Une généralisation de la notion d'automate et applications

Michel Depeyrot

THESE présentée à

UNIVERSITE SCIENTIFIQUE ET MEDICALE DE GRENOBLE

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

POUR OBTENIR LE GRADE DE

DOCTEUR D'ETAT ES SCIENCES

Spécialité INFORMATIQUE

Michel DEPEYROT

UNE GENERALISATION DE LA NOTION D'AUTOMATE ET APPLICATIONS

Thèse soutenue le 24 juin 1975 devant la commission d'examen :

Président : J. KUNTZMANN

Examinateurs : L. BOLLIET, P. FAURRE, G. SAUCIER

Rapporteur : P. AZEMA

UNIVERSITE SCIENTIFIQUE ET MEDICALE DE GRENOBLE

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

M. Michel SOUTIF	Présidents	M. Louis NEEL
M. Gabriel CAU	Vice-Présidents	MM. Lucien BONNETAIN, Jean BENOIT

MEMBRES DU CORPS ENSEIGNANT DE L'U.S.M.G.

PROFESSEURS TITULAIRES

MM.	ANGLES D'AURIAC Paul	Mécanique des fluides
	ARNAUD Paul	Chimie
	AUBERT Guy	Physique
	AYANT Yves	Physique approfondie
Mme	BARBIER Marie-Jeanne	Electrochimie
MM.	BARBIER Jean-Claude	Physique expérimentale
	BARBIER Reynold	Géologie appliquée
	BARJON Robert	Physique nucléaire
	BARNOUD Fernand	Biosynthèse de la cellulose
	BARRA Jean-René	Statistiques
	BARRIE Joseph	Clinique chirurgicale
	BEAUDOING André	Clinique de Pédiatrie et Puériculture
	BERNARD Alain	Mathématiques Pures
Mme	BERTRANDIAS Françoise	Mathématiques Pures
MM.	BEZES Henri	Pathologie chirurgicale
	BLAMBERT Maurice	Mathématiques Pures
	BOLLIET Louis	Informatique (IUT B)
	BONNET Georges	Electrotechnique
	BONNET Jean-Louis	Clinique ophtalmologique
	BONNET-EYMARD Joseph	Pathologie médicale
	BOUCHERLE André	Chimie et toxicologie
	BOUCHEZ Robert	Physique nucléaire
	BOUSSARD Jean-Claude	Mathématiques appliquées
	BRAVARD Yves	Géographie
	CABANEL Guy	Clinique rhumatologique et hydrologie
	CALAS François	Anatomie
	CARLIER Georges	Biologie végétale
	CARRAZ Gilbert	Biologie animale et pharmacodynamie
	CAU Gabriel	Médecine légale et toxicologie
	CAUQUIS Georges	Chimie organique
	CHABAUTY Claude	Mathématiques Pures
	CHARACHON Robert	Clinique Oto-Rhino-Laryngologique
	CHATEAU Robert	Thérapeutique (Neurologie)
	CHIBON Pierre	Biologie animale
	COEUR André	Pharmacie chimique et chimie analytique
	CONTAMIN Robert	Clinique gynécologique
	COUDERC Pierre	Anatomie pathologique
	CRAYA Antoine	Mécanique
Mme	DEBELMAS Anne-Marie	Matière médicale
MM.	DEBERMAS Jacques	Géologie générale
	DEGRANGE Charles	Zoologie
	DELORMAS Pierre	Pneumo-Phtisiologie
	DEPORTES Charles	Chimie minérale
	DESRE Pierre	Métallurgie
	DESSAUX Georges	Physiologie animale
	DODU Jacques	Mécanique appliquée

