ISSN 0291 - 5413

Informatique fondamentale et Applications

Comité de rédaction

E. Bianco
G. Cousin
F. Donnat
P. Isoardi
J.P. Lehmann
J. Roller
R. Stutzmann

Dépositaire

G. Ambard

Sommaire

-Editorial ...P 3

-Remarques sur deux types de modèles en théorie des automates ...P 8

-Le Grafcet, études et réflexions ...P 37

-Vouzzavedibisar ...P 66

Juin 1983

Faculté des Sciences de Luminy
Département de Mathématique-Informatique
70, route Léon Lachamp - Case 901
13288 MARSEILLE CEDEX 9

Editorial

e. bianco

INFORMATIQUE TOUT COURT.

Une brise marine entre par la fenêtre,dehors brille un soleil presque vif,la lumière ruissèle...encore un effort,les vacances sont là.Parler toujours d'informatique,quelle tâche ingrate,surtout parvenus au bout de toutes les fatigues de l'année.Pourtant comment ne pas constater que cette diablesse est partout,elle s'infiltre dans tous les recoins de notre vie,elle nous imprègne elle force notre porte,s'impose sur les ondes,elle nous envahit, elle nous submerge.

C'est à la fois insidieux et inéluctable.Mais qu'est-ce que c'est vraiment que l'informatique?

Sur fond de claire verdure et dans le bruissement doux de l' eau qui s'étire sur le sable,essayons de réfléchir à ce phénomène.

Un outil est né il y a peu de temps finalement,mais il semble que cela fait déja des siècles.Et cet outil paraît si bien adapté à notre société,qu'on est en droit de se demander s'il existe encore quelqu'un qui ne lui ait attribué aucune utilité.

Son usage s'est développé si vite et si grand a été le bouillonnement des idées qu'il a inspiré,qu'un mythe traîne déja derrière lui;et bien souvent ce mythe remplace simplement la réalité,comme on ne l'a vu que trop souvent dans le domaine de la politique,par exemple.

Une ambiguïté subsiste dans ce que je viens de dire,elle est

de taille,mais cela,me semble-t-il n'est que le reflet de la réalité.J'ai parlé d'outil,et derrière se profilent deux images l'une floue: l'informatique,l'autre plus nette,l'ordinateur.Il y a bien souvent confusion,et des phrases telles que: "L'informatique a donné son verdict..." ponctuant l'émission de résultats par l'ordinateur,ne sont pas rares.Or,si nous savons très précisément ce qu'est un ordinateur,je ne serais pas aussi affirmatif en ce qui concerne l'informatique.Et c'est pour essayer d'éclairer les choses en ce domaine et dans la mesure du possible,que j'aimerais un peu insister.

Le problème majeur qui s'est posé depuis sa naissance a été de rendre l'ordinateur plus facilement abordable.En fait,cette machine s'est révélée,depuis que l'humanité conçoit des machines celle dont le mode d'emploi est,de loin,le plus complèxe.Et il a sans doute fallu qu'elle corresponde à quelquechose de bien profond dans la psychologie de l'homme du XXe siècle,ou dans l'organisation de sa société,pour qu'elle obtienne le succès constaté. Masochisme compris.Or,cette difficulté de manipulation induit deux préoccupations importantes,dont l'une concerne tout l'art d'écrire des programmes,et l'autre renvoit aux préoccupations d'une bonne gestion de l'approche de l'engin.

L'ordinateur a donc été utilisé lui-même pour résoudre au moins en partie le problème.On a créé des langages plus faciles à utiliser,et l'ordinateur a servi à les compiler,c'est-à-dire à mettre des phrases de tels langages sous une forme par lui compréhensible.On a muni l'ordinateur de gros programmes de gestion interne,destinés à manipuler automatiquement tout ce qui pouvait l'être; en fait à faire en sorte,si j'ose utiliser une expression

inadaptée,mais qui fait image,que chaque utilisateur ne retienne l'attention de l'ordinateur qu'un minimum de temps.

Les organes obtenus: les compilateurs et les systèmes,se sont assez rapidement figés dans leur structure,mais se sont retrouvés très diversement modulés selon les domaines d'application dont l'évantail s'est lui aussi rapidement étendu.

Au fur et à mesure que l'ordinateur rend des services,sa renommée s'étend et de nouvelles branches d'application se créent spontanément.Et chaque fois,il faut,pressé par la demande,réaliser rapidement.Ainsi sont nées: la gestion,la bureautique,la conduite des processus,la recherche documentaire,et sur des plans plus abstraits,la cybernétique,l'intelligence artificielle,j'en passe la liste est longue et sans grand intérêt.Une rapidité d'évolution exceptionnelle,un marché d'une importance fantastique; Par ailleurs une situation de crise économique poussant à des reconversions peut-être un peu hâtives; Une administration un peu acculée à mettre en évidence qu'elle ne peut rien résoudre,avec peut-être,entre mille autres raisons,l'incapacité d'organiser sur le terrain de véritables enquêtes,préférant tenir compte des potins de la Cour.

Voila le milieu dans lequel s'est développée l'informatique. Daus cette situation de panique où les coups de barre se donnent à tort et à travers,les spécialistes eux-mêmes ne cherchent bien souvent qu'à saisir l'opportunité pour assurer la survie.

De quelle nature est le fond du problème et où se situe-t-il? me paraît être une question importante.

Entre deux chants d'oiseaux qui me ravissent l'âme,car l'air frais du matin m'apporte un message apaisant et plein de douces

promesses,je vais tenter de mettre un peu de bleu dans ma pensée.

Je vois l'informatique comme une étoile aux rayons multiples et foisonnants.Ce qui fait la substance même de l'étoile,alimente en permanence la vigueur des rayons.Cette vision lumineuse inspirée des longues et claires nuits d'été n'est pas destinée à illustrer toutes les facettes de cette envahissante branche d'activités.Je dirai simplement que si le rayonnement de l'informatique se matérialise dans diverses disciplines que je baptiserais volontiers: techniques de l'informatique,il existe un fonds commun de connaissances indispensable pour leur développement.

Qu'il s'agisse de "gestion",de "bureautique",de "contrôle de processus","cybernétique" ou autres,aucune de ces branches ne présente d'intérêt fondamental pour le développement des autres.

Par contre,si l'on considère la trilogie constituée par le processeur,le compilateur,le système,pris dans leur sens le plus large,on s'apperçoit qu'aucune des branches ci-dessus ne saurait se passer d'aucun de ces éléments.

Il me paraît donc naturel de déclarer que la matière fondamentale de l'informatique sera constituée de l'ensemble des préoccupations qui gravitent autour de ces trois éléments.A partir de cet instant,on peut définir les buts d'une recherche fondamentale,qui devra aboutir d'une part à rechercher de nouvelles structures de ces objets fondamentaux quitte à en bouleverser les limites,d'autre part à les adapter au mieux des nécessités imposées par le domaine d'application choisi.

La nature de l'enseignement de l'informatique pourrait également se définir à partir de ces considérations.

Et,par exemple,le raccord aux mathématiques se fait alors tout naturellement par l'intermédiaire d'une épi-informatique constituée de la logique,de la combinatoire,des algèbres de Boole et toutes autres théories qui porteraient sur des objets utiles à l'informatique.Une certaine forme d'élèctronique,par exemple celle des circuits intégrés à commutation,pourrait être également considérée comme une épi-informatique.

Je ne voudrais surtout pas que mes propos soient pris comme l'amorce d'un quelconque dogmatisme académique,l'expérience montre combien ces choses-là sont stérilisantes.Je voulais simplement souligner l'importance des outils fondamentaux,et en même temps rappeler combien leur étude a été négligée jusqu'à ce jour.

Voila donc brossé un tableau de la situation à l'heure actuelle et il suffit que je fasse pivoter tant soit peu mon fauteuil pour que m'apparaisse par la fenêtre un autre tableau,tout aussi agréable: celui des vacances imminentes avec son cortège de filles toutes plus ravissantes les unes que les autres.Dieu que la mode est plaisante cette année! à l'heure des restrictions,cette économie de tissus me ravit.

REMARQUES SUR DEUX TYPES DE MODELES EN THEORIE DES AUTOMATES

(J.Ph.Lehmann)

C.R.:3.80 5.22 5.32 6.1 6.31

Résumé: Soient les modèles suivants d'automates d'états finis:

I $R(t+1) = F(Q(t),S(t))$
$Q(t+1) = G(Q(t);S(t))$

II $R(t) = F(Q(t),S(t))$
$Q(t+1) = G(Q(t),S(t))$

- Dans une première partie, on montre comment il est possible dans les machines réelles, de donner du sens aux hypothèses des modèles, spécialement aux niveaux de la digitalisation de l'information, et de la discrétisation du temps. La description des automates devient plus précise et plus complète.

- Dans une seconde partie, on compare les modèles, particulièrement en étudiant comment, dans chacun, se pose la question de l'assemblage des automates. Les règles permettant de définir de "bons" assemblages, sont moins strictes dans I que dans II; l'unique raison tient à ce que, II serre de plus près la réalité(d'une certaine façon).
  Eu égard aux remarques de la 1ère partie, on montre que la question de la réalisation effective des machines se pose en fait, dans les mêmes termes pour I et II.

- Accéssoirement on met en oeuvre sur des exemples les algorithmes mis en évidence au cours de l'analyse du problème.

REMARQUES SUR DEUX TYPES DE MODELES EN THEORIE DES AUTOMATES

Nous nous préoccupons ici uniquement d'automates à nombre d'états fini(AEF).
les modes de représentation de tels objets, peuvent être très variés:

- On trouve couramment des doubles tableaux, indiquant pour un état et une information donnés à une époque déterminée, l'information résultante et l'état pris par l'automate à l'époque suivante.

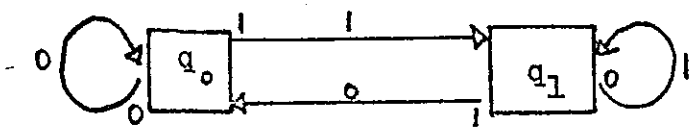
Ainsi un compteur de parité se présente de cette manière:

	q_0	q_1
0	0	1
1	1	0

	q_0	q_1
0	q_0	q_1
1	q_1	q_0

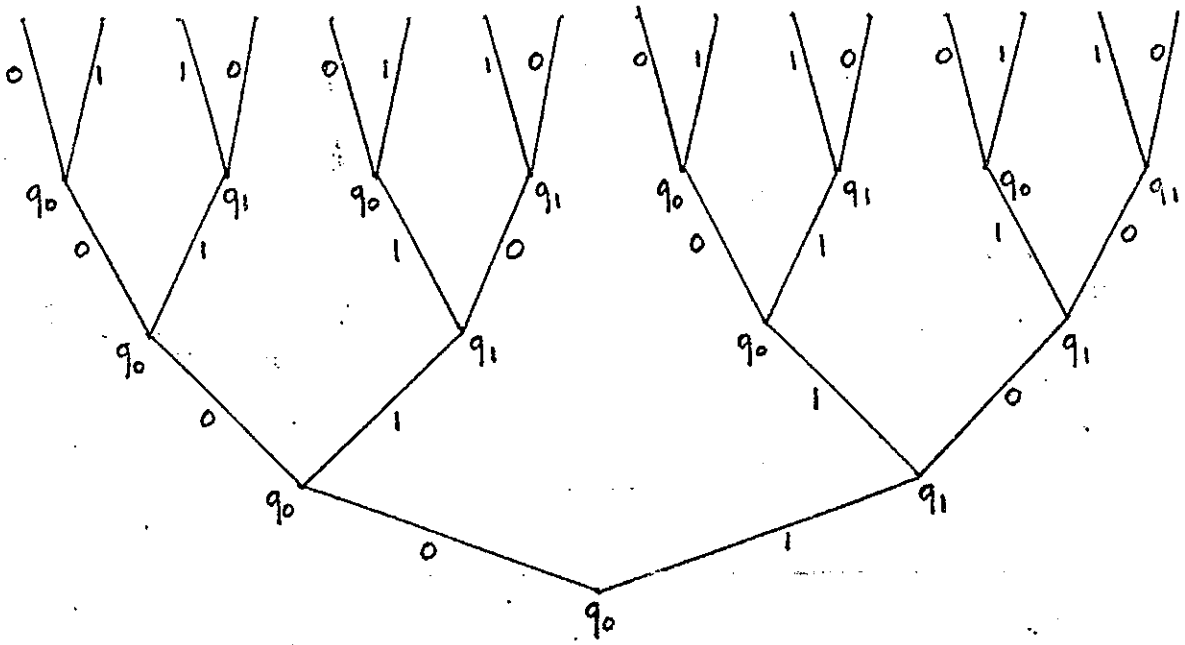
- Les représentations se font également sous forme de graphes: les états sont symbolisés par des carrés, les transitions d'un état vers un autre, par des flèches partant d'un carré et aboutissant à un autre; à l'origine de ces flèches, on indique l'information communiquée à l'automate, au milieu de celles-ci, l'information résultante.

L'automate précédent se présente alors ainsi:



- On peut aussi utiliser des arbres: à partir d'un noeud initial,naissent autant de branches(dans un ordre conventionnellement fixé) qu'il y a d'informations pouvant être communiquées à l'automate; sur chacune de ces branches, on indique l'information résultante. A partir des nouveaux noeuds ainsi déterminés, on réitère la même opération pour l'époque suivante, et ainsi de suite, on charge progréssivement l'arbre

L'automate précédent se présente alors ainsi:



- Un langage logique, convenablement adapté, que nous appellerons, langage de la logique propositionnelle temporelle(éventuellement avec prédicats), permet également de décrire les AEF.

On peut par exemple, représenter l'automate précédent ainsi:

$$z(t) = \left\{\left[x(t) \oplus z(t-1)\right] \wedge (t>1)\right\} \vee \left[x(t) \wedge (t=1)\right]$$

écriture dans laquelle, x(t) représente l'information entrant dans l'automate à l'instant t, z(t) l'information résultante (ce qui suppose un traitement instantané), et les symboles $\oplus$, $\wedge$, $\vee$, respectivement, le OU exclusif logique, la conjonction et la disjonction logiques. A noter qu'il n'y a pas à se préoccuper de la valeur de z(0), la partie gauche de la formule étant forcément égale à 0, pour t=1 début du processus.

Précisons à titre de remarque, que dans ce cas particulier, l'utilisation des quantificateurs n'a pas été nécéssaire, bien que dans de nombreux cas, il faille faire appel à eux.

La diversité des langages de description des AEF, pose de nombreux problèmes, l'un des plus importants résidant dans la compilation de chacun de ces langages dans les autres, et ceci d'autant plus que les descriptions précédentes ne forment pas l'ensemble de celles qui sont possibles.
La maîtrise de ces différents points de vue, ne peut que contribuer favorablement à la compréhension des objets étudiés, mais, et c'est sur ce point que nous voulons attirer l'attention, certains aspects de la théorie, si on se limite aux modes d'exposés dont nous avons donné quelques exemples, ne peuvent être correctement exposés et développés.

Ces problèmes se manifestent principalement autour des questions suivantes:

- Qu'est-ce qu'un état de l'automate?
- Qu'est-ce qu'une information communiquée à l'automate ou émise par lui?
- Comment décrire éfficacement le fonctionnement séquentiel(dans le temps) de l'automate?

Vis à vis de ce type de préoccupations, deux genres d'objections viennent à l'esprit, auxquelles nous voulons répondre d'emblée.

En premier lieu, on peut faire observer que le mérite d'une théorie, ne doit s'apprécier qu'une fois exercé sur son objet, l'éffort d'abstraction et de formalisation. Qu'il n'y a qu'une occasion de perdre son temps, à vouloir juger, à priori, de l'adéquation des concepts élémentaires à l'objet donné, et que ce n'est qu'à travers les développements de la théorie que se révèleront ses avantages et ses inconvénients éventuels.
On peut même préciser que cet effort de formalisation permet d'atteindre des résultats, que l'intuition à elle seule, eut été incapable de révéler. Dès lors, on découvrira après coup, le sens qu'il faut accorder à ces notions d'état, d'information, de fonctionnement séquentiel, y compris dans ce qu'elles pouvaient avoir d'insoupçonnable.

Pour ce qui nous concerne, nous ferons d'abord remarquer, qu'une telle attitude, sur le plan de la transmission des connaissances a fait des ravages; la formalisation et l'abstraction maximum apparaissent toujours comme un aboutissement qui vient couronner une période préalable, d'hésitations et d'approximations succéssives; si nécéssaire que soit l'économie de ce cheminement progréssif, il est douteux qu'elle puisse consister en la simple désignation des derniers modèles élaborés, sans fournir une réflexion suffisante sur la façon dont ils se sont graduellement structurés.

Si on nous accorde ce point, l'objection initiale peut toutefois se maintenir, en s'affinant, en se confinant au seul public des spécialistes: ceux-là sont supposés évoluer dans la théorie avec l'aisance requise, et les précautions évoquées ci-dessus n'ont plus de justification. Bien que nous estimions que même à ce niveau, nos critiques précédentes trouvent à s'appliquer, notamment en constatant à quel point les spécialistes d'une question sont le plus souvent ceux d'<u>une</u> théorie de la question, nous développerons notre second ensemble de remarques dans une autre direction.

Dans l'étude des automates, il y a un aspect sur lequel il nous parait indispensable d'insister: les dispositifs qu'on décrit et qu'on analyse, ont vocation à être matérialisés. Partant d'un schéma fonctionnel abstrait, il faut pouvoir aboutir en réalisant la structure concrète correspondante. Habituellement, les choses se présentent différamment; une théorie physique, par exemple, construit une image abstraite, plus ou moins formalisée d'une réalité-objet. Dans le cas qui nous occupe, la situation est complètement renversée; les concepts forment le point de départ, information, état, traitement.

A l'arrivée, par contre, il faut pouvoir trouver un dispositif concret, sur lequel, s'exercent des actions appropriées, représentatives(comment?) des informations communiquées, et qui délivre en retour d'autres actions représentatives de l'information émise, en fonctionnant conformément au schéma abstrait qu'il matérialise, et ce en faisant évoluer sa structure(comment?) au fur à mesure que s'appliquent sur lui les actions.

Lorsqu'on limite le sujet de l'étude aux automates abstraits, ces considérations, perdent une partie seulement de leur intérêt. Il est clair en effet, que même les résultats obtenus dans un tel cadre, ne trouveront pleinement leur sens, que dans la mesure où la question de la réalisation éffective des automates considérés aura été convenablement traitée.

La théorie qui nous occupe ici, est donc pour une partie éssentielle, celle de la matérialisation des automates abstraits.

De ce fait il n'est plus possible de se contenter, de tenir pour établi de façon externe à la théorie, ce que c'est qu'un état de la machine, une information, une époque. Sur tous ces points, il faut accorder l'attention nécéssaire, exercer l'éffort d'abstraction et de description qui s'impose , autrement dit, les intégrer dans la théorie.

Ces préoccupations se rencontrent à des degrés divers chez de nombreux auteurs. Certains n'y font aucune allusion, et n'évoquent que des automates abstraits. D'autres les abordent plus ou moins explicitement, en étudiant l'aspect structurel des automates, car alors la nécéssité de présenter les machines comme assemblages d'autres machines plus simples, pose directement ou indirectement ces questions(M.Minsky, Finite and infinite machines).Il arrive enfin que la réalisation des machines soit explicitement traitée(du moins en partie), et même que des développements importants soient consacrés aux dispositifs physiques intervenant dans leur construction(Kobrinskii, Trakhtenbrot, Theory of finite automata).

Nous allons plus précisément considérer deux types de modèles, que nous noterons I et II.

I
$$R(t+1) = F(Q(t),S(t))$$
$$Q(t+1) = G(Q(t),S(t))$$

II
$$R(t) = F(Q(t),S(t))$$
$$Q(t+1) = G(Q(t),S(t))$$

Le modèle I est utilisé notamment par Minsky, dans l'ouvrage référencé ci-dessus, et le modèle II par Trakhtenbrot et Kobrinskii.

Dans ces deux modèles, le processus de fonctionnement est discrétisé dans le temps, et l'information est supposée digitale. Nous montrerons comment dans les machines réelles, il est possible de donner du sens à ces hypothèses, et nous indiquerons ce qu'il faut adjoindre aux modèles, et comment les modifier, pour que la description de l'automate soit plus précise et plus complète.

