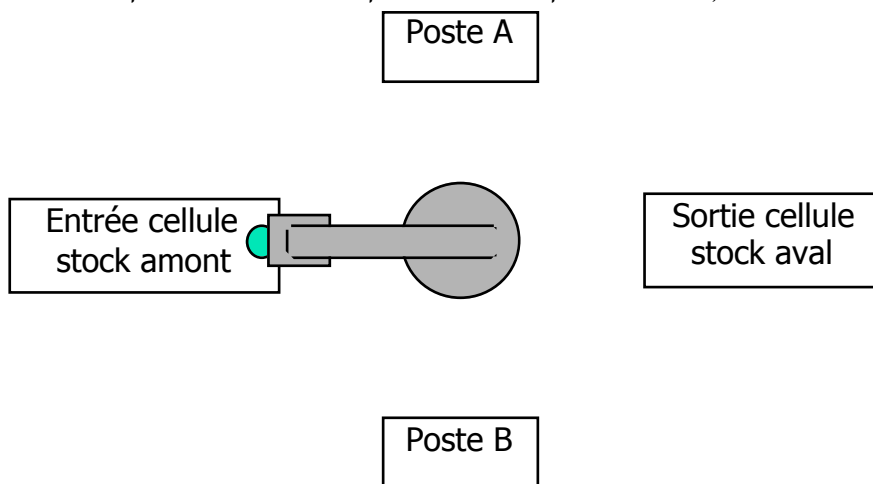
Si cette méthode peut fonctionner pour des systèmes simples ou très séquentiels (peu de choix OU), elle trouve assez vite ses limites pour des systèmes complexes.

Pour illustrer ces propos, essayer de faire le grafset du problème suivant :

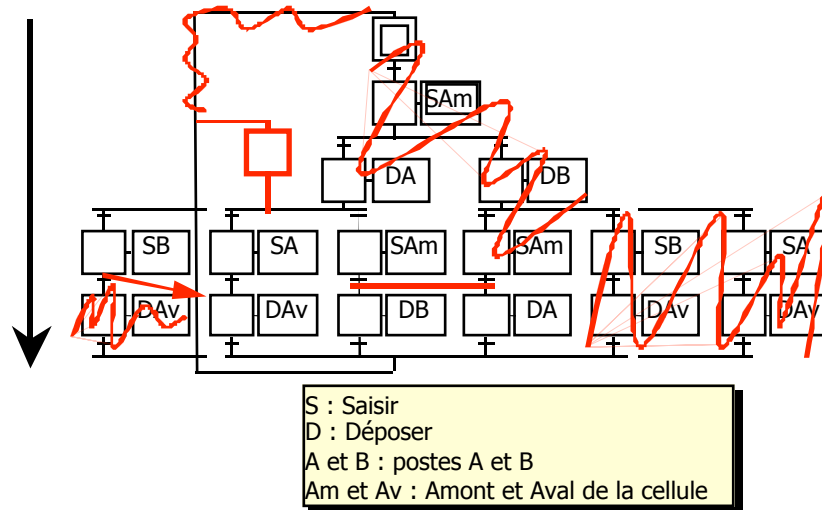
Soit un robot chargé de gérer un flux de pièces à travers une cellule composée de 2 postes :

les pièces qui se présentent en amont de cette cellule subissent une opération qui peut être réalisée aussi bien sur le poste A que sur le poste B, avant de rejoindre le stock en aval.

Le robot est en attente de l'arrivée d'une pièce au stock amont. Une pièce se présente, le robot la saisit puis la dépose au poste A. Une autre pièce se présente, le robot va la prendre et la dépose en B (car le poste A est occupé). Si une autre pièce se présente, il ne la prend pas car les deux postes sont occupés. Le poste A a fini, le robot va chercher la pièce et la dépose en sortie. Le robot peut maintenant s'occuper de la pièce arrivée pour la déposer en A (ou chercher la pièce en B si elle est prête et selon la priorité donnée).