MM.	DOLIQUE Jean-Michel	Physique des plasmas
	DREYFUS Bernard	Thermodynamique
	DRUCROS Pierre	Cristallographie
	DUGOIS Pierre	Clinique de dermatologie et syphiligraph
	FAU René	Clinique neuro-psychiatrique
	GAGNAIRE Didier	Chimie physique
	GALLISSOT François	Mathématiques pures
	GALVANI Octave	Mathématiques pures
	GASTINEL Noël	Mathématiques appliquées
	GAVEND Michel	Pharmacologie
	GEINDRE Michel	Electroradiologie
	GERBER Robert	Mathématiques pures
	GERMAIN Jean-Pierre	Mécanique
	GIRAUD Pierre	Géologie
	JANIN Bernard	Géographie
	KAHANE André	Physique Générale
	KLEIN Joseph	Mathématiques pures
	KOSZUL Jean-Louis	Mathématiques pures
	KRAVTCHENKO Julien	Mécanique
	KUNTZMANN Jean	Mathématiques appliquées
	LACAZE Albert	Thermodynamique
	LACHARME Jean	Biologie végétale
	LAJZEROWICZ Joseph	Physique
	LATREILLE René	Chirurgie générale
	LATURAZE Jean	Biochimie pharmaceutique
	LAURENT Pierre-Jean	Mathématiques appliquées
	LEDRU Jean	Clinique médicale B
	LLIBOUTRY Louis	Géophysique
	LONGEQUEUE Jean-Pierre	Physique nucléaire
	LOUP Jean	Géographie
Mle	LUTZ Elisabeth	Mathématiques pures
	MALGRANGE Bernard	Mathématiques pures
	MALINAS Yves	Clinique obstétricale
	MARTIN-NOEL Pierre	Seméiologie médicale
	MAZARE Yves	Clinique médicale A
	MICHEL Robert	Minéralogie et pétrographie
	MICOUD Max	Clinique maladies infectieuses
	MOURIQUAND Claude	Histologie
	MOUSSA André	Chimie nucléaire
	MULLER Jean-Michel	Thérapeutique (néphrologie)
	NEEL Louis	Physique du solide
	OZENDA Paul	Botanique
	PAYAN Jean-Jacques	Mathématiques pures
	PEBAY-PEYROULA Jean-Claude	Physique
	RASSAT André	Chimie systématique
	RENARD Michel	Thermodynamique
	RINALDI Renaud	Physique
	DE ROUGEMONT Jacques	Neuro-chirurgie
	SEIGNEURIN Raymond	Microbiologie et hygiène
	SENGEL Philippe	Zoologie
	SIBILLE Robert	Construction mécanique
	SOUTIF Michel	Physique générale
	TANCHE Maurice	Physiologie
	TRAYNARD Philippe	Chimie générale
	VAILLANT François	Zoologie
	VALENTIN Jacques	Physique nucléaire
	VAUQUOIS Bernard	Calcul électronique
Mme	VERAIN Alice	Pharmacie galénique
MM.	VERAIN André	Physique
	VEYRET Paul	Géographie
	VIGNAIS Pierre	Biochimie médicale
	YOCCOZ Jean	Physique nucléaire théorique

PROFESSEURS ASSOCIES

MM.	CHEEKE John	Thermodynamique
	COPPENS Philip	Physique
	CORCOS Gilles	Mécanique
	CRABBE Pierre	CERMO
	GILLESPIE John	I.S.N.
	ROCKAFELLAR Ralph	Mathématiques appliquées