Dans une seconde partie, nous comparerons ces modèles, en démontrant:

1) Dans le modèle I, tout assemblage d'automates définit un automate, en respectant seulement les règles suivantes: les canaux de sortie ne doivent jamais être réunis entre eux; un canal de sortie peut être relié à un ou plusieurs canaux d'entrée, sinon il constitue une sortie du système total; il existe au moins un canal d'entrée dans le système total;il existe au moins un canal de sortie dans le système total.

2) Dans le modèle I, il n'est possible d'assurer de cohérence sur le plan sémantique, qu'en supposant que les informations internes au système sont synchroniques avec les informations externes qui lui parviennent. Si cette hypothèse n'est pas garantie, le système peut toutefois aboutir à des états stables selon les circonstances.

3) Dans le modèle II, l'hypothèse de la propagation instantanée($R(t) = F(Q(t),S(t))$), interdit à elseule d'envisager n'importe quel assemblage d'automates, et impose la nécéssité de définir des assemblages correctement organisés.

4) Le problème de la réalisation effective des machines décrites dans les modèles I et II, se résout de la même façon pour l'un et l'autre, en analysant les phénomènes qui doivent intervenir, dans la partie implicite de ces modèles, savoir l'époque allant de t à t+1. La nécéssité de réaliser des mémoires s'impose alors naturellement, plus précisément des mémoires permanentes, si l'on souhaite un fonctionnement asynchrone.

I. DISCRETISATION DU TEMPS.DIGITALISATION DE L'INFORMATION

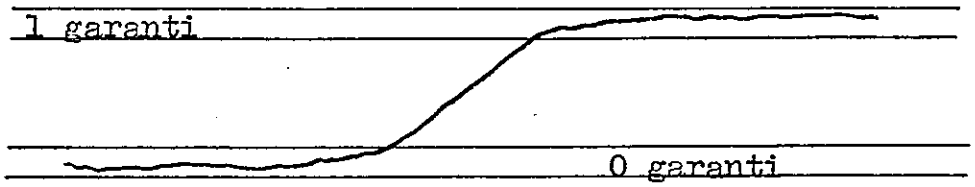
L'essentiel des remarques qui suivent, consiste à mettre en évidence, que du point de vue physique, il n'y a pas moyen d'envisager ces deux questions séparément, qu'en fait elles se conditionnent mutuellement.

Commençons par quelques remarques sur la notion d'information. Considérons par exemple, la machine levier, en supposant qu'il est inutile de décrire son protocole d'utilisation, et en admettant que chacun en a une connaissance suffisante. Pour un tel dispositif, les actions exercées, et de même les effets produits, sont regardés pour ce qu'ils sont, savoir des forces: l'énergie mécanique est elle-même l'objet que l'on désire soumettre à la transformation, en vue d'obtenir les effets appropriés. Faisant alors référence au schéma logique que cet objet matérialise, et à toute la théorie qui le sous-tend., et bien que cet objet n'ait pas été conçu à cet effet, nous sommes libres de le considérer comme une forme particulière de machine à multiplier; en entrée, on présente des nombres(sous la forme requise), et suivant la position du point d'appui, la longueur de la barre, on obtient en sortie, le résultat de la multiplication de ces nombres par un certain facteur K. Cette dualité est présente dans toutes les machines: notre attention peut se fixer, sur la nature, le niveau, l'intensité des entrées ou des sorties, ou bien ne les considérer que pour ce qu'elles représentent, non pour ce qu'elles sont.
La particularité des machines que nous souhaitons réaliser, ne tient donc qu'à l'usage qu'on en fera; dès que les actions exercées ou résultantes sont associées à des sens, dès que le statut de symbole leur est applicable, la machine est spécifiée dans la classe considérée. D'un autre coté, ces signes, ces symboles,ont forcément un mode physique d'existence. Difrents d'une technologie à l'autre, ces modes déterminent dans chacune néammoins,des contraintes communes ou similaires.

A l'ensemble fini de signes communiqués ou émis, doit correspondre un ensemble d'actions différentes et parfaitement définies. Il est clair que ce qui rend une action différente d'une autre, ne peut tenir qu'aux effets distincts que l'une et l'autre déterminent pour un récepteur donné. Or quel que soit le récepteur envisagé, aussi grandes soient ses possibilités de discernement, elles trouvent toujours(physiquement) une limite dans une certaine frange d'incertitude.
Ainsi par exemple, une porte logique fonctionne conformément à son schéma à la condition que les tensions exercées aux entrées, soient garanties dans une certaine plage, pour celle représentative du 0, dans une autre plage parfaitement disjointe de la précédente, pour celle représentative du 1. En retour, elle délivrera des tensions, elles aussi garanties.

Préoccupons nous maintenant de la façon dont ces actions vont devoir s'exercer; pour développer nos réflexions nous nous appuyons toujours sur l'exemple d'un dispositif électrique, que le lecteur n'aurait aucun mal à transposer.

Le canal par lequel s'exerce les actions sur la machine, doit nécéssairement pouvoir prendre plusieurs états. Ainsi le fil va tantôt être porté à la tension 0, tantôt à la tension 1. Notons à ce propos que l'exercice d'une action invariable, interdit toute apparition d'information, puisque précisément celle-ci ne peut surgir qu'à l'occasion de procéssus de différenciation. Ainsi le plus simple des boutons-poussoirs, n'est en mesure d'engendrer de l'information, qu'à cause des deux positions possibles dans lesquelles il peut se trouver, et des effets distincts associés à ces deux états. Il est souvent commode de présenter l'état des canaux reliés à un automate en utilisant une terminologie du genre "tout" ou "rien","actif" ou"en repos", mais celles-ci se révèlent souvent dangereuses à cause de l'ambiguité qu'elles entretiennent entre les états physiques du canal et le sens qu'il faut accorder à ces états. Ainsi il n'est pas correct de laisser sous-entendre qu'il ne se passe"rien"dans un circuit lorsque l'interrupteur est ouvert, et "quelque chose", lorsqu'il est fermé. De fait le canal est toujours dans un certain état, et le récepteur auquel il est connecté, toujours sous l'influence d'un des états possibles. Le fait que, dans la réalité, le récepteur est en permanence soumis à des actions, que donc, constamment l'information lui parvient, doit être rapproché d'un autre: dans tous les dispositifs réels, le passage d'un état à un autre ne peut pas se faire instantanément; quelque soit la technologie mise en oeuvre, le temps de commutation ne sera jamais nul; ainsi la transition entre deux états de voltage garantis se présentera de cette façon:



Partant des modèles I et II, et se posant la question de la réalisation effective des machines qu'ils sont supposés décrire, il faut prendre en compte les contraintes inévitables, que nous avons signalées plus haut, en résolvant les contradictions qu'elles semblent introduire: actions en réalité permanentes, et dans les modèles,discrètes, processus de transition en réalité continus et dans les modèles,discontinus.
Là encore, la solution ne saurait être recherchée en tentant la construction de machines d'un nouveau genre, résistant aux principes physiques les mieux établis.
Si l'on observe toutes les situations, où l'information joue un rôle, on constate, aussi bien au niveau de la transmission, qu'à celui du traitement et de l'émission, que ce qui est essentiel, c'est la durée: écrire un symbole au tableau noir, requiert un certain temps, et ce n'est que lorsque ce temps s'est écoulé, qu'on s'autorise alors à prendre connaissance de l'information. De même en tournant la manivelle du machine à calculer manuelle, et constatant dans l'afficheur, le déroulement des chiffres, on ne se décide à lire le résultat qu'après que la manivelle ait buté, ou qu'une sonnerie ad hoc, ait retenti. C'est dire tout simplement que l'information n'est prise en compte qu'à certaines époques, elles-mêmes demandant

à être signalées, et constituant de ce fait, des informations particulières. La question posée par la variation d'un état à un autre se résout identiquement: puisque les transitions sont inévitables, on ne s'obstinera pas à essayer de les supprimer, et on fermera les yeux pendant qu'elles se déroulent. Ainsi les opérations de discrétisation se révèlent logiquement liées à celles de digitalisation, et correspondent uniquement à la manière dont le fonctionnement de la machine est observé.

Un dernier aspect des choses demande à être abordé, celui de l'enchainement et de la coordination des différentes opérations en cause: ces questions, bien qu'implicites dans les modèles, sont en réalité d'une grande importance; il faut d'une part envisager, les époques durant lesquelles les actions représentatives d'informations, auront à s'exercer, d'autre part les "temps morts" séparant chaque action de la suivante, en veillant à ce que, durant ces périodes, les résultats des traitements précédents qui peuvent influer sur les sorties futures, (Q(t+1)), ne soient en aucune façon perturbés. Ceci mérite d'autant plus l'attention, qu'il faut se souvenir que la machine est en permanence soumise à des actions, et que les temps morts idéaux du modèle, ont dans la réalité une contrepartie éminement active, siège de plusieurs transformations, dont nous allons donner une des descriptions possibles.
Décomposons l'ensemble de ces actions en différentes phases:

- Phase 1: exercice de l'action représentative d'information.

  Au sein de cette phase, deux périodes sont à considérer: la première pendant laquelle on attend que le signal soit garanti dans une plage donnée, la seconde pendant laquelle l'action sera maintenue au niveau garanti, le temps nécéssaire au bon déroulement de la phase 2.

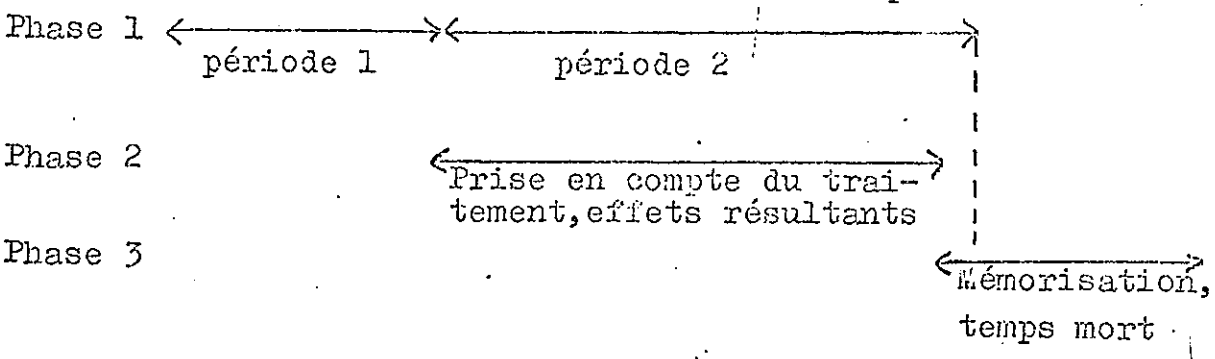
- Phase 2: traitement de l'information.

  Ce n'est que pendant cette phase, que l'action exercée est prise en compte, parce qu'elle est alors garantie. La période 2 de la phase 1 doit durer le temps nécéssaire à ce que le traitement soit accompli, et les actions résultantes garanties de la mêmanière qu'avaient du l'être, les actions appliquées.

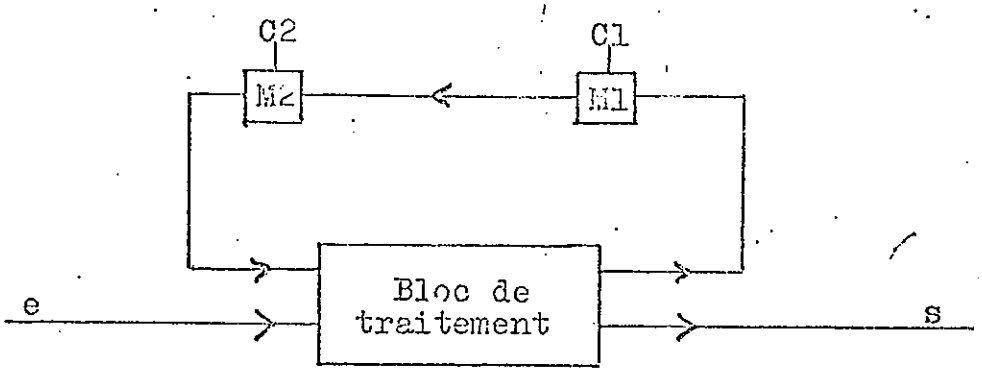
- Phase 3: le "temps mort".

  La particularité des machines qu'on étudie, tient à ce que les résultats du traitement à un moment donné, doivent pouvoir conditionner ultérieurement le traitement à venir, notamment si on ne veut pas se limiter au seul cas des automates à un état. De ce fait les actions résultantes, doivent à l'époque suivante, pouvoir être exercées sur la machine, en même temps que celles provenant de l'extérieur du système et suivant le même protocole qu'on a décrit: il est donc indispensable, à la fois de les mémoriser et de les synchroniser avec celles-ci.

Schématiquement, les choses peuvent ainsi se présenter:



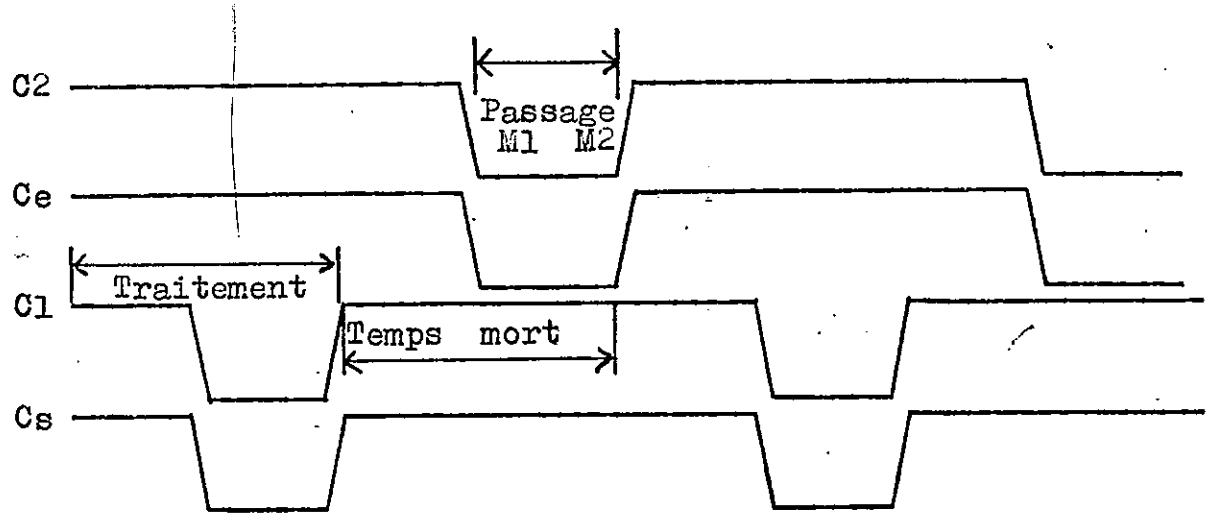
Indiquons maintenant un type de solution possible, réalisable en technologie électronique.



Dans le schéma précédent, les organes M1 et M2 sont des latchs. Ils fonctionnent de la manière suivante: lorsque la commande C est à 0, ils sont transparents, c'est à dire que l'état des fils de sortie reflète celui des fils d'entrée. Quand la commande passe à 1, la dernière information présente en entrée au moment de la transition est mémorisée et émise en permanence en sortie.
Le bloc de traitement représente ici un automate quelconque. Parmi les résultats du traitement, certains doivent être recueillis pour être fournis au système, en entrée à l'époque suivante. Ceci est indiqué dans le déssin, par un fil sortant du bloc de traitement, connecté à M1, elle-même connectée à M2, elle-même se présentant en entrée du bloc.

A l'initialisation C2 est à 1, et M2 emet en permanence une information stable, que rien ne peut perturber, tant que C2 est maintenue à 1.
Notons bien, que des actions s'exercent sans cesse à l'entrée du bloc de traitement, celle définie par M2 étant constante. On présente alors par l'entrée e, suivant le protocole qui a été décrit plus haut, l'information de l'époque t; à cet effet, on peut par exemple imaginer qu'en amont de l'entrée e, se trouve un organe du type M1 ou M2.
En aval du bloc de traitement, on disposera alors du résu tat s, et d'une information à destination de M1; pendant que ces informations sont valides(tenues au niveau requis), C1 est mise à 0, puis à 1, C2 étant toujours à 1. La phase 2 est alors terminée, les résultats du calcul disponibles à la fois sur s et dans M1, pour la phase 3.

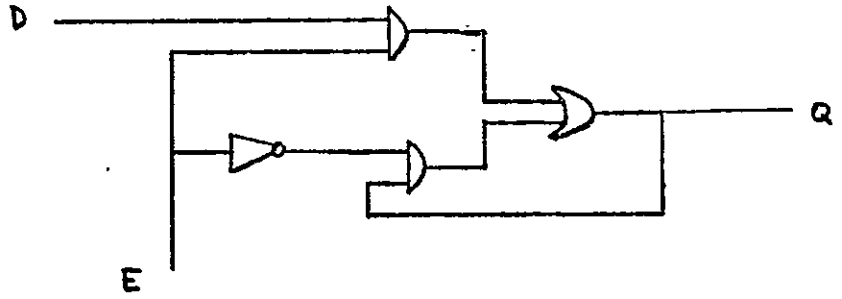
Cette dernière phase, dite du "temps mort", va consister simplement à opérer le transfert dans M2, de l'information mémorisée dans M1: pendant que C1 est à 1, C2 est mise à 0, le temps nécéssaire, puis remontée à 1, C1 étant toujours maintenue à 1.
Tout est alors prêt, pour le cycle suivant, qui se déroulera très exactement comme celui qui vient d'être décrit.
Si l'on suppose, que l'information qui transite par le canal e, est délivrée par un organe Me, de commande Ce, et que de même sur s il y a un organe Ms, de commande Cs, le schéma du fonctionnement dans le temps peut se présenter ainsi:



Ainsi il apparait que le fonctionnement de nos machines, ne peut réellement s'envisager, qu'à travers une gestion soignée de commandes du genre de C1 et C2. Parcailleurs la génération de ces commandes, ne pose rien d'autre que la problème de la construction d'un nouvel automate, dont la fonction est de contrôler celui qui effectue le traitement, en définissant la synchronisation nécéssaire, et en fournissant l'énergie, la force motrice indispensable à l'enchainement des opérations. C'est la manivelle du système.

Pour terminer cette partie, nous indiquerons comment peut se réaliser un montage du type latch; ceci nous donnera l'occasion d'appliquer, la méthode d'analyse qui a été exposée dans l'article paru dans le numéro précédent de ce bulletin, sous le titre "quelques remarques sur un système séquentiel".
De plus le montage qu'on va considérer, étant formé d'un assemblage d'automates, il nous permettra d'aborder naturellement la seconde partie de notre exposé.

Considérons l'assemblage:



L'analyse se fait de façon dynamique: pour chaque configuration fixe de D et E, on envisage pour Q, l'ensemble des hypothèses possibles. Il est clair, en effet qu'aucune ne peut à priori être écartée, puisqu'au moment où le système sera sollicité, rien ne permet de savoir quel est l'état du fil Q. La seule question qui se pose alors est celle de l'éventualité que le montage évolue vers un état stable.

Fixons E à 1.

Si D est à 1 et maintenu, supposons que l'entrée Q soit interprétée(ce qu'elle sera forcément), à l'époque t, comme 0; on voit qu'à l'époque t+1, la sortie Q sera alors mise à 1, et ce, de façon stable.
Ce régime stable aurait été atteint d'emblée si au départ Q avait été interprété comme 1.

Si D est à 0 et maintenu, on constate sans mal qu'un état stable est défini avec Q=0.

Au total, on est assuré, lorsque E est à 1, qu'au bout d'un certain temps, Q et D ont la même valeur.