Vous pouvez obtenir :

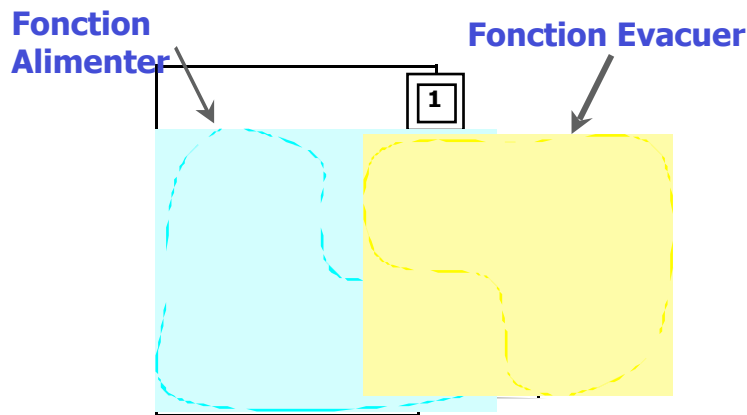


Cette façon de procéder n'est pas efficace car c'est une démarche sans rigueur, elle conduit à plusieurs phases d'essais et un tracer peu clair et conduit à des erreurs (tous les cas ne sont pas forcément prévus et il est difficile de trouver toutes les simplifications possibles).

L'approche fonctionnelle consiste en :

- référencer les entrées et sorties
- définir la fonction principale (en utilisant du vocabulaire général)
- décomposer et définir les sous fonctions avec mise en évidence des structures de base du grafcet (ébauche)
- la synthèse donne la structure générale du grafcet
- reste à compléter les réceptivités...

Dans l'exercice précédent, l'approche fonctionnelle donne :



BIBLIOGRAPHIE

Norme internationale CEI 60848, Langage de spécification Grafcet pour diagrammes fonctionnels en séquence, Deuxième édition, février 2002

R. David, H. Alla, Du grafcet aux réseaux de Petri 2^{ième} édition, Traité des nouvelles technologies, série Automatique, Hermès, 1992, 499p.

ouvrage de base pour toutes les parties (A, B et C), présentant les notions et les principes généraux. Contient des exos corrigés.

M. Blanchard, Comprendre et maîtriser le Grafcet, Automatisation et production, Capadues éditions, 1979, 169p.

La norme grafcet présentée par un de ses créateurs.

GREPA, Le Grafcet : de nouveaux concepts, Automatisation et production, Capadues éditions, 1985, 104p.

Il s'agit de la suite du précédent.

M. Courvoisier, R. Valette, Commande des procédés discontinus, logique séquentielle, Dunod université, 1986, 134p.

introduction aux systèmes séquentiels (graphes d'états, tableaux), exemples de réalisations matérielles et logicielles.

Les automatismes programmables, Automatisation et production, Capadues éditions, 1987, 185p.

Bon ouvrage sur les généralités et l'environnement des automates, le Gemma. Quelques exemples concrets.

S. Moreno et E. Peulot, Le Grafcet : conception - Implantation dans les API. Editions Casteilla, Collection A. Capliez, 1996, 251p.

Bon ouvrage avec des exemples concrets.

M. Pinot, R. Jégoux, J.P. Maillard, Du Grafcet aux automates programmables, Edition Foucher, 1991, 127p. *Sur les mode de programmation, les actionneurs, pré-actionneurs, capteur, la pneumatique, etc...*

S. Thelliez, J.M. Toulotte, Applications industrielles du Grafcet, Eyrolles 1985, 187p.

C. Sourisse, Les automatismes industriels, Technologies de pointe, Hermès 1989, 63p.

J.C. Bossy, P. Brard, P. Faugère, C. Merlan, Le Grafcet, sa pratique et ses applications, Educalivre 1979.

Bernard Reeb, Développement des grafkets. Editions Ellipses, 1999, 190p.

Autres sources

J.P. Denat, Maître de Conférences à l'ESIA (Ecole Supérieure d'Ingénieurs d'Annecy), Polycopié de son cours de logique séquentielle.

G. Vitry, Maître de Conférences à l'Ecole Centrale de Lyon, Polycopié de son cours d'automatique des systèmes à événements discrets.

B. Riera, Professeur à l'IUT de Champagne Ardennes, Présentations de son cours d'automatique séquentielle 2003.