PROFESSEURS SANS CHAIRE

Mlle	AGNIUS-DELORD Claudine	Physique pharmaceutique
	ALARY Josette	Chimie analytique
MM.	AMBROISE-THOMAS Pierre	Parasitologie
	BELORIZKY Elie	Physique
	BENZAKEN Claude	Mathématiques appliquées
	BERTRANDIAS Jean-Paul	Mathématiques pures
	BIAREZ Jean-Pierre	Mécanique
	BILLET Jean	Géographie
Mme	BONNIER Jane	Chimie générale
MM.	BOUCHET Yves	Anatomie
	BRUGEL Lucien	Energétique
	CONTE René	Physique
	DEPASSEL Roger	Mécanique des fluides
	GAUTHIER Yves	Sciences biologiques
	GAUTRON René	Chimie
	GIDON Paul	Géologie et Minéralogie
	GLENAT René	Chimie organique
	GROULADE Joseph	Biochimie médicale
	HACQUES Gérard	Calcul numérique
	HOLLARD Daniel	Hématologie
	HUGONOT Robert	Hygiène et Méd. Préventive
	IDELMAN Simon	Physiologie animale
	JOLY Jean-René	Mathématiques pures
	JULLIEN Pierre	Mathématiques appliquées
Mme	KAHANE Josette	Physique
MM.	KUHN Gérard	Physique
	LOISEAUX Jean	Physique nucléaire
	LUU-DUC-Cuong	Chimie organique
	MAYNARD Roger	Physique du solide
	PELMONT Jean	Biochimie
	PERRIAUX Jean-Jacques	Géologie et minéralogie
	PFISTER Jean-Claude	Physique du solide
Mlle	PIERY Yvette	Physiologie animale
MM.	RAYNAUD Hervé	Mathématiques appliquées
	REBECQ Jacques	Biologie (CUS)
	REVOL Michel	Urologie
	REYMOND Jean-Charles	Chirurgie générale
	RICHARD Lucien	Biologie végétale
Mme	RINAUDO Marguerite	Chimie macromoléculaire
MM.	ROBERT André	Chimie papetière
	SARRAZIN Roger	Anatomie et chirurgie
	SARROT-REYNAULD Jean	Géologie
	SIROT Louis	Chirurgie générale
Mme	SOUTIF Jeanne	Physique générale
MM.	VIALON Pierre	Géologie
	VAN CUTSEM Bernard	Mathématiques appliquées

MAITRES DE CONFERENCES ET MAITRES DE CONFERENCES AGREGES

MM.	AMBLARD Pierre	Dermatologie
	ARMAND Gilbert	Géographie
	ARMAND Yves	Chimie
	BARGE Michel	Neurochirurgie
	BEGUIN Claude	Chimie organique
Mme	BERIEL Hélène	Pharmacodynamique
M.	BOUCHARLAT Jacques	Psychiatrie adultes
Mme	BOUCHE Liane	Mathématiques (CUS)
MM.	BRODEAU François	Mathématiques (IUT B)
	BUISSON Roger	Physique
	BUTEL Jean	Orthopédie
	CHAMBAZ Edmond	Biochimie médicale
	CHAMPETIER Jean	Anatomie et organogénèse
	CHARDON Michel	Géographie
	CHERADAME Hervé	Chimie papetière
	CHIAVERINA Jean	Biologie appliquée (EFP)
	COHEN-ADDAD Jean-Pierre	Spectrométrie physique
	COLOMB Maurice	Biochimie médicale
	CORDONNIER Daniel	Néphrologie
	COULOMB Max	Radiologie
	CROUZET Guy	Radiologie
	CYROT Michel	Physique du solide
	DELOBEL Claude	M.I.A.G.
	DENIS Bernard	Cardiologie
	DOUCE Roland	Physiologie végétale
	DUSSAUD René	Mathématiques (CUS)
Mme	ETERRADOSSI Jacqueline	Physiologie
MM.	FAURE Jacques	Médecine légale
	FONTAINE Jean-Marc	Mathématiques pures
	GAUTIER Robert	Chirurgie générale
	GENSAC Pierre	Botanique
	GIDON Maurice	Géologie
	GRIFFITHS Michaël	Mathématiques appliquées
	GROS Yves	Physique (stag.)
	GUITTON Jacques	Chimie
	HICTER Pierre	Chimie
	IVANES Marcel	Electricité
	JALBERT Pierre	Histologie
	KOLODIE Lucien	Hématologie
	KRAKOWIAK Sacha	Mathématiques appliquées
Mme	LAJZEROWICZ Jeannine	Physique
MM.	LEROY Philippe	Mathématiques
	MACHE Régis	Physiologie végétale
	MAGNIN Robert	Hygiène et médecine préventive
	MARECHAL Jean	Mécanique
	MARTIN-BOUYER Michel	Chimie (CUS)
	MICHOULIER Jean	Physique (IUT A)
Mme	MINIER Colette	Physique
MM.	NEGRE Robert	Mécanique
	NEMOZ Alain	Thermodynamique
	PARAMELLE Bernard	Pneumologie
	PECCOUD François	Analyse (IUT B)
	PEFFEN René	Métallurgie
	PERRET Jean	Neurologie
	PHELIP Xavier	Rhumatologie
	RACHAIL Michel	Médecine interne
	RACINET Claude	Gynécologie et obstétrique
	RAMBAUD Pierre	Pédiatrie
Mme	RENAUDET Jacqueline	Bactériologie
MM.	ROBERT Jean-Bernard	Chimie-Physique