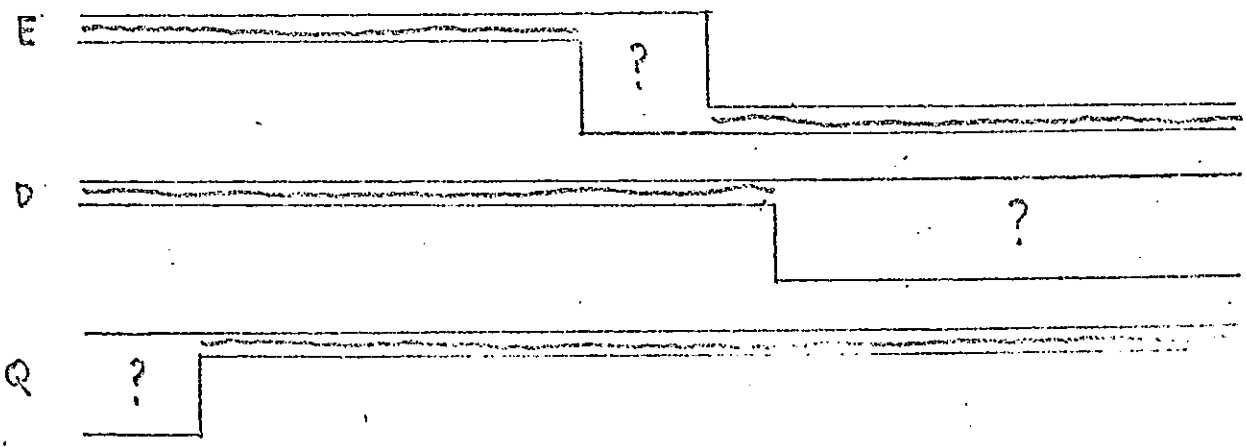
Supposons maintenant que E tombe à 0. La valeur de D, devient sans importance, et à partir du moment où le système interprète E comme 0, la dernière valeur présente sur Q est mémorisée.
Le mode d'utilisation du système se définit donc ainsi:

Tenir D au niveau garanti requis, représentatif suivant le cas d'un 1 ou d'un 0, et tenir E à 1, le temps que Q ait pris la même valeur que D,

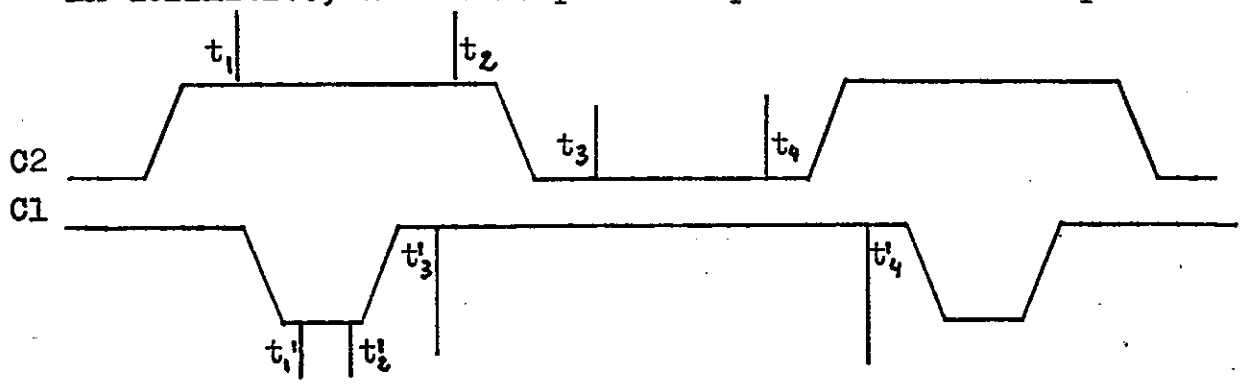
puis,

alors que D est maintenu, faire tomber E à 0. Aussi complexe que soit cette transition, au cours de laquelle on peut même imaginer que l'état 1 soit éventuellement repris, on est assuré qu'au bout d'un temps fini, la valeur 0 est garantie, et qu'alors le système émet en permanence la dernière valeur de Q. Puisque, tout au long de cette transition, cette valeur a été maintenue constante, la validité de l'information obtenue est alors garantie.

Le schéma du fonctionnement dans le temps peut se présenter ainsi: (on a supposé que la donnée à mémoriser D était à 1.



En définitive, un modèle possible pourra être ainsi présenté:



On peut admettre que la période de C1 et C2 est la même, T.

$$\exists T\ \forall \lambda \in N \qquad \begin{array}{ll} C2(t+\lambda T) = C2(t) & \forall t \\ C1(t+\lambda T) = C1(t) & \forall t \end{array}$$

Considérons alors la propriété suivante:

$$\exists t \exists t',\ t < t',\ \forall \lambda \in N, \forall \tau \in [t+\lambda T, t'+\lambda T] \qquad C2(\tau) = 1$$

t_1 est alors le inf des t vérifiant la propriété, et t_2 est le sup de ces t.
De même sont définis t_3 et t_4 pour la plage $C2(\tau) = 0$.

De même sont définis t'_1 et t'_2 pour $C1(\tau) = 0$, et t'_3, t'_4 pour $C1(\tau) = 1$.

La fonction réponse est alors garantie constante et valide au moins dans l'espace $[t'_2 \quad t_2]$, on limitera l'ensemble de définition à cet espace.

La fonction, nouvel état(Q(t+1)), sera garantie dans les mêmes conditions, pour l'espace de définition $[t_1 + T,\ t_2 + T]$.

La fonction, entrée, aura par exemple pour espace $[t'_1 \quad t'_2]$.

Partout ailleurs, les fonctions précitées, seront considérées, comme non définies.

Quelle que soit la technologie mise en oeuvre, les problèmes se rencontreront, et les différences qu'on constatera avec le modèle proposé, ne seront jamais essentielles. Dans tous les cas, la réalisation de l'automate passe par la mise en place d'un automate supplémentaire, contrôlant le précédent par la définition des plages temporelles où l'information est validée, et par la mémorisation des résultats intermédiaires du calcul durant les phases "inactives" (de non-validation).

II DEUXIEME PARTIE

1) ASSEMBLAGES D'AUTOMATES DANS LE MODELE I

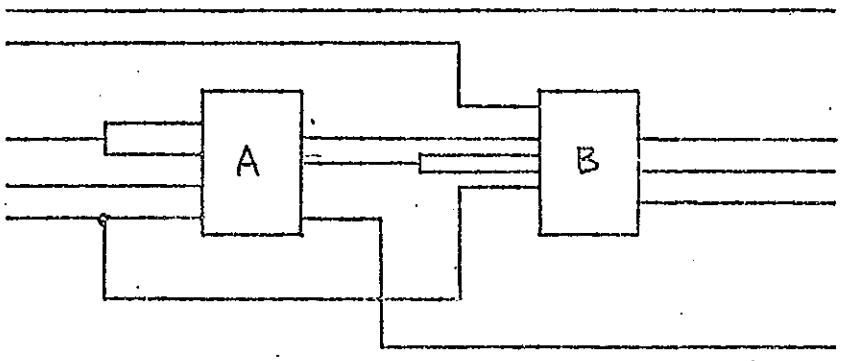
$$R(t+1) = F(Q(t),S(t))$$
$$Q(t+1) = G(Q(t),S(t))$$

Rappelons les seules règles que nous observerons :

- Doivent exister au moins un canal d'entrée et un canal de sortie.
- un canal de sortie peut être relié à un ou plusieurs canaux d'entrée, mais jamais relié à un autre canal de sortie.
- les canaux d'entrée d'un automate qui ne sont pas reliés à des canaux de sortie d'un autre* automate, sont des entrées de l'assemblage; les canaux de sortie d'un automate qui ne sont pas reliés à des canaux d'entrée d'un automate sont des sorties de l'assemblage
- La propagation le long des canaux est supposée instantanée.

a) Assemblages sans boucles

Dans un premier temps, considérons l'assemblage suivant:



On observera que dans le cas présent, aucune boucle n'existe dans le circuit. Par ailleurs, on constate sans mal, que les règles de bon assemblage décrites ci-dessus, ont été respectées.

A un instant t donné, sur chacun des canaux de l'assemblage, est présente une information; d'autre part chacun des automates est dans un état spécifié. Dès lors, il suffit de consulter la table de chacun d'eux, pour déterminer les sorties de l'époque suivante et les nouveaux états pris.
Les simples remarques qui précèdent, si on adopte la représentation des AEF en termes d' arbres, indiquent clairement , que la description de l'assemblage pourra également se faire au moyen d'un arbre. Cependant il est peut-être moins clair, même si l'intuition le laisse deviner, que l'arbre ainsi obtenu sera à son tour formé avec un nombre fini de sous-arbres distincts.

* éventuellement le même.

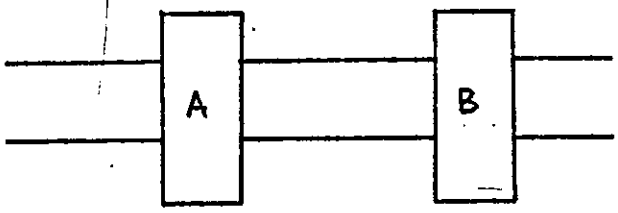
Nous démontrerons ce fait sur un exemple, en utilisant la représentation sous forme de tableaux. La démonstration générale, ne présenterait aucune difficulté de plus que celle que nous indiquons, si ce n'est au niveau de la notation.

Soient les deux automates A et B définis par les tableaux:

$e_1 e_2$ \ $s_1 s_2$	q_0	q_1
00	$01/q_0$	$10/q_0$
01	$00/q_1$	$11/q_0$
10	$00/q_1$	$01/q_1$
11	$11/q_1$	$11/q_0$

	r_0	r
00	$01/r_0$	$11/r_1$
01	$00/r_0$	$01/r_1$
10	$10/r_1$	$00/r_1$
11	$11/r_0$	$10/r_0$

On considère l'assemblage:



En toute logique, une analyse de compatibilité s'impose pour initialiser correctement le système. Le modèle nous oblige en effet à indiquer, non seulement les états initiaux de A et de B, mais aussi l'information émise par A au début du processus. Or ces sorties de A ne sont pas forcément compatibles avec tous les états de A. Dans l'exemple traité, il est clair que q_0 et 00, ainsi que q_1 et 10, sont à exclure, puisqu'aucun passé de l'automate n'a pu déterminer cette situation..

On dénombre alors pour deux états de A possibles, trois informations émises possibles, elles-mêmes associées à deux états possibles de B, soit au total 2x3x2=12 états.

Le tableau de la page suivante est celui de l'automate assemblé.

Indiquons comment il a été construit:

- chaque colonne indiquant les sorties contient un seul symbole; en effet la sortie est déterminée par l'état de B et la valeur présente sur le canal intermédiaire, et ceci indépendamment de l'entrée de A.
- Le tableau des états qui sont notés sous la forme $q_i XY q_j$, est formé ainsi: pour q_i et XY fixés, on lit dans le tableau de A, le nouveau q_i et le nouveau couple XY. Le symbole r_j est le même dans toutes les colonnes, car lui aussi est indépendant de l'entrée de A.

En posant:

$q_0 01\ r_0 = Q_1$, $q_0 10\ r_1 = Q_4$, $q_1 00\ r_1 = Q_7$, $q_1 01\ r_1 = Q_{10}$,

$q_0 01\ r_1 = Q_2$, $q_0 11\ r_0 = Q_5$, $q_1 00\ r_1 = Q_8$, $q_1 11\ r_0 = Q_{11}$,

$q_0 10\ r_0 = Q_3$, $q_0 11\ r_1 = Q_6$, $q_1 01\ r_0 = Q_9$, $q_1 11\ r_1 = Q_{12}$,

on obtient alors le tableau:

	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	Q_9	Q_{10}	Q_{11}	Q_{12}
00	00	01	10	00	11	10	01	11	00	01	11	10
01	00	01	10	00	11	10	01	11	00	01	11	10
10	00	01	10	00	11	10	01	11	00	01	11	10
11	00	01	10	00	11	10	01	11	00	01	11	10
00	Q_1	Q_2	Q_2	Q_2	Q_1	Q_1	Q_3	Q_4	Q_3	Q_4	Q_3	Q_3
01	Q_7	Q_8	Q_8	Q_8	Q_7	Q_7	Q_5	Q_6	Q_5	Q_6	Q_5	Q_5
10	Q_7	Q_8	Q_8	Q_8	Q_7	Q_7	Q_9	Q_{10}	Q_9	Q_{10}	Q_9	Q_9
11	Q_{11}	Q_{12}	Q_{12}	Q_{12}	Q_{11}	Q_{11}	Q_5	Q_6	Q_5	Q_6	Q_5	Q_5

Une rapide étude de minimisation, montre en outre que les états ainsi déterminés sont tous distincts. On peut en effet constater que tout état est distinct de tout autre pour certaines séquences bien choisies de longueur maximum 2.

Les minimisations éventuelles qu'on peut opérer dans les tableaux construits selon la méthode qu'on a exposée, et de même les questions de compatibilité , ne peuvent être traitées qu'en fonction des cas particuliers rencontrés, mais nous pouvons donner, grâce à la méthode générale, une majoration du nombre des états de l'automate représentatif de l'assemblage.

Si K est le nombre des états de A, M le nombre de signes distincts pouvant être transmis à B, K' le nombre des états de B, alors le nombre des états de l'automate assemblage de A et B, sera inférieur ou égal à K.M.K' .

Notons à titre de remarque, que cette borne peut effectivement être atteinte: il suffit de coupler l'automate B avec lui-même, pour définir un assemblage à 16 états, tous distincts; on laissera la satisfaction au lecteur de s'en convaincre.

En guise de conclusion, nous pouvons donc énoncer maintenant, <u>que tout assemblage d'automates, respectant les règles annoncées, et ne comportant pas de boucle, définit un automate</u>.

Ceci est une conséquence immédiate du résultat établi dans le cas de deux automates, et généralisé par récurrence à un nombre <u>fini</u> quelconque d'AEF.

Soit un assemblage des n automates $A_1 A_2 \ldots A_n$, correctement défini et sans boucle.
$K_1, K_2, \ldots K_n$ sont les nombres des états des automates correspondants. $M_2\ M_3 \ldots M_n$ sont les nombres des signes pouvant être transmis aux automates correspondants.
L'assemblage de A_1 et A_2 définit un automate à nombre d'états maximum $K_1 M_2 K_2$ qui lui-même assemblé avec A_3, en définit un nouveau et ainsi de suite.

Au total le maximum des états est donné par la formule:

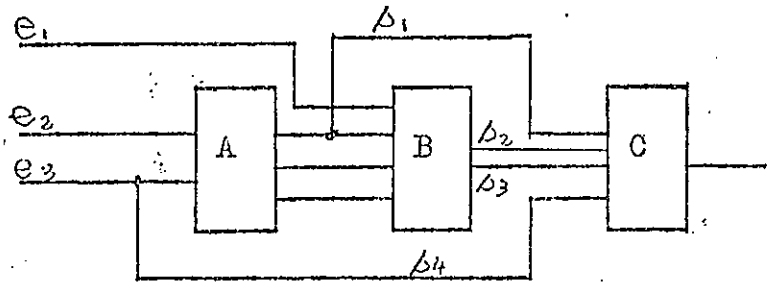
$$((\ldots((K_1 M_2 K_2) M_3 K_3) M_4 K_4) \ldots\ldots M_n K_n)$$

Soit $(\prod_{i=1}^{n} K_i)(\prod_{j=2}^{n} M_j) = P.$

<u>P est le nombre maximum des états de l'assemblage considéré.</u>

Avant d'étudier les assemblages avec boucles, faisons une remarque. On pourrait craindre qu'un certain type de cablage pourtant sans boucle et correctement formé, n'a pas été jusqu'ici envisagé.

Considérons par exemple l'assemblage suivant:



A la différence de l'exemple étudié précédemment, certaines entrées du système sont à la fois reliées à plusieurs automates, et de même certaines sorties d'automates constituent les entrées de plusieurs autres.
On peut en fait se ramener au cas déjà étudié, enprocédant de proche en proche de la façon suivante:

Les conditions d'initialisation se définissent exactement comme on l'a fait dans le cas précédent. Là encore une analyse de compatibilité s'impose, sur laquelle nous n'insisterons pas, puisqu'il n'y a rien de nouveau à signaler.

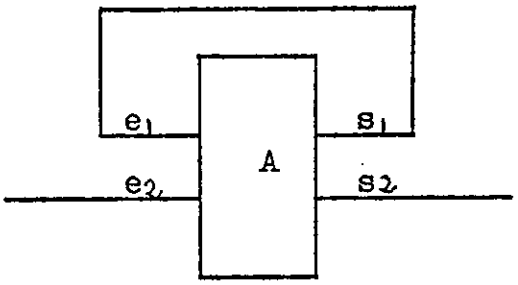
On considère alors l'assemblage formé par A et B, en spécifiant comme entrées de l'assemblage, tous les canaux provenant de l'extérieur et formant des entrées directes de A ou de B. De même on spécifie comme sorties de l'assemblage, tous les canaux qui ne constituent, ni des entrées de A, ni des entrées de B.
Dans le cas particulier e_1, e_2, e_3 sont les entrées du système. s_1, s_2, s_3, s_4 (qui est aussi e_3) sont les sorties du système.
En appliquant la méthode qui a été utilisée dans l'exemple précédent, on peut alors identifier l'assemblage de A et B, à un nouvel automate AEF. En fait la seule différence avec la situation déjà examinée réside dans la réduction du nombre des états à envisager: en effet alors que tout à l'heure, l'ensemble des configurations entrant dans B, devait à priori être retenu dans sa totalité, ici seul les canaux entrant dans B et qui ne sont pas des entrées du système devront être pris en compte pour la définition des états.
En définitive, un tel cablage, loin d'entrainer une complication de la machine, contribue en fait à réduire le nombre des états de l'assemblage.
Une fois A et B, assimilés à un AEF, noté AB, il ne reste qu'à réitérer la méthode qu'on vient d'exposer sur l'assemblage formé par AB et C.

On notera que la méthode en question a un caractère parfaitement algorithmique, et se met en oeuvre de façon automatique, sans aucune ambiguité.

Nous allons maintenant envisager les assemblages avec boucles.

b) Assemblages avec boucles

On reprend l'automate A, dont la table est donnée 4 pages plus haut, et on forme l'assemblage suivant:



Les caractéristiques du modèle, permettent d'envisager un tel montage sans aucune restriction: aucune contradiction logique n'apparait dans ce type de schéma, à la différence de ce qu'on constatera dans le modèle II.

Initialiser l'automate consiste à définir, comme on l'a fait précédemment, l'état des éléments internes, ici l'état de l'automate(sans tenir compte de la boucle) et l'information présente sur la boucle. Clairement, une analyse de compatibilité est encore nécéssaire: on doit en effet exclure les couples état-s_1, qui ne correspondent à aucun passé possible de la machine. Dans le cas particulier, tous les couples qu'on peut former sont possibles.
L'ensemble de ces couples forme l'ensemble desétats de l'assemblage: en effet, que ceux-ci doivent tous être envisagés, est manifeste, puisqu'ils forment les états initiaux possibles; et qu'ils soient les seuls provient de ce que, à tout instant, un état sera pris par A(sans tenir compte de la boucle), et une action s'exercera à l'entrée e_1.

Dans le cas présent, on obtient le tableau suivant:

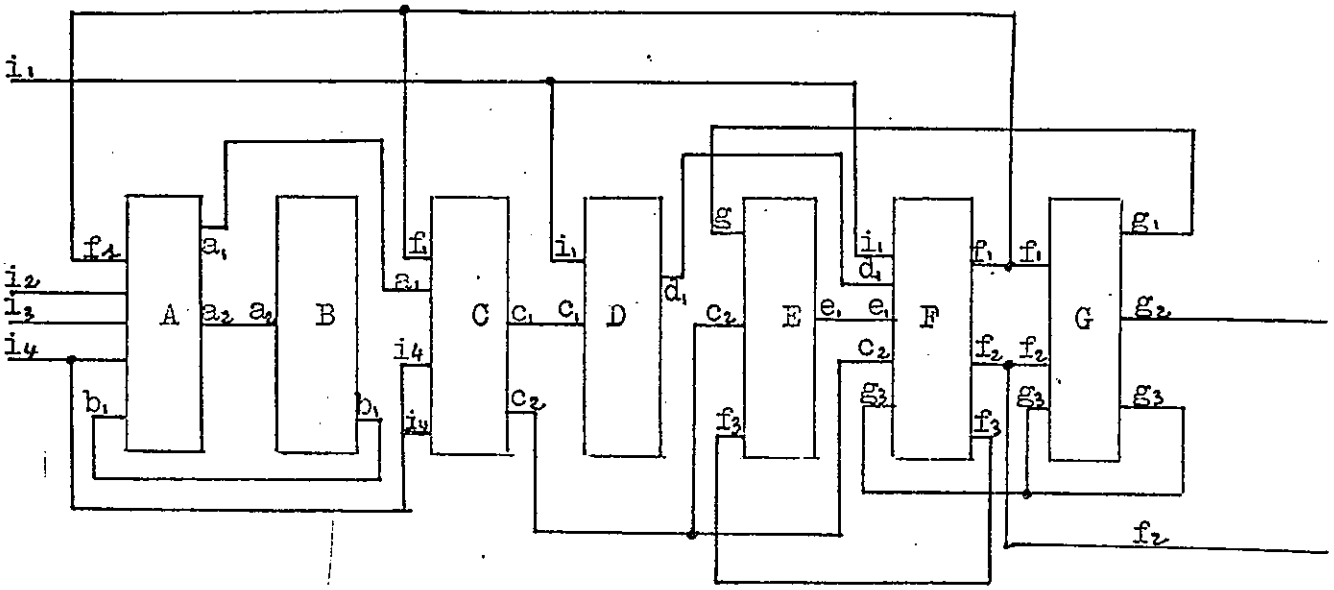
	$0\ q_0$	$1\ q_1$	$0\ q_1$	$1\ q_1$
0	$1/0q_0$	$0/0q_1$	$0/1q_0$	$1/0q_1$
1	$0/0q_1$	$1/1q_1$	$1/1q_0$	$1/1q_0$

De façon générale, on peut énoncer que, tout automate AEF dont certaines sorties sont prises pour entrées, définit un automate AEF, dont le nombre d'états est au maximum, le produit du nombre d'etats de l'automate avant le bouclage, par le nombre des configurations pouvant être présentes sur la boucle.