MM.	ROMIER Guy	Mathématiques (IUT B)
	SHOM Jean-Claude	Chimie générale
	STIEGLITZ Paul	Anesthésiologie
	STOEBNER Pierre	Anatomie pathologique
	VROUSOS Constantin	Radiologie

MAITRES DE CONFERENCES ASSOCIES

MM.	COLE Antony	Sciences nucléaires
	FORELL César	Mécanique
	MOORSANI Kishin	Physique

CHARGES DE FONCTIONS DE MAITRES DE CONFERENCES

MM.	BOST Michel	Pédiatrie
	CONTAMIN Charles	Chirurgie thoracique et cardio-vasculaire
	FAURE Gilbert	Urologie
	MALLION Jean-Michel	Médecine du travail
	ROCHAT Jacques	Hygiène et hydrologie

Fait à Saint Martin d'Hères, OCTOBRE 1974.

"MEMBRES DU CORPS ENSEIGNANT DE L'I.N.P.G."

PROFESSEURS TITULAIRES

MM.	BENOIT Jean	Radioélectricité
	BESSON Jean	ELectrochimie
	BONNETAIN Lucien	Chimie Minérale
	BONNIER Etienne	Electrochimie, Electrométallurgie
	BRISSONNEAU Pierre	Physique du solide
	BUYLE-BODIN Maurice	ELectronique
	COUMES André	Radioélectricité
	FELICI Noël	Electrostatique
	PAUTHENET René	Physique du solide
	PERRET René	Servomécanismes
	SANTON Lucien	Mécanique
	SILBER Robert	Mécanique des fluides

PROFESSEUR ASSOCIE

M.	BOUDOURIS Georges	Radioélectricité

PROFESSEURS SANS CHAIRE

MM.	BLIMAN Samuel	Electronique
	BLOCH Daniel	Physique du solide et cristallographie
	COHEN Joseph	Electrotechnique
	DURAND Françis	Metallurgie
	MOREAU René	Mécanique
	POLOUJADOFF Michel	ELectrotechnique
	VEILLON Gérard	Informatique fondamentale et appliquée
	ZADWORNY François	Electronique

MAITRES DE CONFERENCES

MM.	BOUVARD Maurice	Génie mécanique
	CHARTIER Germain	Electronique
	FOULARD Claude	Automatique
	GUYOT Pierre	Chimie minérale
	JOUBERT Jean Claude	Physique du solide
	LACOUME Jean Louis	Géophisique
	LANCIA Roland	Physique atomique
	LESPINARD Georges	Mécanique
	MORET Roger	Electrotechnique nucléaire
	ROBERT François	Analyse numérique
	SABONNADIERE Jean Claude	Informatique fondamentale et appliquée
Mme	SAUCIER Gabrièle	Informatique fondamentale et appliquée