L'exemple précédent, nous montre de plus, que ce maximum peut être atteint.

Etudions maintenant des assemblages plus complexes.

Considérons l'assemblage suivant:



Les entrées de chaque automate, sont à sa gauche et ses sorties à droite. Un examen attentif du déssin, montre que tous les automates figurés, sont dépendants les uns des autres, en ce sens que tout automate détermine des sorties, qui influent sur les entrées de tout autre, éventuellement par l'intermédiaire d'autres automates.
On voit bien, que la complexité de l'assemblage augmentant, il sera de plus en plus difficile, de déterminer si les dispositifs, sont, oui ou non, dépendants les uns des autres. Nous indiquerons plus loin une méthode permettant, de résoudre systèmatiquement cette question.
Supposons la, pour l'instant résolue, et considérons les assemblages, formés d'automates mutuellement dépendants, comme c'est le cas ci-dessus.

Nous décrivons une procédure algorithmique, permettant de ramener, tout assemblage de ce type, à un seul automate déterminé.

-Préparatifs: on commence par nommer toutes les entrées et sorties de chaque automate, en tenant compte des liaisons, c'est à dire, en répétant le même nom, quand la sortie d'un automate, constitue l'entrée d'un autre.

-1a)On choisit alors n'importe quel automate de l'assemblage; il n'y a, en effet, aucun sens à rechercher certains qui seraient les premiers, ou les derniers, puisqu'en réalité, tous sont dépendants les uns des autres.
Les sorties dont la destination n'est pas unique, sont ramifiées en autant de branches que nécéssaire; lorsque plusieurs ramifications de même nom, aboutissent au même automate, on les confond et on remplace l'automate, par celui déterminé par cette restriction. On réitère le procédé jusqu'à ce que les noms des [illegible] soient tous distincts

-1b) S'il existe une sortie et une entrée de même nom, on remplace l'automate par celui défini par le bouclage correspondant, selon la méthode qu'on a exposée plus haut.
Si la sortie bouclée à l'entrée n'était pas ramifiée, le nouvel automate ne possède plus, ni cette entrée, ni cette sortie. Dans le cas contraire, le nouvel automate aura cette sortie ramifiée, en autant de branches que précédemment, moins une.
On recommence le procéssus, jusqu'à ce que chaque boucle ait été fermée. Il subsiste alors forcément des sorties(et des entrées), car sinon aucun automate ne serait dépendant de celui que nous avons choisi, ce qui est contraire à nos hypothèses.

-2a) Puis, on choisit n'importe quel automate(qui n'a pas été déjà choisi) dont la liste des noms des entrées, contient au moins un nom de sortie de l'automate précédent, et on lui applique le traitement décrit dans le paragraphe 1.

-2b) Lorsque plusieurs ramifications de sortie du premier, portent le même nom qu'une entrée du second, on relie l'une d'entre elles à cette entrée, et on laisse subsister les autres comme sorties.
Les entrées du nouveau système, sont alors formées de toutes celles du premier automate, et de celles du second qui ne sont pas des sorties du premier. Les sorties du nouveau système, sont formées de toutes celles du second automate, et de celles du premier qui ne sont pas des entrées du second.

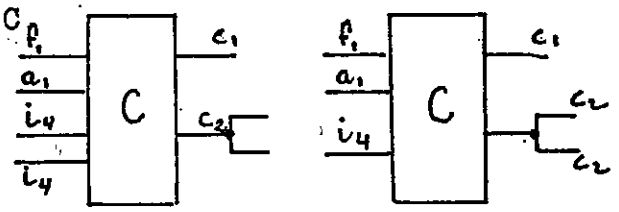
-2c) On remplace alors l'assemblage obtenu, par l'automate équivalent, en utilisant la méthode qui a été décrite plus haut, et qui nécéssite une analyse de compatibilité.

-2d) On applique au résultat obtenu, le traitement décrit au paragraphe 1, et on recommence le traitement décrit au paragraphe 2, jusqu'à ce que tous les automates aient été choisis.

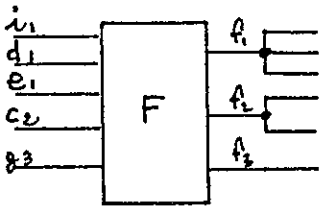
Le procéssus est alors terminé.

Dans ce qui suit nous appliquons l'algorithme au cas particulier de l'assemblage formé par les automates A, B, C, D, E, F, G, dessiné plus haut.

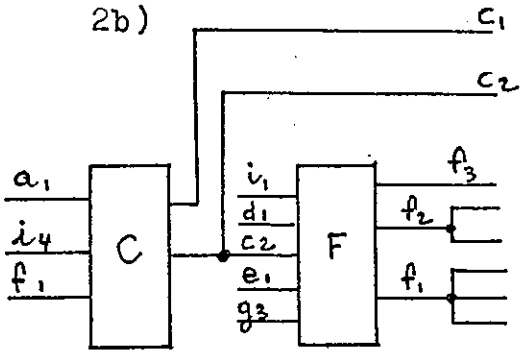
1a)choix:C
1b)



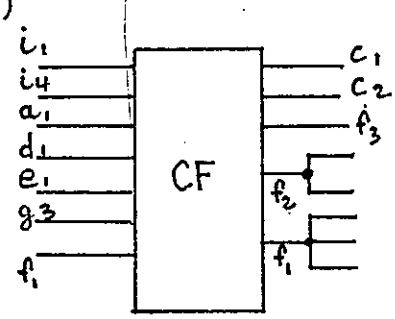
2a)choix:F



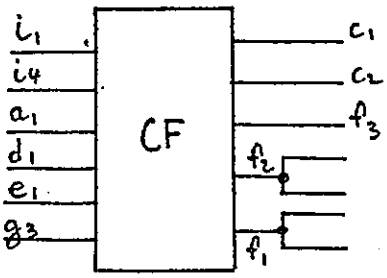
2b)



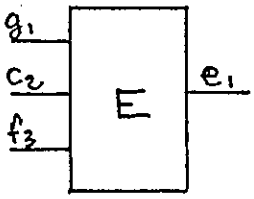
2c)



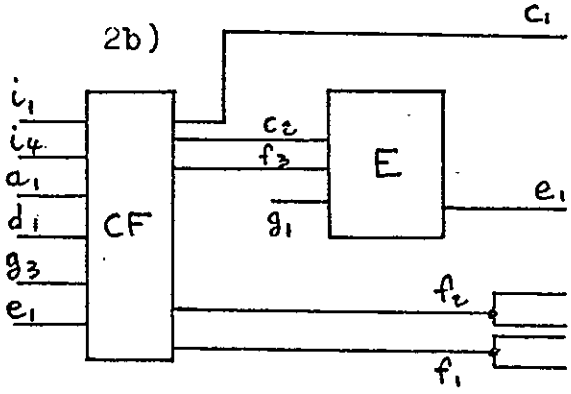
2d)



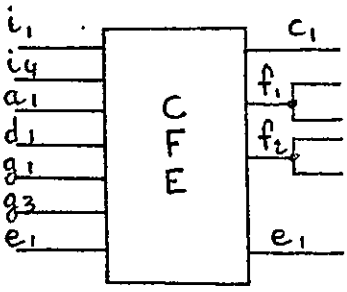
2a)choix:E



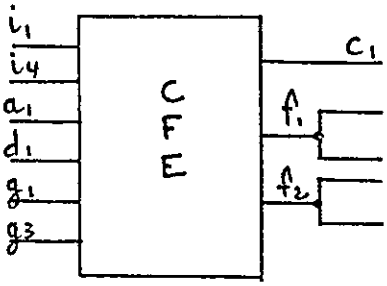
2b)



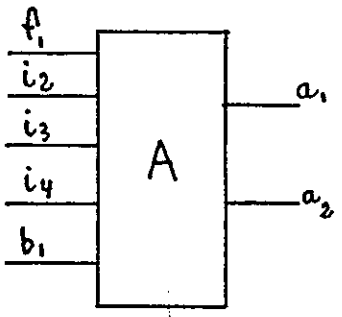
2c)



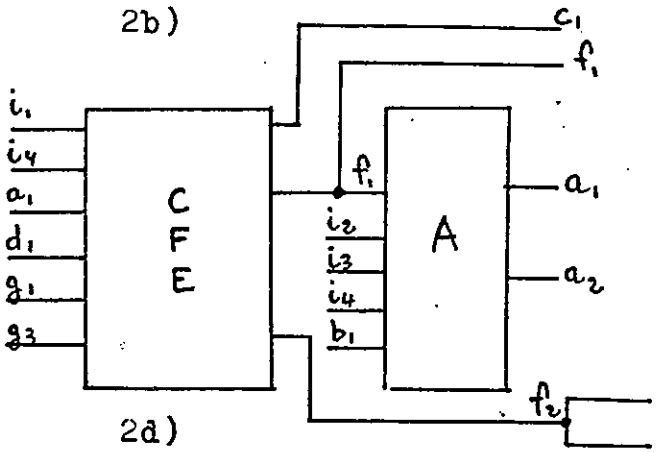
2d)



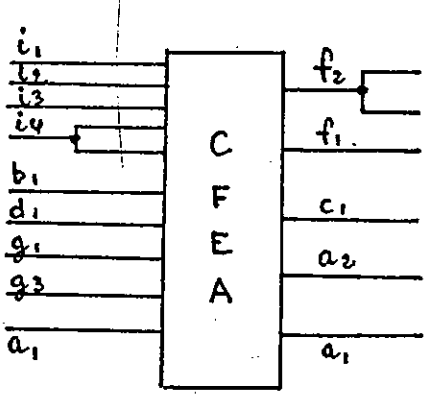
2a)



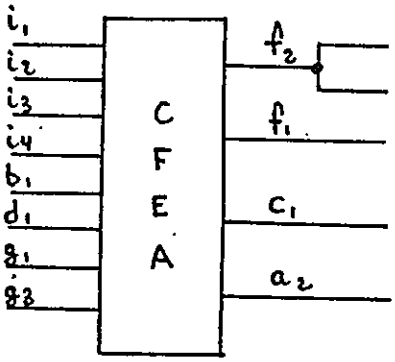
2b)



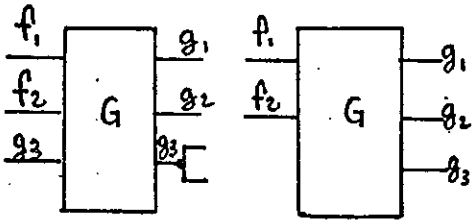
2c)



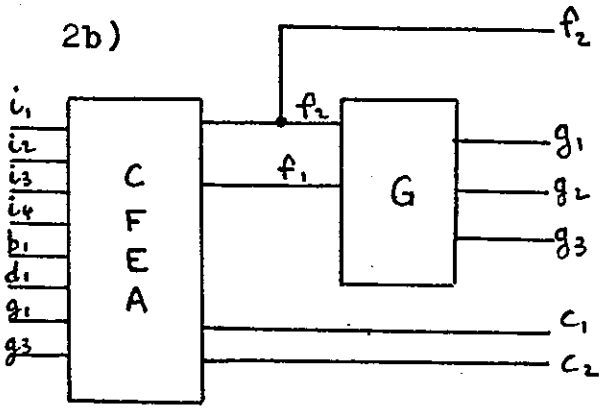
2d)



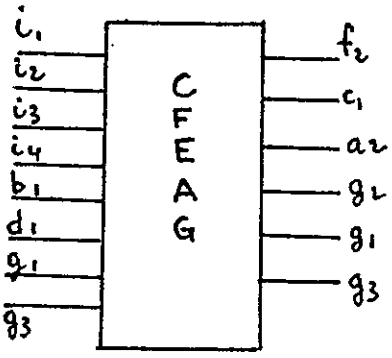
2a)



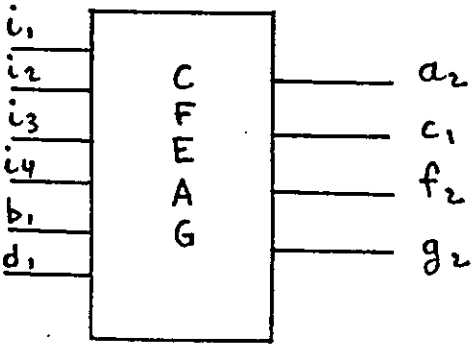
2b)



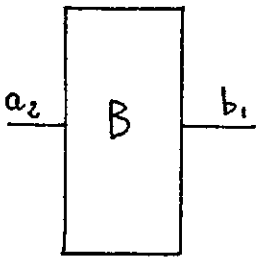
2c)



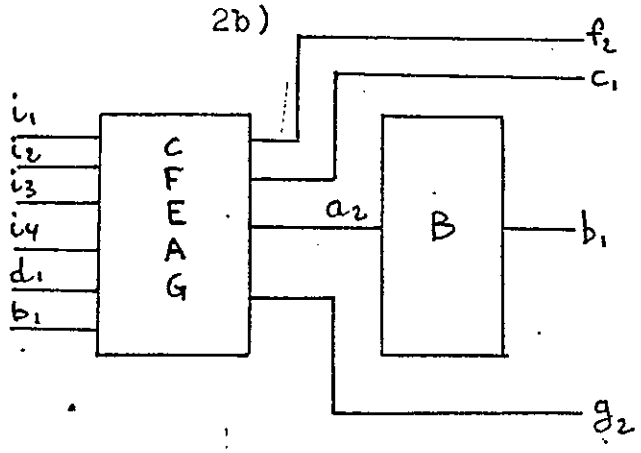
2d)



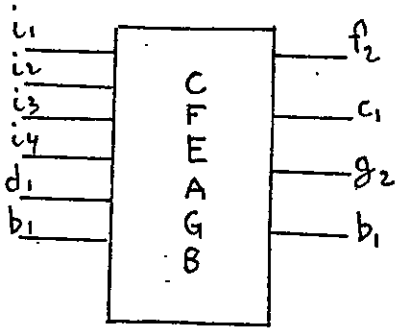
2a)



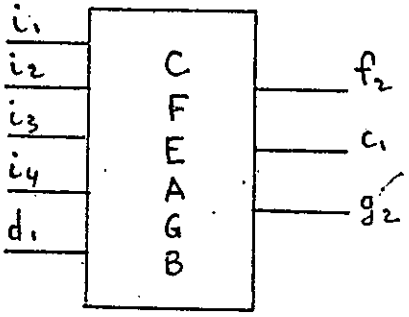
2b)



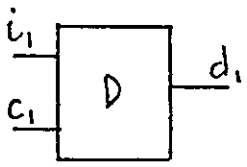
2c)



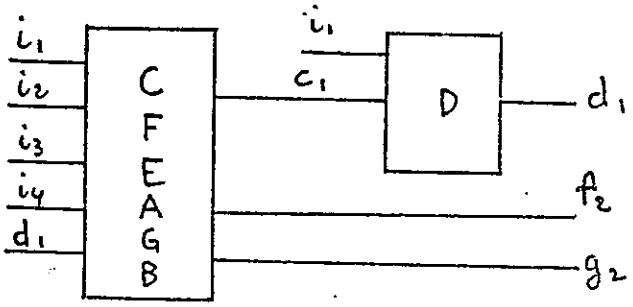
2d)



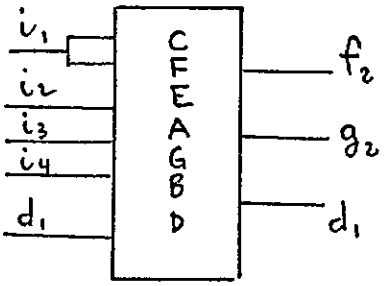
2a)



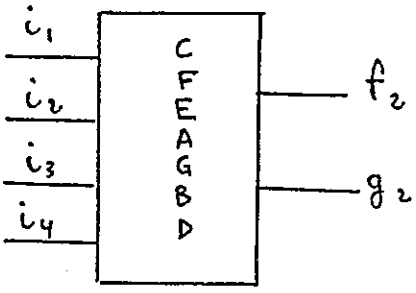
2b)



2c)



2d)



c) Assemblages quelconques

Si on a seulement en vue, de ramener un assemblage quelconque à un automate, il n'est pas indispensable de procéder à une mise en ordre de cet assemblage. Toutefois, il s'avère dans de nombreux cas, qu'une présentation structurée de la circuiterie, peut rendre de grands services. Donc, bien qu'il soit possible d'appliquer à un assemblage quelconque, le traitement algorithmique qui a été décrit plus haut, et ainsi de le ramener à un automate unique, nous allons néammoins indiquer comment mettre en ordre un assemblage, de façon à mettre en évidence, un type de structure canonique, sous laquelle, il est commode de le représenter.

Il y a, deplus, un autre avantage à ordonner les assemblages, dans la mesure où nous montrerons que la technique proposée, contribue à minimiser le nombre des états de l'automate qui remplacera l'assemblage.

- Mise en ordre d'un assemblage:

Dans un premier temps, on s'assure que les règles de bon assemblage ont été respectées; en outre on élimine tout automate qui n'a aucune sortie, ou aucune entrée.

Dans un second temps, on assimile le réseau à un graphe de la façon suivante: chaque automate est un sommet du graphe, et les flèches orientées sont constituées par toutes les liaisons sortie d'automate vers entrée d'automate.
On décompose alors le graphe en composantes fortement connexes. La relation d'équivalence qui est ici en jeu, est la dépendance mutuelle entre automates, au sens où nous l'avons définie dans les pages précédentes.
Formellement on peut exprimer les choses ainsi: un automate B, est dit dépendant d'un autre A, s'il existe une suite finie d'automates Z_1 Z_2 Z_3.....Z_n, telle qu'il existe une sortie de A qui soit une entrée de Z_1, une sortie de Z_1 qui soit une entrée de Z_2, ..., une sortie de Z_n qui soit une entrée de B.

A et B sont dits mutuellement dépendants, si A est dépendant de B, et B de A. On constate sans mal que cette relation entre automates, est une équivalence.