MAITRE DE CONFERENCES ASSOCIE

M.	LANDAU Ioan Doré	Automatique

CHARGE DE FONCTIONS DE MAITRES DE CONFERENCES

M.	ANCEAU François	Mathématiques appliquées

Fait à Saint Martin d'Hères JANVIER 1974

PREFACE

L'objet de cette thèse est la "Conception Structurée" des Systèmes Informatiques, domaine trop vaste et doté de facettes trop diverses pour permettre une dissertation concise, de sorte que le texte présenté n'est l'exposé que d'une facette parmi bien d'autres, celle des applications d'un modèle dotant les machines séquentielles d'une certaine sémantique. Cette dissertation fait donc appel à des connaissances linguistiques, fonctionnelles, architecturales et algébriques regroupées selon un point de vue qui se démarque des Mathématiques Appliquées pour tenter d'être représentatif de l'approche informatique. Ceci explique la satisfaction de l'auteur que son jury soit précisément composé de spécialistes reconnus de la Science de la programmation, de la théorie des langages, de l'architecture des machines. de la modélisation des automates et de l'Automatisme.

Que le Professeur Kuntzmann ait bien voulu assurer la présidence du jury de cette thèse est pour l'auteur un honneur et une chance pour lesquels il tient à exprimer sa profonde gratitude. Les travaux de recherche dirigés à Grenoble par le Pr. Kuntzmann ont eu en effet une importante influence sur cette dissertation.

L'auteur tient à remercier particulièrement le Professeur L. Bolliet qui pendant plus de quatre ans l'a incité, encouragé et orienté pour mener à bien cette thèse. Le Professeur C.G. Bell de Carnegie-Mellon et Vice-président d'Ingénierie de DEC a été pour l'auteur l'exemple de la vraie "Universalité", celle qui assume les problèmes industriels avec rigueur et les problèmes universitaires avec réalisme.

Cette dissertation est pour l'auteur l'occasion d'exprimer à P. Faurre une reconnaissance qui porte sur sept années de coopération qu'il a marquées par son exemple et ses encouragements. Comme en témoignent les textes indiqués en référence, cette thèse est imprégnée de travaux et de publications effectués en commun.

La tâche la plus ardue, de rapporteur, qu'a bien voulu assumer M. Bétourné l'a amené à faire refondre et alléger considérablement le texte de cette thèse. L'auteur lui en sait gré car cette démarche était nécessaire pour rendre les résultats obtenus accessibles à la majorité des chercheurs intéressés. Après tout, publier une thèse, c'est un peu payer son dû à la communauté scientifique pour ce que l'on a reçu d'elle.

De plus, ce travail a été accompli au long de plusieurs années durant lesquelles l'auteur a pu dans le cadre de la CII et de l'IRIA bénéficier de discussions et collaborations avec de multiples ingénieurs et chercheurs sur des sujets proches. Qu'ils en soient tous remerciés et en particulier R. Bavoux, D. Delpuech, A. Jorry et C. Masson.

Enfin, l'auteur est reconnaissant au Professeur Lions de ses conseils et encouragements constants et de l'environnement de recherche qu'il lui a procuré dans le cadre du LABORIA.

La dactylographie de cet ouvrage est l'oeuvre de Mme Barny dont les lecteurs peuvent apprécier la qualité de la présentation.

Versailles, Octobre 1974.

TABLE DES MATIERES

SOMMAIRE

Partant de l'idée que le relatif insuccès des techniques de machines séquentielles formalisées par des automates tient à l'absence de sémantique dans les modèles algébriques utilisés, cette dissertation présente une généralisation du concept d'automate qui permet de distinguer deux classes d'information en entrée : les commandes et les données.

Après une présentation des limitations concrètes des théories actuelles, le modèle de polymate est exposé et ses propriétés abstraites sont détaillées jusqu'à permettre une évaluation de complexité de son identification en temps réel

Les modèles d'algorithmes proposés par TURING et WANG sont alors présentés et composés de manière à mettre en évidence le modèle de polymate tout en dégageant le schéma classique des calculateurs de type VON NEUMANN, et en dégageant une formalisation du parallélisme de fonctionnement.

Les résultats essentiels s'ensuivent :

- une formalisation du langage PMS de description de calculateurs, lui donnant une sémantique réaliste en termes de polymates, dépassant ainsi les limitations précédemment décrites des théories actuelles.

- une technique dite duale d'assignation des états permettant le remplacement de la microprogrammation ou de la diagrammation dans les circuits intégrés à grande échelle par une logique séquentielle universelle programmable sans diodes et bien adaptée au fonctionnement parallèle.

- un modèle formel des machines séquentielles à transferts de registres mettant en évidence une relation de dualité avec les machines séquentielles à transitions d'états.

CHAPITRE I

CHAPITRE I

GENESE ET ORIENTATION DE CETTE THESE.

I.1.EVOLUTION DE LA CYBERNETIQUE

De nombreux systèmes formels permettant la représentation des processus de traitement de l'information ont été développés dans le deuxième tiers de ce siècle :

-Systèmes normaux de POST,
-Machines universelles de TURING,
-Opérateur LAMBDA de CHURCH,
-Algorithmes normaux de MARKOV,
-Langages procéduriels comme ALGOL,
-Langages fonctionnels comme LISP,
-Réseaux de PETRI et de HOLT,
-Théories de fonctions récursives,
-Shémas de programmes de IANOV,KOLMOGOROV,KARP et MILLER, RUTLEDGE,FLOYD,etc
-Modèles sémantiques de STRACHEY et SCOTT,
-Graphes hiérarchisés de PRATT,
-Graphes pondérés de MARTIN et ESTRIN,
-Structures de calcul asynchrone de MULLER, LUCONI, BRUNO et ALTMAN, etc...
-Processeurs discrets de GLUSHKOV et LETICHEVSKII,
-etc...

Cette multiplicité de modèles répond bien sûr à la diversité des questions abordées, mais elle révèle un processus historique de divergence : ces théories algorithmiques se détachent en effet du cadre initial de la logique mathématique pour se développer en symbiose avec l'Architecture des Systèmes Informatiques, c'est à dire se dégagent de la Science pure pour se rapprocher d'une Technique récente.

Ainsi les questions initiales d'existence d'un algorithme universel, des propriétés des classes d'algorithmes, de leurs relations d'équivalence et de leurs lois de composition, ont fait place aux questions de complexité et d'optimalité structurales (minimalité, cyclicité, déterminisme, parallélisme) permettant de guider la conception des systèmes de programmes et des calculateurs-mêmes, puis permettant leur simulation, l'évaluation de leurs performances ainsi que l'optimisation de leur exploitation (allocation des ressources, ordonnancement des processus parallèles).

Les bases mathématiques actuellement connues pour une approche "structurale" des problèmes sont nécessairement des théories algébriques ; il est d'ailleurs remarquable que l'accroissement de la demande de résultats dans ces domaines ait correspondu historiquement à l'émergence rapide d'Ecoles Mathématiques appropriées comme celle de la théorie des catégories, celles des théories syntaxiques et combinatoires et même du Bourbakisme. Cependant, alors que la démarche des "Mathématiques Appliquées" permet de riches développements (Théories développées autour de modèles donnés), la démarche transposée en Informatique de la Physique expérimentale ou de la Biologie de Synthèse (Théorie des systèmes à partir d'un phénomène réel) se butte à l'immaturité de l'objet-même de cette recherche : le système informatique. En effet, il a fallu attendre le langage PMS pour que quelques concepts Architecturaux soit "fixés" fonctionnellement, tandis que l'évolution rapide de la technologie ne permet pas d'espérer une stabilisation à court-terme ni de la structure des machines, ni de la structure des programmes. En ce sens, la caractéristique de cette science (appelée actuellement "Cybernétique") est de porter sur des objets expérimentables (comme la "Physique") mais de synthèse humaine (comme la "mathématique") et d'un haut degré de complexité (comme la "biologie").