Si on appelle $A^* = \{Z \ / \ Z \text{ dépende de } A\}$

et ${}^*A = \{Z \ / \ A \text{ dépende de } Z,\}$

la classe d'équivalence de A est alors définie par l'intersection des deux ensembles précédents:

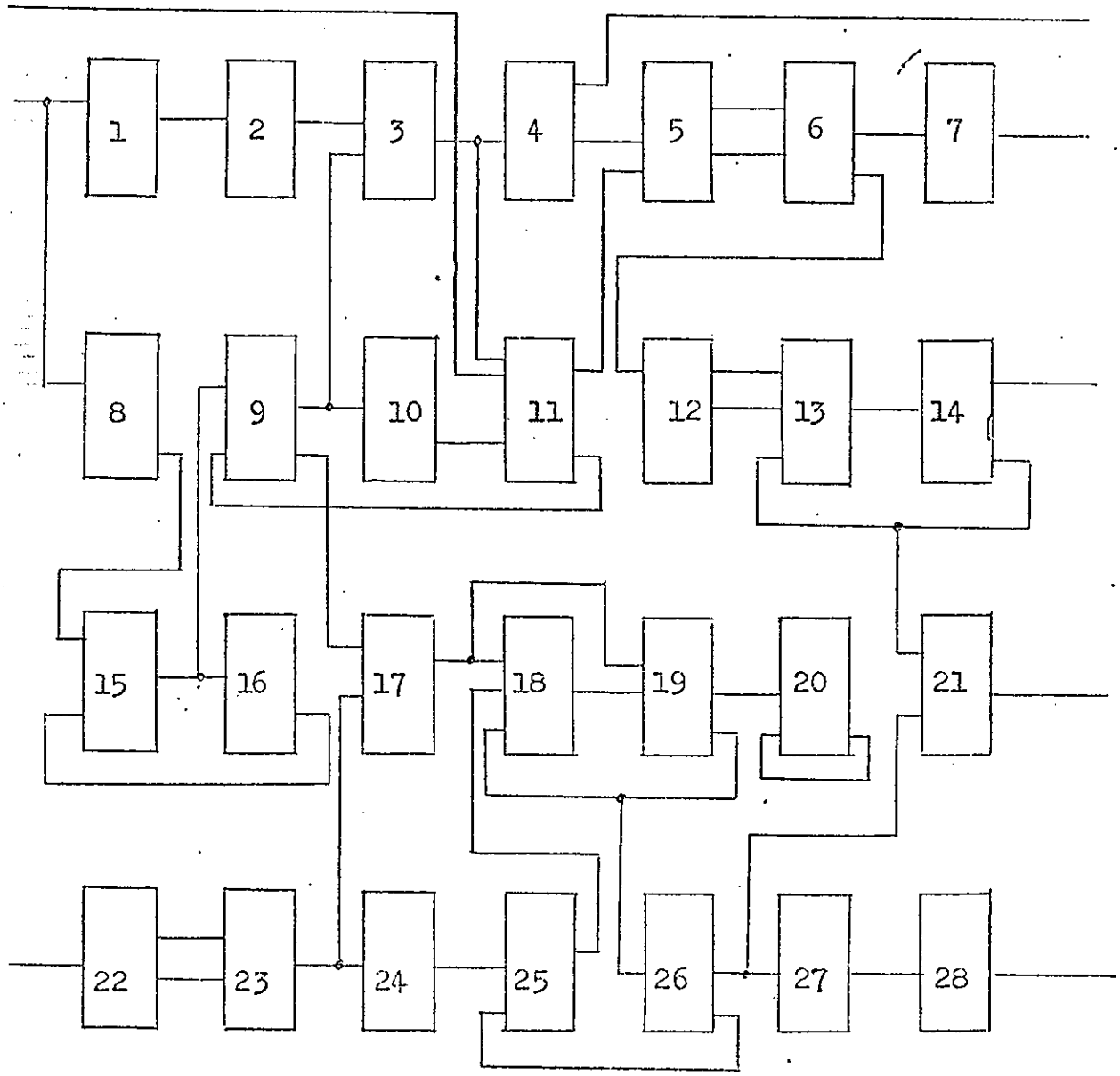
$$\mathcal{C}(A) = A^* \cap {}^*A$$

Nous verrons plus loin sur un exemple, comment mettre en oeuvre l'algorithme représentant cette procédure.

Dans un troisième temps, on forme le graphe réduit de l'assemblage, en assimilant à des sommets, les classes d'équivalence déterminées dans la phase précédente.
Le graphe obtenu, est alors sans circuit, puisque les classes sont distinctes; il est alors possible de le"mettre en ordre". Pour ce, on définit un premier niveau, en isolant les sommets qui n'ont pas de suivant. Il en existe forcément, puisque le graphe est sans circuit. Puis on détermine un second niveau, en isolant les sommets qui n'ont pas d'autres suivants que ceux qui constituent le premier niveau, et ainsi de suite, en régréssant dans le graphe, jusqu'à épuiser l'ensemble des sommets.

Là encore on précisera sur l'exemple comment mettre en oeuvre un algorithme adéquat.

Considérons maintenant l'exemple en question; soit l'assemblage suivant:



Représentons l'assemblage sous forme d'un graphe:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
1		1																										
2			1																									
3				1							1																	
4					1																							
5						1																						
6							1					1																
7																												
8															1													
9			1							1							1											
10											1																	
11					1				1																			
12													1															
13														1														
14													1								1							
15									1							1												
16															1													
17																		1	1									
18																			1									
19																		1		1						1		
20																				1								
21																												
22																							1					
23																	1							1				
24																									1			
25																		1										
26																					1				1		1	
27																												1
28																												

L'automate 20 est supprimé, puisqu'il n'a aucune sortie.
En appliquent l'algorithme défini plus haut, on détermine les classes d'équivalence:

$\mathcal{C}(1)=\emptyset$, $\mathcal{C}(2)=\emptyset$, $\mathcal{C}(3)=\{3,9,10,11\}$, $\mathcal{C}(4)=\emptyset$, $\mathcal{C}(5)=\emptyset$, $\mathcal{C}(6)=\emptyset$
$\mathcal{C}(7)=\emptyset$, $\mathcal{C}(8)=\emptyset$, $\mathcal{C}(12)=\emptyset$, $\mathcal{C}(13)=\{13,14\}$, $\mathcal{C}(15)=\{15,16\}$
$\mathcal{C}(17)=\emptyset$, $\mathcal{C}(18)=\{18,19,25,26\}$, $\mathcal{C}(21)=\emptyset$, $\mathcal{C}(22)=\emptyset$, $\mathcal{C}(23)=\emptyset$
$\mathcal{C}(24)=\emptyset$, $\mathcal{C}(27)=\emptyset$, $\mathcal{C}(28)=\emptyset$

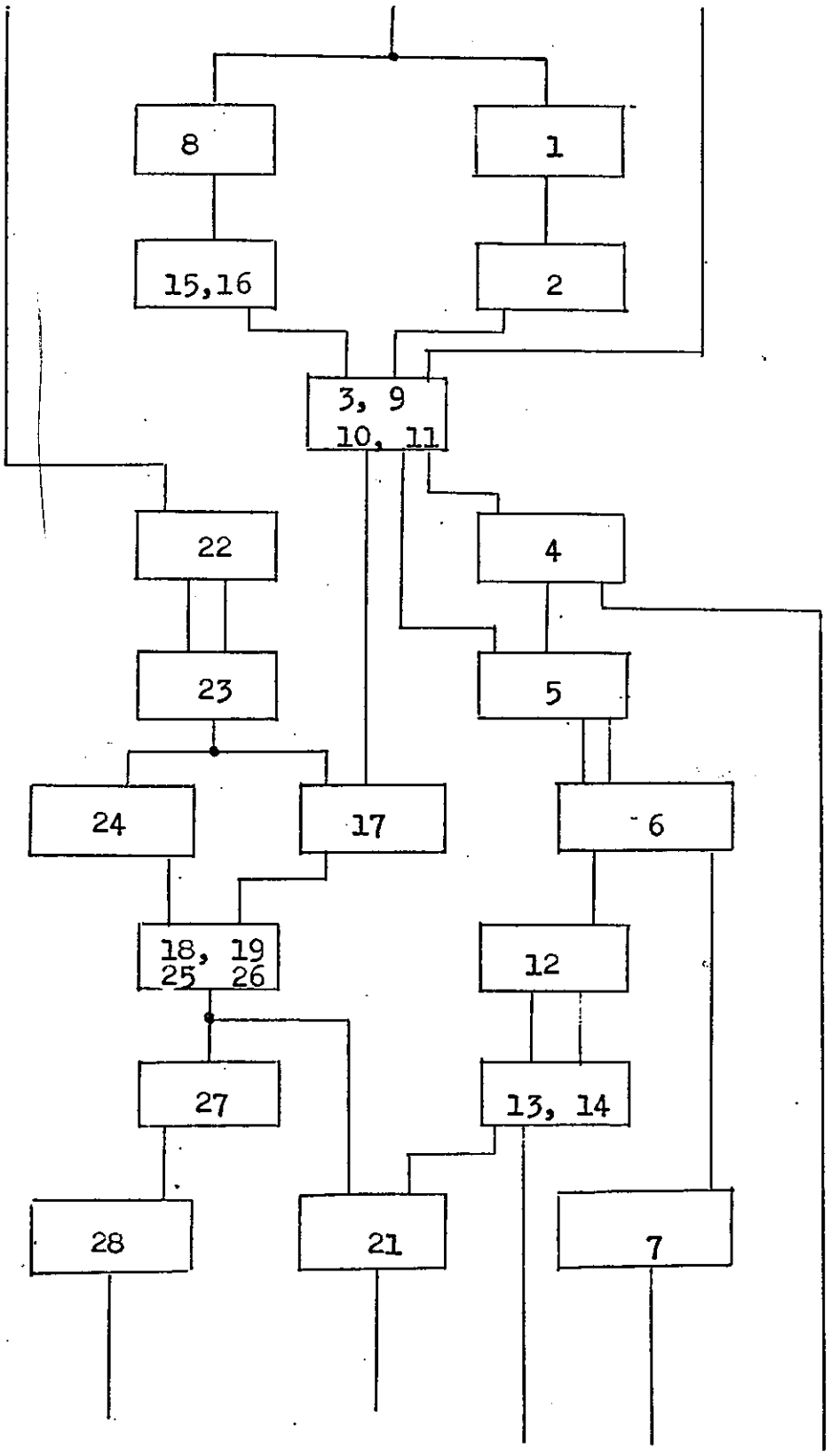
On obtient alors le graphe réduit, dont le tableau est le suivant:

	1	2	3	4	5	6	7	8	12	13	15	17	18	21	22	23	24	27	28
1		1																	
2			1																
3				1	1							1							
4					1														
5						1													
6							1		1										
7																			
8											1								
12										1									
13														1					
15			1																
17													1						
18															1			1	
21																			
22																1			
23												1					1		
24													1						
27																			1
28																			

Dans ce graphe, 7, 21, 28, apparaissent sans successeurs; ils constituent le niveau 1.
On supprime alors dans le graphe, les colonnes et les lignes correspondantes, et on cherche ceux qui sont alors sans successeurs; on trouve {13,14} et 27.
Puis on recommence la même procédure; en définitive, on détermine la succession suivante:

12 et {18,19,25, 26}
6 et 17 et 24
5 et 23
4 et 22
{3, 9, 10, 11}
2 et {15,16}
1 et 8

Le graphe, "mis en ordre", présente alors la forme suivante:
(on a identifié chaque composante fortement connexe à l'automate équivalent)



(à suivre)

LE GRAFCET

Etude et Réflexions

Patrick ISOARDI

CR : Subject classification informatics B.7.1 C.1 C.2.1. D.3.1 J.6 J.7

Résumé : Le GRAFCET est un langage graphique de description pour la conception des automatismes ; ce n'est pas une méthode de résolution.
Cet article est à double but :

- A partir d'un exemple introductif, l'auteur présente les notions fondamentales qui sont suffisantes pour l'interprétation complète et efficace d'un cahier des charges.
- Des réflexions montrent la puissance du GRAFCET et le situent par rapport aux systèmes logiques séquentiels traditionnels.

En matière de conclusion, la dernière partie est une ouverture sur l'utilisation de moyens matériels ou logiciels adaptés à l'implantation du GRAFCET

Sommaire

	Page
• Introduction	1
• Automatismes et cahier des charges	2
• GRAFCET et automates	4
• Les notions fondamentales du GRAFCET	9
• Compléments sur le GRAFCET	16
• Evolutions simultanées	22
• GRAFCET et systèmes logiques séquentiels	28
• Du GRAFCET à la réalisation de l'automatisme	34
• Bibliographie	38

INTRODUCTION

Le GRAFCET n'est pas une méthode de résolution. C'est un langage graphique de description d'automate, conçu pour les automaticiens. Pour la clarté de l'exposé, je le présenterai dans son cadre naturel : l'origine de sa conception, son historique, les notions fondamentales, ... Son utilisation dans un autre domaine (système, échanges, ...) pourra être envisagé ultérieurement.

Le GRAFCET a une histoire récente.

Au début des années 70, le bilan de la conception des systèmes logiques n'était pas brillant. Dans l'industrie, les méthodes théoriques de synthèse (méthode d'Huffman, ...) étaient jugées trop complexes et on préférait des méthodes pratiques le plus souvent empiriques.

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

En 1975, l'AFCET créa un groupe de travail "systèmes logiques" dont le but était de normaliser la représentation du cahier des charges d'un automatisme logique. En 1977, deux ans plus tard, la commission dans son rapport final définissait le GRAFCET : Graphe de Commande Etape-Transition : GR-AFCET pour le rôle important joué par l'AFCET dans cette création.

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) reprit le flambeau pour donner au GRAFCET une forme normalisée. En le diffusant largement, notamment auprès des PMI et de l'enseignement, l'ADEPA a assuré la promotion du GRAFCET.

AUTOMATISMES ET CAHIER DES CHARGES

I. Méthodologie dans la conception des systèmes automatisés.

Elle est nécessaire et repose sur trois idées fondamentales :

1) Dès la conception, le système doit être décomposé en deux parties :
   - La partie opérative,
   - la partie commande.

2) Il est important de donner une description précise du fonctionnement de la partie commande.

3) Il faut utiliser un langage spécifique et rigoureux pour cette description Le GRAFCET.

II. Partie opérative. Partie commande.

Dans un système automatisé, la partie opérative est la machine proprement dite et la partie commande: l'équipement de commande.

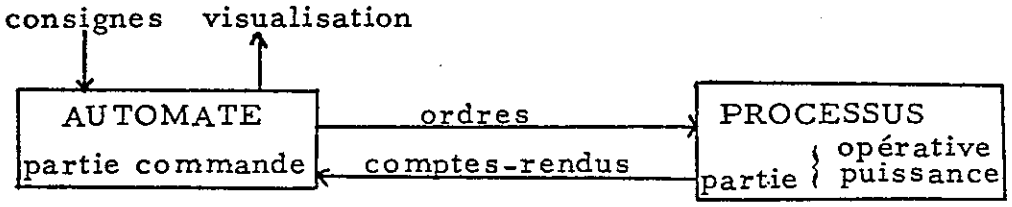
Les deux parties ne sont pas entités disjointes tant sans faut :

. La partie opérative effectue des opérations, lorsque l'ordre lui est donné par la partie commande.
. Grâce aux comptes rendus fournis par la partie opérative, la partie commande est tenue informée de l'état d'avancement des opérations effectuées.

Outre ce dialogue par ordres et comptes-rendus, la partie commande peut échanger des informations avec l'extérieur dont elle reçoit des consignes et à qui elle fournit des comptes-rendus visuels ou sonores.

En résumé :
La partie opérative est le processus à automatiser,
La partie commande est un automatisme qui élabore des ordres de sortie en fonction des informations qu'il reçoit en entrée..

On se limitera ici aux automatismes logiques qui présentent un caractère de tout ou rien.



III. Approche du cahier des charges de la partie commande.

L'automaticien chargé de la conception et de la réalisation de l'automate (ou partie commande) doit rechercher dans le cahier des charge une description de l'équipement à réaliser.

Pour y parvenir, il est souhaitable de diviser la description en deux niveaux :

- le premier niveau décrit le comportement de la partie commande vis à vi de la partie opérative. On spécifie les fonctions.
- le deuxième niveau ajoute aux exigences fonctionnelles les spécifications technologiques.

1) Spécifications fonctionnelles : niveau 1.

Elles caractérisent les réactions de l'automatisme face aux informations issues de la partie opérative.

Ellesdoivent donc définir de façon claire et précise, les différentes fonctions, informations et commandes impliquées dans l'automatisation de la partie opérative, sans préjuger en aucune façon des technologies.

2) Spécifications technologiques : niveau 2.

Elles précisent la façon dont l'automatisme devra physiquement s'insérer dans l'environnement.

C'est à ce niveau que doivent intervenir les renseignements sur la nature exacte des capteurs et actionneurs employés dans la partie opérative.

Le GRAFCET se propose de répondre à de telles exigences.

GRAFCET ET AUTOMATE

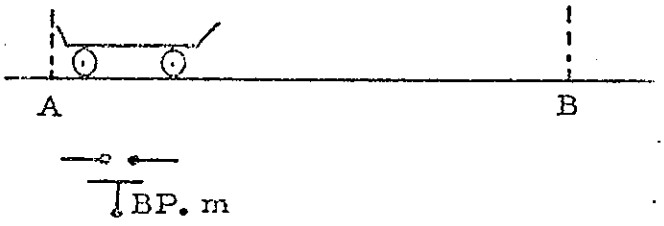
Avant d'introduire les concepts fondamentaux et leur représentatio on va montrer leur importance ainsi que leur signification pratique à partir d'un exemple simple.

I. Les deux niveaux du GRAFCET.

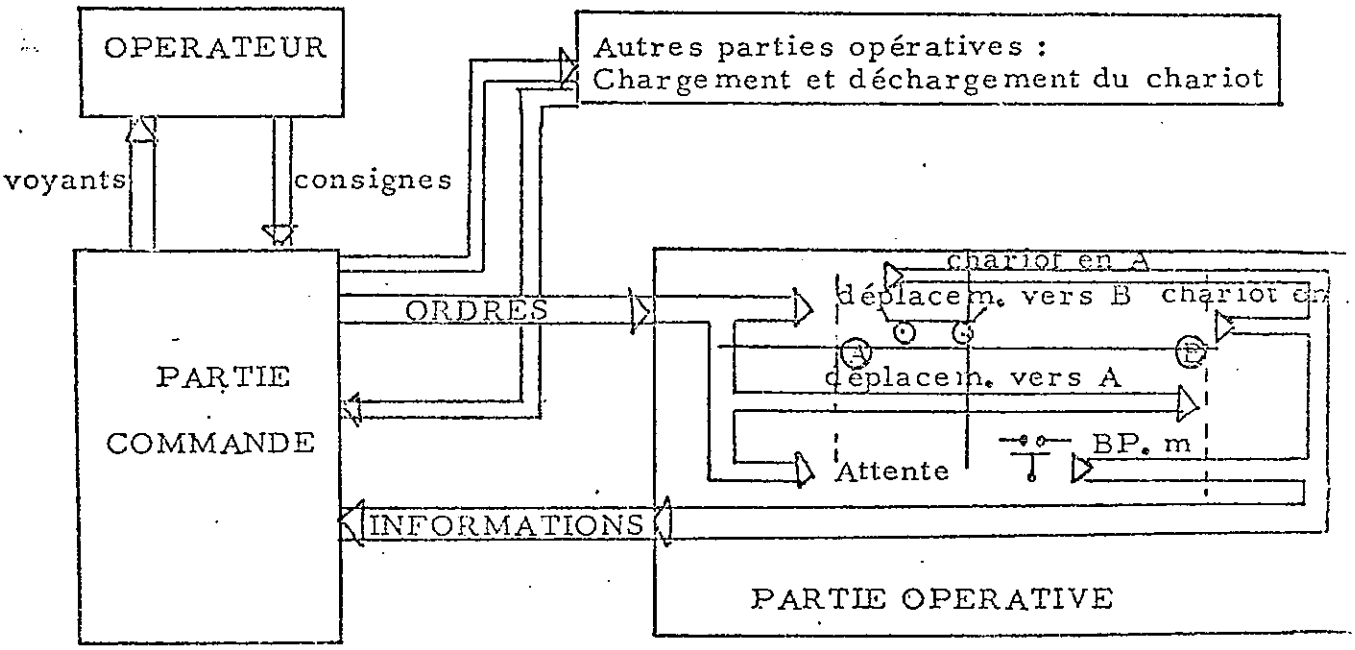
On se propose d'étudier l'automatisation du déplacement d'un chariot d'un point A à un point B et retour en A.

1) Cahier des charges.

Le chariot étant au repos en A, on veut lorsque l'ordre en est donné BP. m (impulsion) qu'il effectue le cycle ABA



2) Découpage partie opérative-partie commande.



Le cycle de travail est une suite d'actions qui ne pourront être obtenues que si la partie commande émet des ordres convenables au moment voulu. Ces moments voulus seront déterminés d'après les informations provenant de la partie opérative.

3) Etude de la partie commande.

Nous dirons que la partie commande demeure dans une "étape" tant que son comportement est constant.

Elle reste dans cette "étape" jusqu'à ce que les informations pour lesquelles elle est "réceptive" provoquent le franchissement d'une "transition" conduisant à une nouvelle étape où la partie commande adoptera alors un nouveau comportement.

Nous pouvons maintenant représenter le fonctionnement de la partie commande comme une succession alternée d'étapes et de transitions

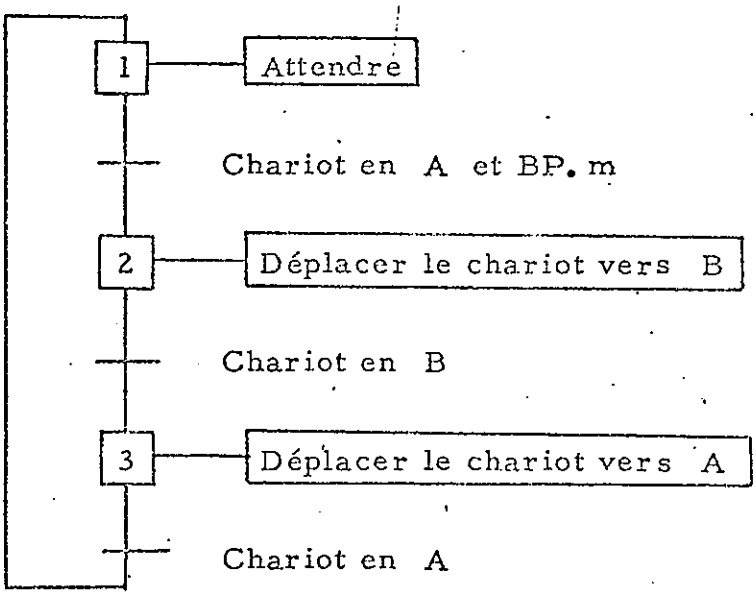
- A chaque étape nous associerons les actions à effectuer.
- A chaque transition, nous associerons les informations permettant leur franchissement sous forme d'une condition logique appelée réceptivité.

De cette manière le fonctionnement de la partie commande nécessaire au fonctionnement du chariot sera ainsi décrit.

Etape 1 ; action, attente.
Transition 1-2 ; réceptivité ; chariot en A et BP. m.
Etape 2 ; action ; Déplacer le chariot vers B.
Transition 2-3 ; réceptivité ; chariot en B .
Etape 3 ; action ; Déplacer le chariot vers A .
Transition 3-1 ; réceptivité ; chariot en A .

Il s'avère plus commode de représenter ce fonctionnement sous forme graphique d'où l'intérêt du GRAFCET.

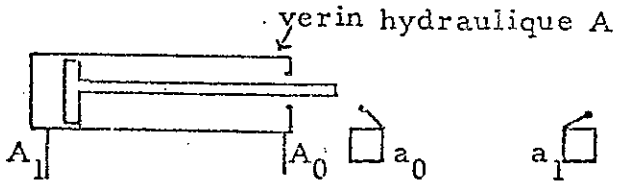
4) Le GRAFCET de niveau 1 : Spécifications fonctionnelles.



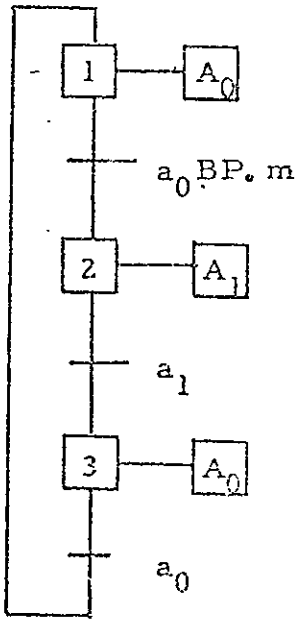
5) Passage au niveau 2 : Définition des capteurs et additionneurs.

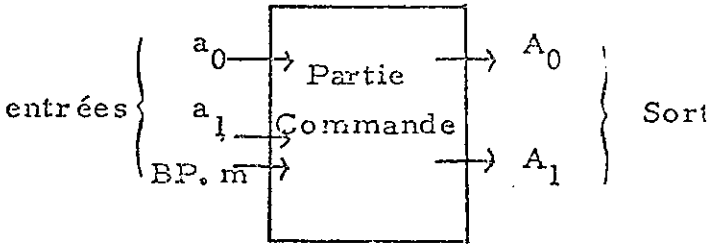
Le GRAFCET que nous venons d'établir ne prend en compte que l'aspect fonctionnel sans aucune implication technologique.

Il convient maintenant de préciser les choix technologiques des actionneurs et capteurs :



6) GRAFCET de niveau 2 : Spécifications technologiques.





II. Construction de l'automate.

La vue du GRAFCET nous permet d'obtenir le schéma ci-dessous de la partie commande.

1) Equations des mémoires (étapes).

N° de l'étape	Equations d'activations	Effacement ou désactivation
1	$a_0 . 3$	2
2	$a_0 . BPm . 1$	3
3	$a_1 . 2$	1

avec la règle : "on active avant de désactiver".

2) Equations des sorties.

$A_0 = 1+3$
$A_1 = 2$

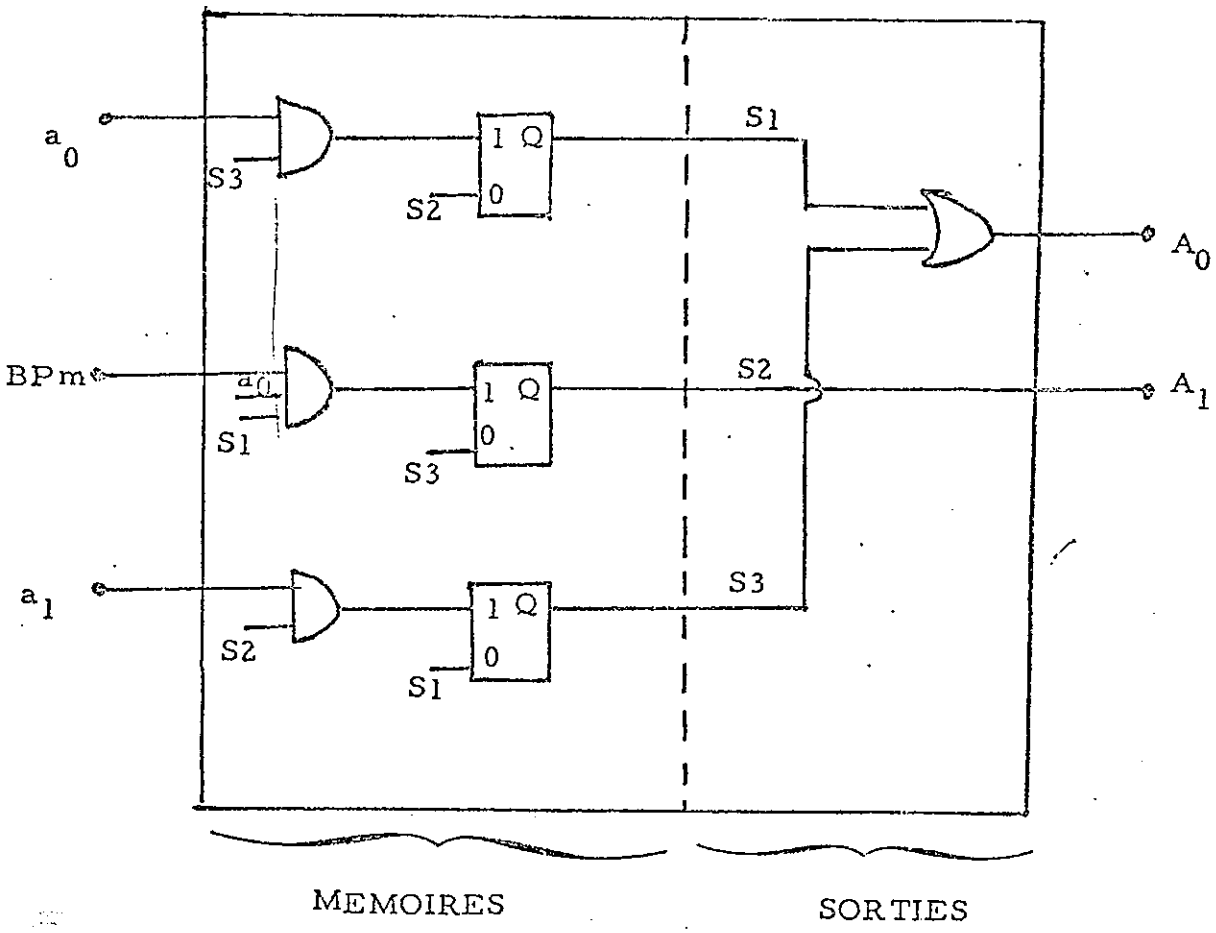
3) Automatisation de l'étape.

Le module d'étape comprend 3 fonctions de base.

- Une fonction mémoire
  Elle maintient l'étape en activité pendant toute la durée de la phase.
- Une fonction ET
  C'est la fonction d'activation.
- Une fonction OU
  C'est la fonction de désactivation.
  Elle permettra la mise à 0 de l'étape.

NB. L'activité d'une étape est concretisée par une sortie.

4) Logigramme: Cablage de l'automate



LES NOTIONS FONDAMENTALES DU GRAFCET

GRAFCET est le sigle : Graphe de Commande Etape-Transition.

Le GRAFCET n'est pas une méthode de résolution mais un langage graphique de description.

I. Les objets.

Le fonctionnement d'un automatisme peut être représenté graphiquement par un ensemble :

- d'étapes auxquelles sont associées des actions
- de transitions liées à des réceptivités
- de liaisons (ou arcs) orientés.

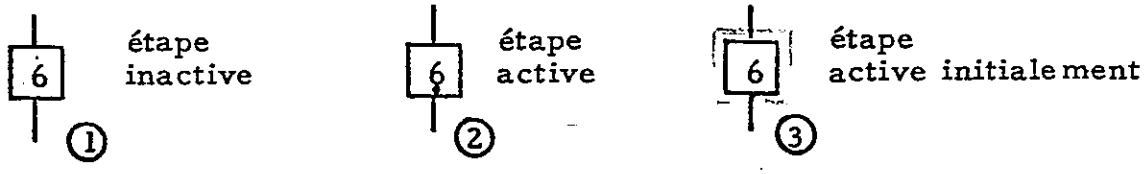
Chaque liaison orientée relie une étape à une transition ou une transition à une étape.

Le sens des parcours des informations est généralement du haut vers le bas. Sinon l'orientation de l'arc est obligatoire.

1) Les étapes.

. Une étape est une situation du système dans laquelle tout ou partie de l'organe de commande est invariant par rapport aux entrées et aux sorties.

. L'étape se représente par un carré ou un rectangle repéré numériquement (fig. 1)



. Une étape peut être active ou inactive.
A un moment donné, la situation de l'automatisme est entièrement définie par l'ensemble des étapes actives.
On précise les étapes actives à un moment donné, en plaçant à l'intérieur du carré un point (ou jeton) (fig. 2).

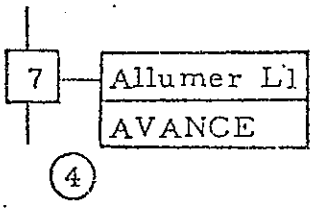
. Les étapes initialement activées se représentent par un double carré et indiquent de ce fait le début du fonctionnement du cycle (fig. 3).

2) Les actions.

Pour chaque étape, on précise les actions à effectuer caractéristiq de la situation du système.

Ces actions ne sont exécutées que lorsque l'étape est active.

Ces actions sont décrites littéralement ou symboliquement à l'intérieur d'un rectangle de dimensions quelconques situé à droite de l'étape (fig. 4).



L'exécution de ces actions peut être soumise à d'autre conditions logiques fonctions :

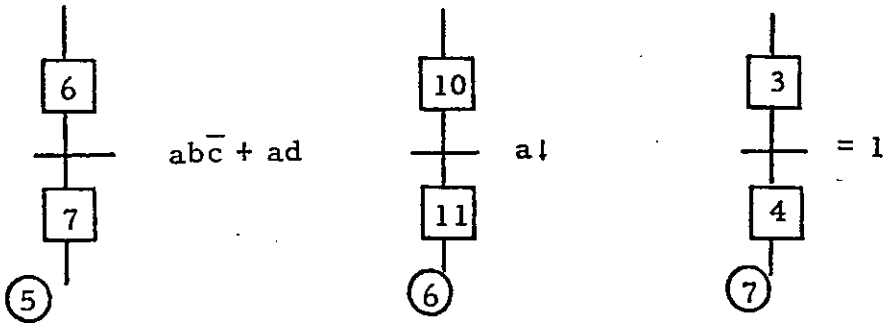
- de variables d'entrée
- de variables auxiliaires
- de l'état actif ou inactif d'autres étapes

Nous reviendrons ultérieurement sur les actions conditionnelles.

3) Transitions - Réceptivités.

Une transition indique la possibilité d'évolution entre étapes. A chaque transition, on associe une condition logique appelée réceptivité. La réceptivité est une fonction combinatoire des variables d'entrée, et de l'état actif ou inactif d'autres étapes (éventuellement).

La transition se représente par une barre. Sur la partie droite de cette barre on indique la réceptivité associée (fig. 5).



Les réceptivités peuvent aussi faire intervenir des changements d'état de variables.

- La notation a↑ représente le "front montant" de la variable a (passage de 0 à 1)
- La notation a↓ représente le "front descendant" de la variable a (fig. 6

Une réceptivité toujours vraie est écrite = 1 (fig. 7).

II. Les règles d'évolution.

Le caractère actif ou inactif de chacune des étapes devant évoluer, les concepts précédents ne peuvent suffire à définir un GRAFCET.

Il est en plus nécessaire de fixer un ensemble de règles d'évolution

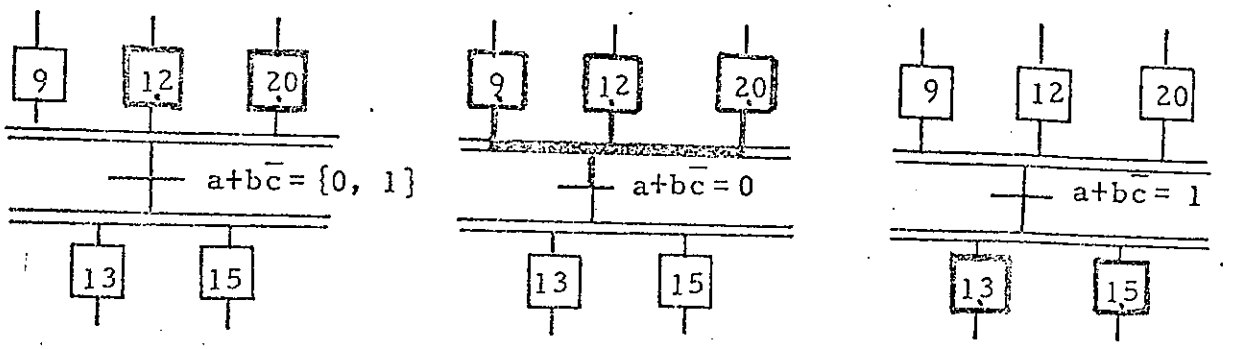
Règle 1 : L'initialisation repérée par le doublage des côtés d'une étape (fig. précise les étapes actives au début du fonctionnement. Elles sont activées sans condition.

Règle 2 : Une transition est soit validée soit non validée.
Elle est validée lorsque toutes les étapes immédiatement précédentes sont actives. Elle ne peut être franchie que :

et
- lorsqu'elle est validée
- la réceptivité associée à la transition est vraie.

La transition est alors obligatoirement franchie.

Règle 3 : Le franchissement d'une transition entraîne l'activation de toutes les étapes immédiatement suivantes et la désactivation de toutes les étapes immédiatement précédentes.



Transition non validée Transition validée Transition franchi

N. B. Lorsque plusieurs étapes sont reliées d'une même transition, on convient de représenter le regroupement de liaisons par deux traits parallèles.

Règle 4 : Plusieurs transitions simultanément franchissables sont simultanément franchies.

Règle 5 : Si au cours du fonctionnement une même étape doit être désacti et activée simultanément, elle reste activée.

Remarque : La durée de franchissement d'une transition ne peut jamais être rigoureusement nulle, même si théoriquement elle peut être rendue aussi petite que l'on veut.

Il en est de même de la durée d'activation d'une étape.

III. Comprendre et appliquer les notions fondamentales du GRAFCET.

Nous allons ici expliquer les définitions du GRAFCET que nous venons de donner dans le chapitre précédent.

Il n'est pas possible de préciser l'une après l'autre ces notions de bases tant elles sont fortement imbriquées.

Nous allons tenter de le faire sentir à partir d'un exemple de la vie courante.

Exemple : Un piéton P veut traverser une avenue protégée par un feu de signalisation.

P attend que le feu pour piétons passe au vert.
Lorsque le feu est vert P marche.
Quant le feu passe au rouge P court.
Lorsque P a atteint le trottoir opposé P pour suit son chemin en marcha normalement.

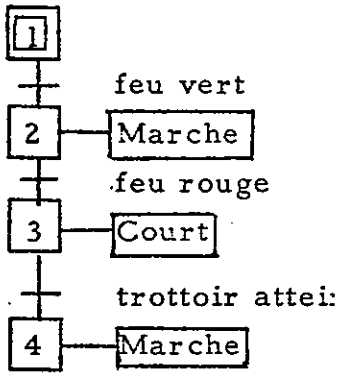
Exprimons cette scène par un GRAFCET.

Qu'est-ce qu'une étape ?

C'est une situation dans laquelle le comportement de P ne varie pas : Un état stable.
Les comportements stables de P dans le temps sont :

- attente
- marche
- court
- marche

} auxquels on associe les étapes



III. 1

Les actions ?

Les actions possibles de P sont MARCHE et COURSE.
A l'attente n'est associée aucune action. On dit aussi que c'est un état initial de repos.

Elles correspondent aux sorties d'un automate.

La transition ? La réceptivité ?

Elle indique la possibilité d'évolution entre étapes. Cette condition est la réceptivité de P à cet instant. C'est une entrée de l'automate.

Comment se fait l'évolution ?

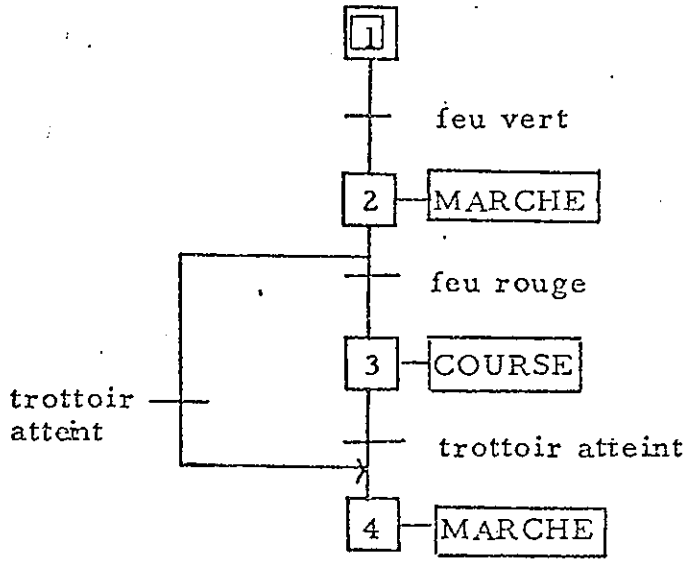
Le GRAFCET est un outil dynamique. C'est la notion d'activité d'une étape qui introduit l'évolution.
A la vue de cet exemple.
Pour que l'évolution d'une étape à une autre se produise il faut :

et - que l'étape soit active
- que la transition soit validée

Si P n'est pas présent sur le trottoir au moment où le feu passe au vert le franchissement de la transition n'est pas réalisé.

Le GRAFCET de la figureIII. 1 ne représente que la scène que nous avons décrite. Nous n'avons pas envisagé l'éventualité que P atteigne le trottoi avant que le feu passe au rouge.
Nous allons rajouter cette éventualité à notre GRAFCET.
Il faut rajouter une transition permettant d'évoluer de l'étape 2 à l'étape
La réceptivité associée à cette transition doit être : "trottoir atteint".

On a le nouveau GRAFCET suivant :



Dans ce nouveau GRAFCET, si le feu passe au rouge en même temps que le trottoir est atteint les transitions sont franchies simultanément d'après la règle 4. Il y a alors incompatibilité sur l'action qui est COURSE en même temps que MARCHE.

Il faut donc que le choix soit exclusif et même prioritaire pour le trottoir atteint. Réfléchissons. Que doit faire P lorsque le trottoir est atteint, il doit marcher. Quand doit-il courir ? Lorsque le "trottoir n'est pas atteint et que le feu est rouge". Voici donc la nouvelle transition entre l'étape 2 et l'étape 3, qui remplace la précédente : "feu rouge

Cet exemple montre déjà les pièges que peut recéler un GRAFCET si tous les cas possibles n'ont pas été attentivement examinés.
Ceci justifie l'affirmation précédente : le GRAFCET est un langage graphique Si ce n'est pas une méthode de résolution, il permet pourtant de poser des question ; C'est donc une aide à la conception.

COMPLEMENTS SUR LE GRAFCET

Ces compléments qui viennent s'ajouter aux notions de base, permettent de préciser certaines définitions.

I. Le temps : une réceptivité particulière.

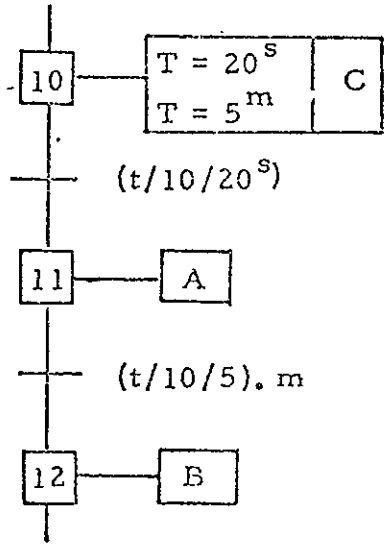
On peut pour compter le temps utiliser un élément technologique particulier qui délivre un signal de fin de comptage après enclenchement et éventuellement affichage de la durée.

Le problème ici posé est de représenter explicitement le temps dans un GRAFCET, sans faire aucune hypothèse sur l'élément technologique utilisé.

Pour faire intervenir le temps dans une réceptivité, il suffit d'indiquer son origine et sa durée.

- L'origine sera le début de la dernière activation d'une étape antérieure. Lorsqu'une étape se trouve à l'origine d'un temps, il est utile de l'indiquer dans les actions relatives à cette étape.

- La réceptivité attachée au temps prendra une notation particulière t/i/d où :
  - t est un repère indiquant qu'on caractérise la variable temps
  - i est le numéro de l'étape origine
  - d est la durée depuis la dernière activation de l'étape origine



- Ici, 20^s après l'activation de l'étape 10 on passe dans l'étape 11

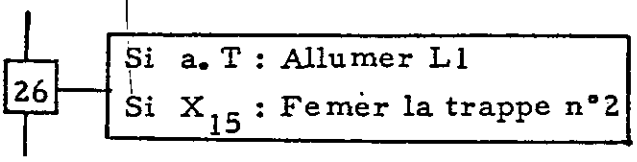
- 5 minutes après l'activation de l'étape 10 et si m = 1 on passe dans l'étape 12 .

II. Actions conditionnelles.

Dans les notions fondamentales du GRAFCET, nous avons décrit des actions inconditionnelles qui sont effectuées dès que l'étape est active. Mais l'exécution de certaines actions peut être soumise à d'autres condition logiques fonctions, des variables d'entrées, de l'état actif ou inactif d'autres étapes, ...

Une telle action est dite conditionnelle.

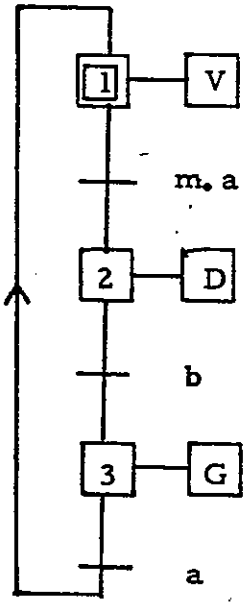
- Elle est exécutable lorsque l'étape est active.
- Elle sera exécutée lorsque la condition associée est vraie.



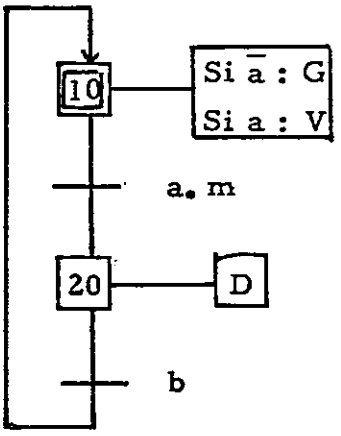
N. B. : On note X_i la variable booléenne correspondant au caractère actif ou inactif de l'éta i

L'utilisation d'actions conditionelles est souvent un moyen de simplifier un GRAFCET. Nous allons illustrer cette simplification sur l'exemple suivant : C'est une amélioration de l'exemple introductif.

Un chariot se déplace de A en B (A : chargement, B : déchargem en A et B nous avons des capteurs a et b.
Le chariot se déplace en actionnant un bouton poussoir de mise en marche : A l'état initial le chariot est en A. Ceci est signifié par un voyant V.
Le GRAFCET de la figure 1 décrit le processus.



1



Ceci peut être consid comme un sous-GRAFCET.

2

On peut également représenter le processus par le GRAFCET de la figure 2
Il existe des méthodes pour passer du GRAFCET de la figure 1 au GRAFCET de la figure 2 en fusionnant des étapes.
En tous les cas, les deux GRAFCET décrivent le même fonctionnement norm
Mais si on sort des hypothèses normales faites sur le comportement du processus, les fonctionnements peuvent différer. En effet :
Supposons que le chariot lors de son retour en A dépasse la position du capteur a.

- Dans le cas de la figure 1, lorsque le chariot atteint A l'étape 1 est activée et on est réceptif à m : que a reste vrai ou non, le chariot reste immobilisé jusqu'à ce que l'action de l'opérateur sur le bouton m provoqu un nouvel aller-retour.
- Dans le cas de la figure 2, lorsque le chariot atteint A, sa situation ne change pas : l'étape 10 déjà active reste active, mais l'action G conditionnée par $\bar{a}$ cesse. S'il y a dépassement de la position A, a va devenir de nouveau faux et l'action conditionnelle G sera de nouveau effectuée ce qui entraîne le chariot vers l'infini "gauche". Même une action sur le bouton poussoir m reste inefficace.

Cette différence de comportement entre les deux GRAFCET introduit une nouvelle notion :

Définitions :

- Une situation S est réceptive à une condition logique, si cette condition logique fait évoluer la situation du GRAFCET.
- Une situation S est sensible à une condition logique si cette condition logique, sans faire évoluer la situation, a une influence sur l'exécution des actions exécutables.

- La situation (3) du GRAFCET de la figure 1 est réceptive à la condition a.

La situation (10) du GRAFCET de la figure 2 est réceptive à la condition a. m et sensible aux conditions a et $\bar{a}$.

III. Actions à niveau - Actions impulsionnelles.

Nous avons vu des actions conditionnelles et inconditionnelles. Sans entrer dans le détail, nous allons maintenant montrer que les actions peuven encore être de natures différentes.

On conçoit que les deux actions :

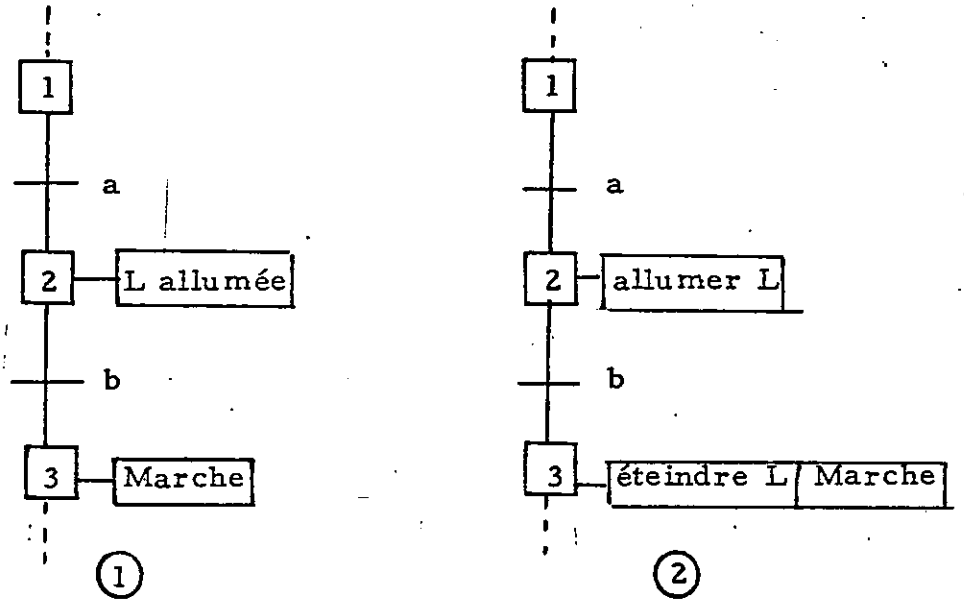
- maintenir la lampe L allumée

ou

- allumer la lampe L

sont différentes bien qu'elles conduisent au même effet : l'éclairage de la lampe.

Considérons les deux GRAFCET suivants :



Nous comprenons immédiatement le premier GRAFCET :

Tant que l'étape 2 est active la lampe est allumée.

Lorsque la réceptivité b est vraie, l'étape 3 est activée, l'étape 2 desactivée et la lampe L éteinte.

Nous avons une action continue dite à niveau.

Dans le deuxième exemple, les actions associées aux étapes sont exécutées une seule fois, dès que l'étape devient active.

Ainsi, dès que l'étape 2 est active, la lampe est allumée.

Elle restera allumée, même si l'étape 2 est desactivée, jusqu'à ce qu'une nouvelle action vienne l'éteindre. Ici, c'est le cas de l'action éteindre L associer à l'étape 3.

Nous avons ici deux actions impulsionnelles ou ponctuelles.

Pour être correct, il faudra préciser la durée d'exécution de l'action impulsionnelle.

Tout ceci est formalisé de la façon suivante :

Définitions :

- L'action à niveau est exécutée tant que l'étape à laquelle elle est associée reste active.
- L'action impulsionnelle est exécutée dès que l'étape devient active. On lui associe une durée déterminée.

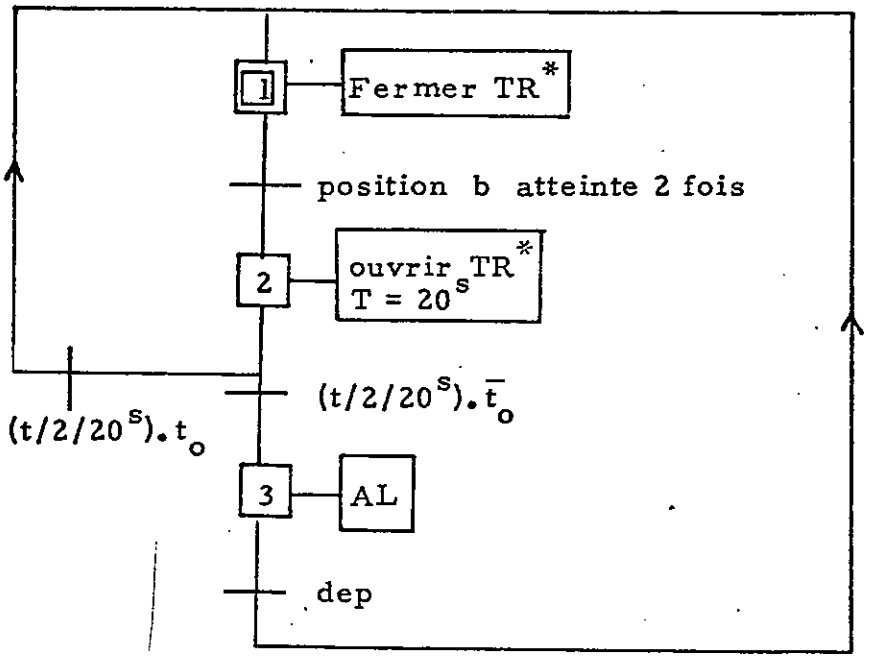
Pourfaire la distinction entre ces deux actions, on notera une action impulsionnelle d'un astérisque suivi d'une durée.
Par exemple L*/2ms signifie que la sortie L reste à l'état 1 pendant 2ms.

Le tableau ci-dessous résume toutes les combinaisons possibles entre actions conditionnelles, inconditionnelles, à niveau ou impulsionnelles et les liens qui existent entre action exécutable et effectivement exécutée.

Action	à niveau		impulsionnelle	
	inconditionnelle	conditionnelle	inconditonnelle	conditionnelle
exécutable	si l'étape associée est active			
exécutée	Tant qu'elle est exécutable	. Si elle est exécutable . et tant que la condition associée est vraie	dès qu'elle devient exécutable et une seule fois	Si elle est exécutable et chaque fois que la condition logique associée est vraie

Exemples d'actions impulsionnelles :

On veut construire un automatisme assurant l'évacuation d'objets. Ces objets arrivent sur un poste b et doivent être évacués par groupe de deux grâce à une trappe TR que l'on ouvre pendant 20s. (L'intervalle séparant l'arrivée de 2 objets successifs au poste b est supérieur à ce temps).
Il est important de signaler par une alarme (AL) tout incident de fonctionnement de la trappe pour éviter l'accumulation des objets.
Après remise en état de la trappe une information (dep) doit permettre le retour en fonctionnement normal.
L'information trappe ouverte sera notée t_o.
Ceci étant précisé, le cahier des charges de niveau 1 est représenté par le GRAFCET ci-après.


1
Fermer TR*
position b atteinte 2 fois
2
ouvrir TR*
T = 20s
(t/2/20s).t̄o
(t/2/20s).to
3
AL
dep

EVOLUTIONS SIMULTANEES

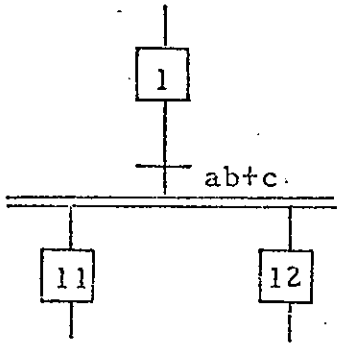
On parle de parallélisme ou d'évolutions simultanées lorsque le GRAFCET possède, à un instant donné, plusieurs étapes actives. La façon la plus simple d'engendrer cet état consiste lors du franchissement d'une transition à activer plusieurs étapes de sortie.
Ceci marque bien le début d'évolutions simultanées ; C'est du parallélisme "structural".

I. Parallélisme structural.

Le parallélisme structural est spécifié par la structure du graphe et quelles que soient les réceptivités associées aux transitions. Dans la représentation normalisée, il est repéré par un double trait de regroupement de liaisons.

1) La distribution : Divergence en ET.

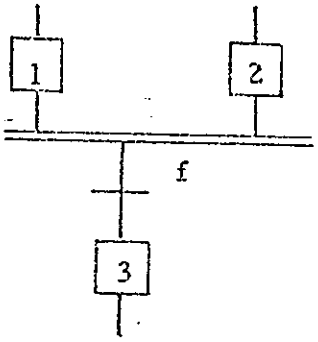
Elle marque le début du parallélisme.



Les étapes 11 et 12 deviennent simultanément actives si l'étape 1 l'
et la condition ab+c est vraie.

2) La fonction : Convergence en ET.

Elle marque la fin des évolutions parallèles.

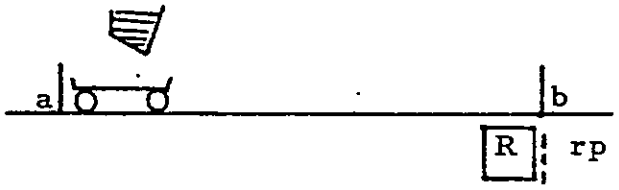


L'étape 3 devient active si les étapes 1 et 2 sont actives et la condition f est vraie.

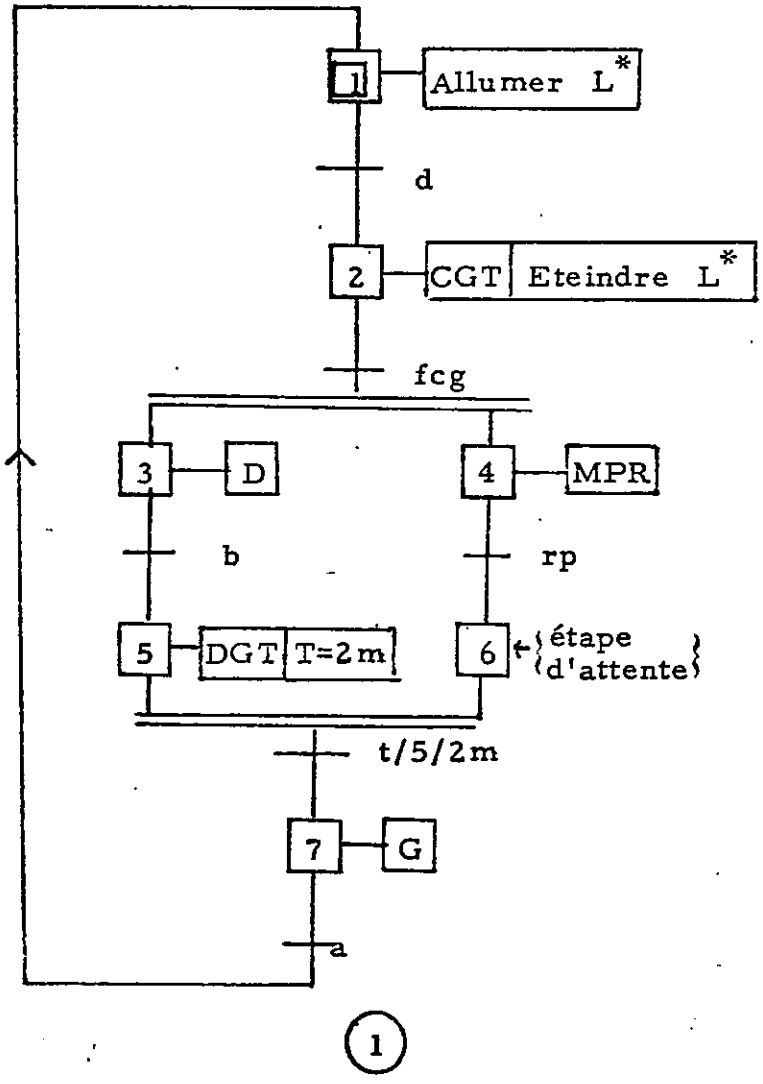
3) Exemple de séquences simultanées.

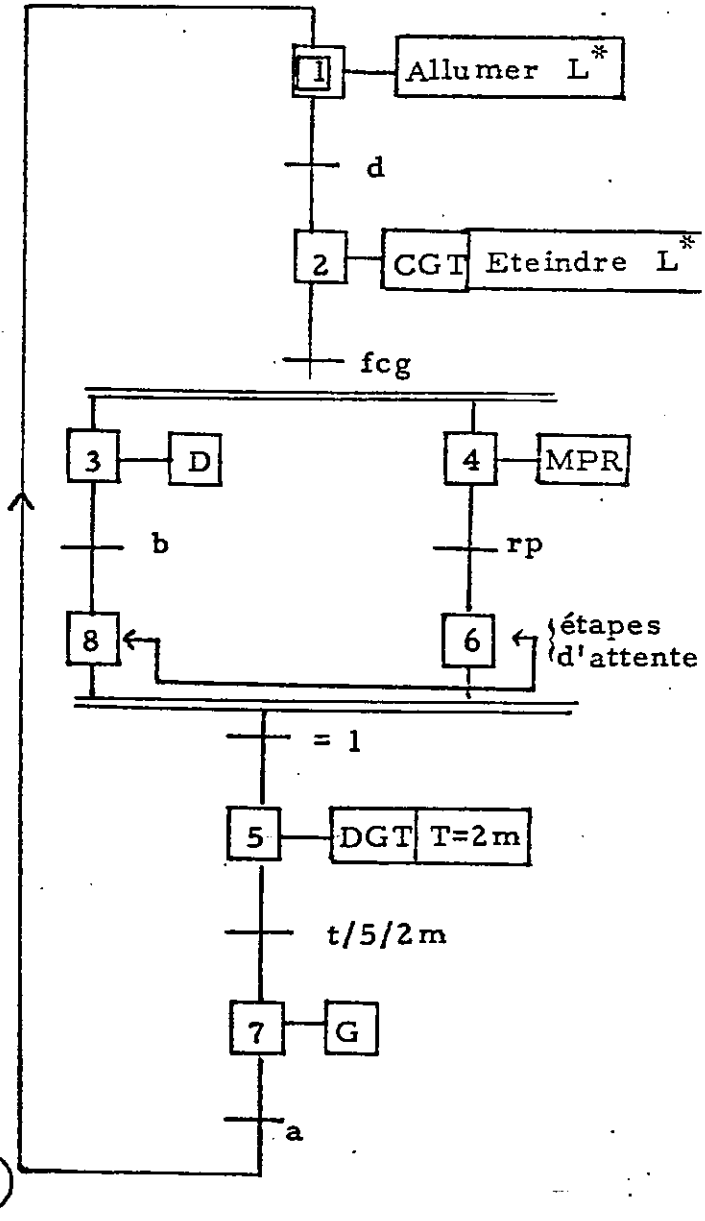
Un chariot se déplace sur des rails entre 2 positions a et b. Le chariot étant au repos en a (visualiser par une lampe L) lorsque l'ordr lui est donné (d), il se charge de poudre (CGT). Lorsque ce chargement est terminé (fcg) il se déplace jusqu'à la position b où son déchargement (DGT) est effectué dans un récipient R. Le déchargement se fait en 2 minutes. Celui-ci terminé, le chariot revient en a.

Le récipient R est mis en place (MPR) pendant que le chariot se dirige vers b. La fin de l'opération de mise en place est signalée par une information (rp).



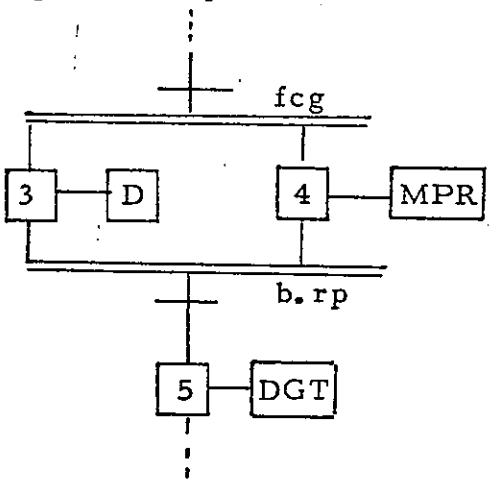
Si on fait une 1ère ébauche rapide, on obtient le GRAFCET (1).





Remarque : Si la mise en place du récipient dure plus longtemps que le traje ab du chariot, le chariot arrive au poste b et effectue son déchargement alors que le récipient n'est pas là. Soit ceci est réellement impossible, soit c'est un oubli du cahier des charges. En tous les cas, il faut mieux envisager cette éventualité pour des raisons de sécurité. Pour cela on va modifier le GRAFCET précédent dans lequel on introduit une étape supplémentaire 8 correspondant à l'arrêt du chariot en b sans déchargement. C'est au même titre que l'étape 6 qui est une étape d'attente pour le récipient, une étape d'attente pour le chariot (figure 2).

A la vue du nouveau GRAFCET obtenu, une nouvelle simplification est tentante. Pourquoi ne pas remplacer les deux étapes 6 et 8 et la transition = 1 par une seule transition b.rp.
La partie en question se représentera ainsi :



C'est un gros risque qui est pris dans ce cas là, car le fonctionnement norma ne sera assuré que si b et rp passe simultanément à 1. Ce qui est peu probable.

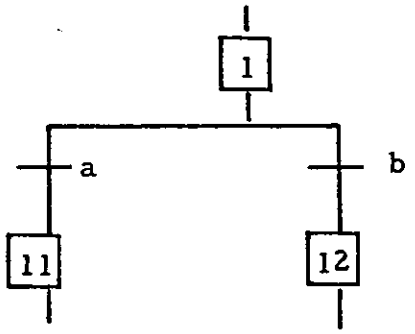
On retrouve là, une propriété précédente du GRAFCET ; Il est une aide pour spécifier l'automatisme. Il permet de poser des questions pour n'oublier aucune situation, aucune action, ...

Le parallélisme structural n'est pas le seul à engendrer des évolutions simultanées. Nous en verrons d'autres exemples.

II. Les séquences multiples.

Le GRAFCET est généralement constitué de plusieurs séquences et il est souvent nécessaire d'effectuer une sélection de l'une parmi les autres.

1) L'aiguillage ou la sélection : Divergence en OU.



- L'étape 11 devient active si l'étape étant active la condition a devient vr
- L'étape 12 devient active si l'étape l'est et la condition b est vraie.

Si les conditions a et b sont simultanément vraies alors que l'étap 1 est active, on assiste dans le respect des règles aux deux évolutions simultanées de l'étape 11 et de l'étape 12

Table d'activation

	a	b	activation
étape 1 active	0	0	1
	1	0	11
	0	1	12
	1	1	11 et 12

On retrouve là, la possibilité d'un parallélisme qui n'est pas structural mais dont le comportement est identique. Or cette éventualité n'a peut-être pas été prévue par le concepteur. Peut-être même, devait-il la rejeter !

Cette structure d'aiguillage n'est pas exclusive. Pour lever cette ambiguïté on se doit d'agir sur la phrase logique de la réceptivité.

- On peut rendre la divergence en "OU exclusif" en remplaçant les conditions a et b respectivement par $a.\bar{b}$ et $\bar{a}.b$

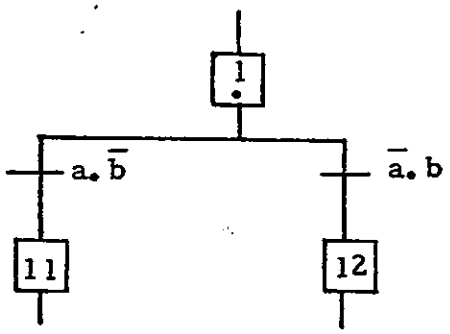


Table d'activation :

	a	b	activation
étape 1 active	0	0	1
	1	0	11
	0	1	12
	1	1	1

- On peut également introduire des priorités.

La figure suivante montre une priorité à la réceptivité a.
La transition 1-11 peut être franchie lorsque a et b sont présents en même temps ; Ce qui n'est pas le cas de la transition 1-12.

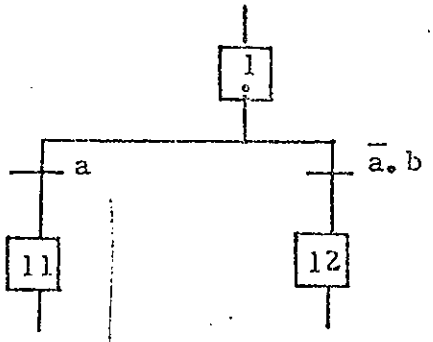


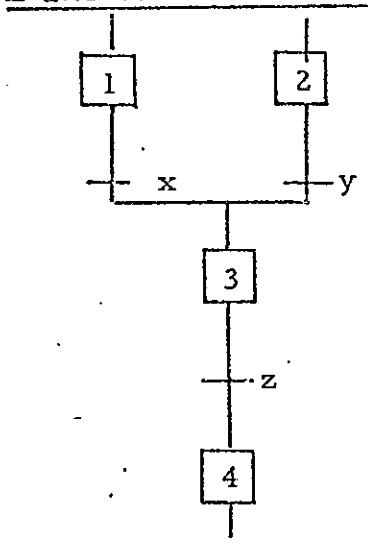
Table d'activation

	a	b	activation
étape 1 active	0	0	1
	1	0	11
	0	1	12
	1	1	11

En conclusion, dans une telle architecture d'aiguillage, les différents chemins sont fonctions de la sémantique attribuée aux réceptivité et non pas du graphisme comme dans le cas du parallélisme structural.

Il est important, pour une meilleure conception que ces notions soient mise en évidence.

2) L'attribution : Convergence en OU.



L'étape 3 devient active si l'étape 1 étant activée la condition x est vraie ou si l'étape 2 étant activée, la condition y est vraie.

On rencontre là une autre forme d'évolutions simultanées. Elles ne sont plus de nature parallèle au sens structure du terme, mais du type "pipeline".

En effet, dans le cas où les étapes 1 et 2 sont simultanément actives ; Si x est vrai, y et z faux alors l'étape 3 est activée et 1 desactivée. Si dans un deuxième temps, y et z deviennent simultanément vrais, l'étape 4 est alors activée et l'étape 3, d'après la règle, n°5 reste active

On constate que le GRAFCET permet un choix considérable d'évolutions simultanées. Mais Monsieur le Concepteur ATTENTION ! Vous devez rester maître de vos propres désirs.

3) Exemple d'aiguillage : Desserte de 3 postes.

Soit un dispositif de manipulation pouvant desservir 3 postes P1, P2 et P3.

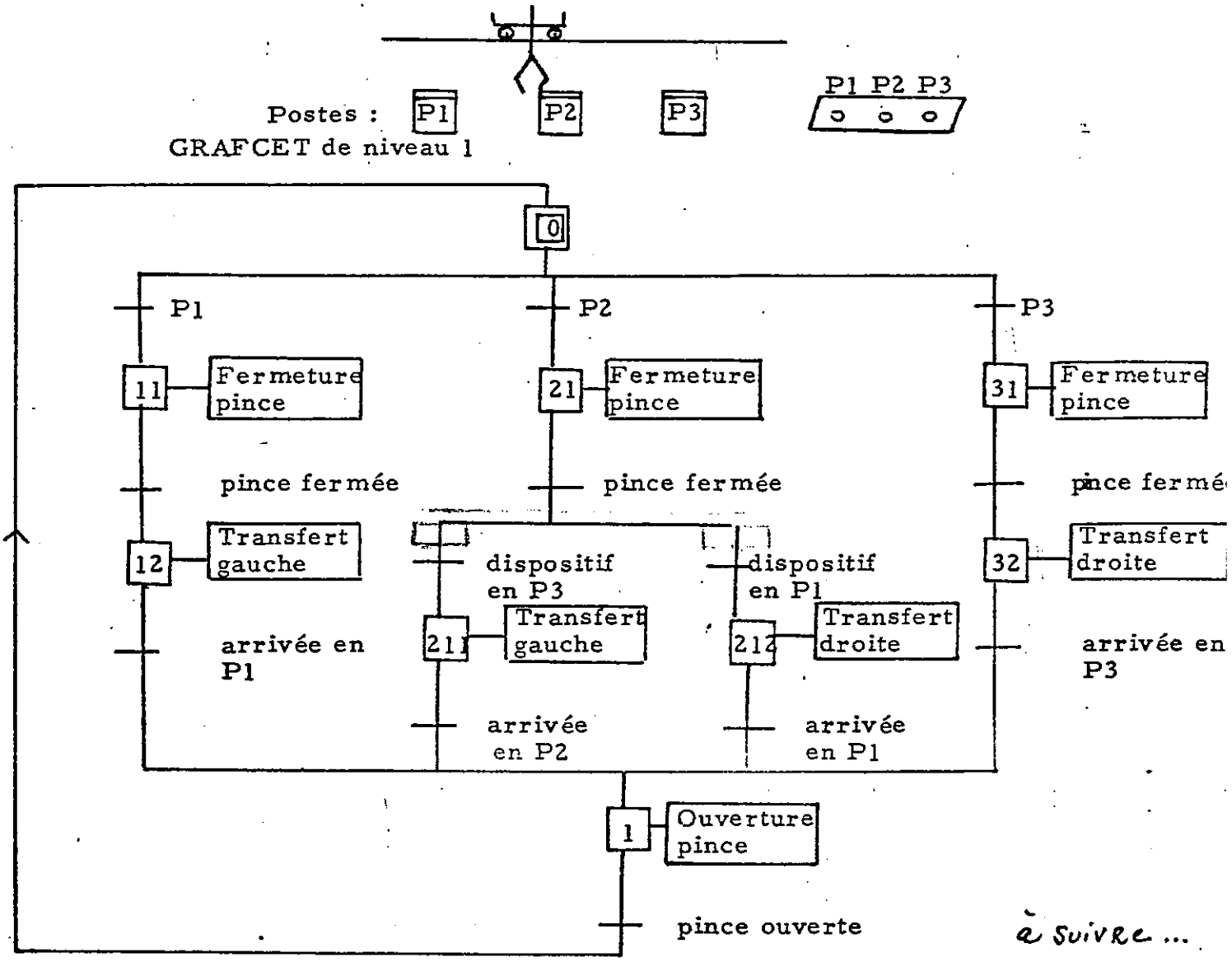
Au repos le dispositif est présent à l'un des 3 postes ; Pince ouverte.

Un pupitre comprend 3 boutons poussoirs correspondant à des demandes de transfert vers l'un des trois postes.

Lorsque le dispositif est au repos, la demande d'un autre poste déclenche la séquence suivante :

- fermeture de la pince
- transfert à gauche ou à droite suivant le cas
- ouverture de la pince dès que le poste désiré est atteint

Un seul appel peut être effectué à la fois.



à suivre ...

BIBLIOGRAPHIE

- LE GRAFCET, diagramme fonctionnel des automatismes séquentiels
  Rapport de l'ADEPA, Avril 1979.

- LE GRAFCET, graphe de commande étape transition
  Bulletin du Lycée Technique d'Alès et du Lycée Technique de Montpellier.

- Comprendre, maîtriser et appliquer le GRAFCET
  M. BLANCHARD : Cepadues édition 1980.

- LE GRAFCET outil de représentation du cahier des charges d'un automatisme logique
  L. TOURRES : l'Ingénieur et le Technicien de l'enseignement Technique n°213, Juillet-Août 1978.

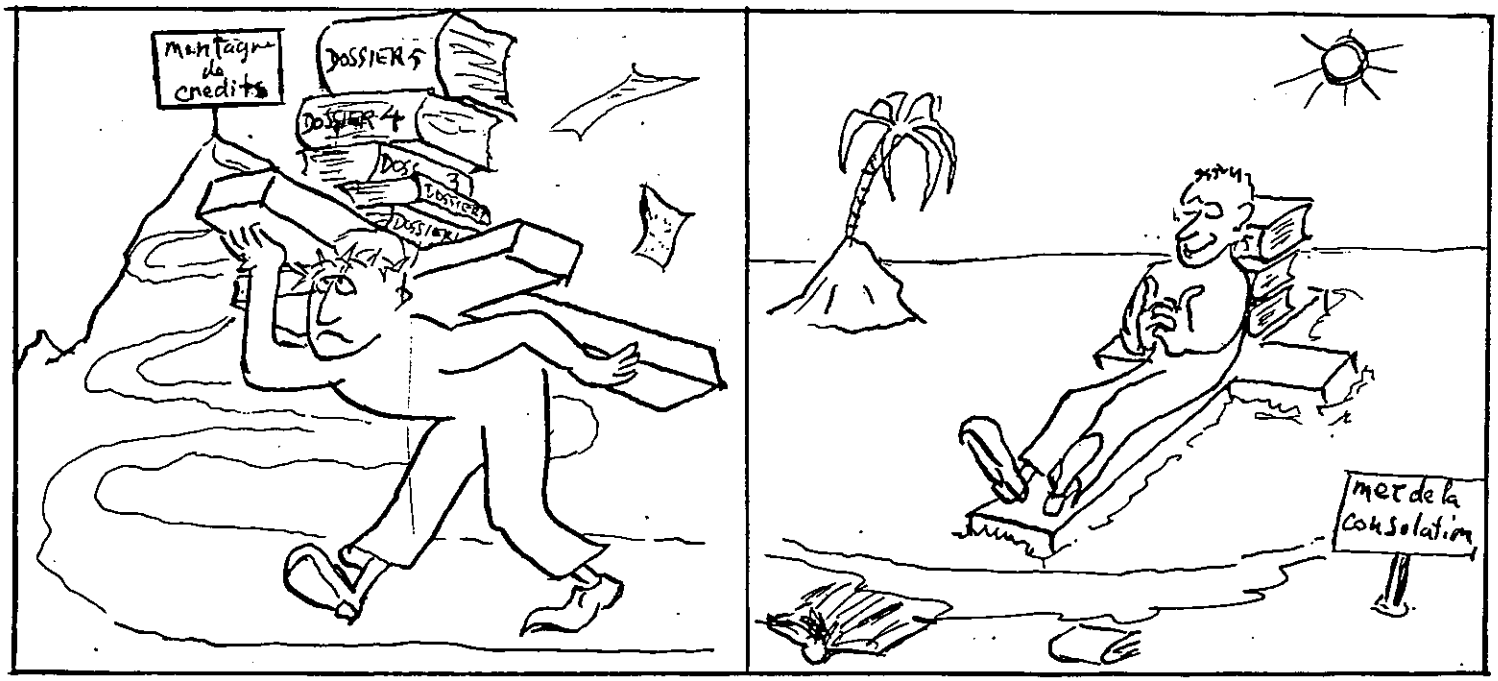
- LE GRAFCET
  L. TOURRES : l'Ingénieur et le Technicien de l'enseignement Technique n°214, Septembre-Octobre 1978.

- Exemple d'utilisation du GRAFCET pour la spécification et la synthèse d'un automatisme. Automatique et Informatique industrielle n°77 Mai 1979.

- LE GRAFCET, sa pratique et ses applications
  J. C. BOSSY, P. BRARD, P. FAUGERE, C. MEKLAUD 1981.

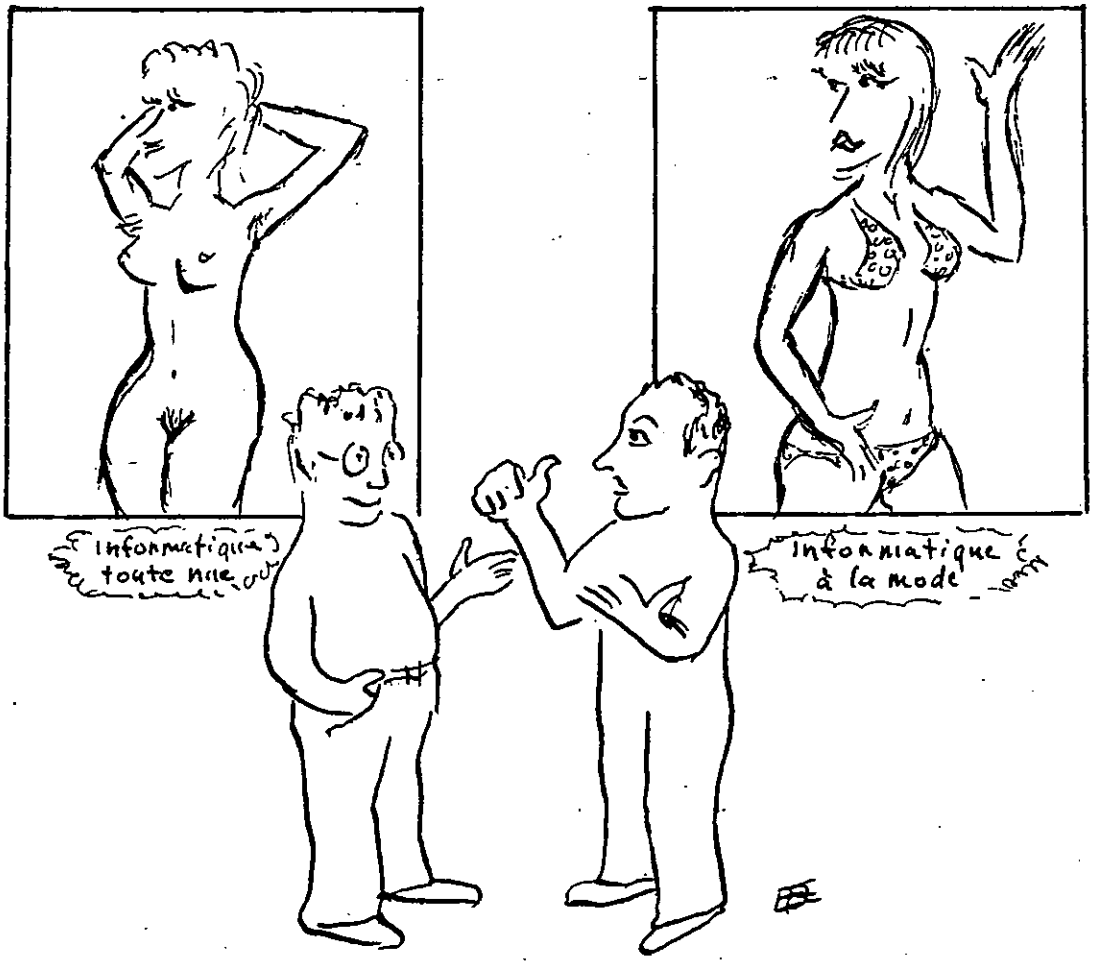
- GRAFCET et logique industrielle programmée
  Eyrolles 1980, S. THELLIEZ et J. M. TOULOTTE

VOUZZAVEDIBISAR.



Informaticien écrasé par ses charges.

Informaticien ayant réussi à surmonter ses problèmes.



- Quel rapport avec l'informatique ?
- Vous vous voyez toute votre vie faire des astuces sur le micro-processeur ?