Il y a donc une motivation spéciale en architecture de systèmes informatiques (comme en chimie organique de synthèse) pour développer des modèles indépendamment du rapport "Nombre de théorèmes/Nombre de définitions" qu'ils engendrent. Cette motivation est d'aider à franchir le mur de la complexité et d'atteindre un niveau synthétique d'intuition des phénomènes essentiels impliqués, tout en dépassant autant que faire se peut le caractère temporaire de l'expérience technologique actuelle.

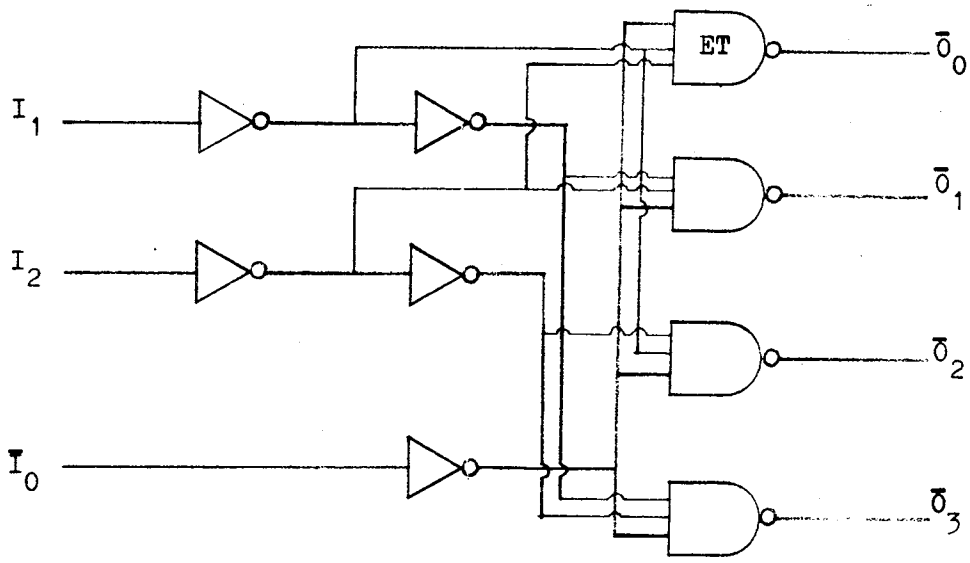
I.2. INSUFFISANCES DE LA NOTION D'AUTOMATE.

La notion d'automate remonte maintenant à plus d'une vingtaine d'années et force est de constater que dans la pratique le mot est plus utilisé que les théorèmes qui ont été développés autour. Il semble que ce relatif insuccès doive être attribué à des causes techniques et peut-être sont-elles pour l'essentiel dans l'absence de sémantique caractérisant les modèles algébriques utilisés ou encore dans l'excessive simplicité du concept.

I.2.1. Sur la distinction entre les signaux de commande et les données.

Considérons le circuit intégré suivant, disponible commercialement à raison de deux fois ce circuit dans un boitier unique à seize connexions.
Il ne s'agit même pas d'un circuit séquentiel puisque sa fonction est combinatoire.

Figure I.1. Circuit logique ST*.



Le symbole o indique une NEGATION, le triangle une amplification.

La table de vérité en logique positive (HIGH = 1, LOW = 0) est la suivante :

$\bar{I}_0$	I_1	I_2	$\bar{O}_0$	$\bar{O}_1$	$\bar{O}_2$	$\bar{O}_3$
1	X	X	1	1	1	1
0	0	0	0	1	1	1
0	1	0	1	0	1	1
0	0	1	1	1	0	1
0	1	1	1	1	1	0

X = indéterminé (0 ou 1)

Il importe maintenant de connaître la sémantique de ce circuit afin de pouvoir s'en servir adéquatement. Pour cela il faut effectuer une partition sur les signaux d'entrée, et considérer que deux classes d'information - les informations de <u>commande</u> I_K et les informations de <u>donnée</u> I_D - peuvent apparaître.

La lecture de la table de vérité invite à traiter I_1 et I_2 comme un tout et à faire jouer à I_0 un rôle spécial.
Parmi les quatre cas possibles :

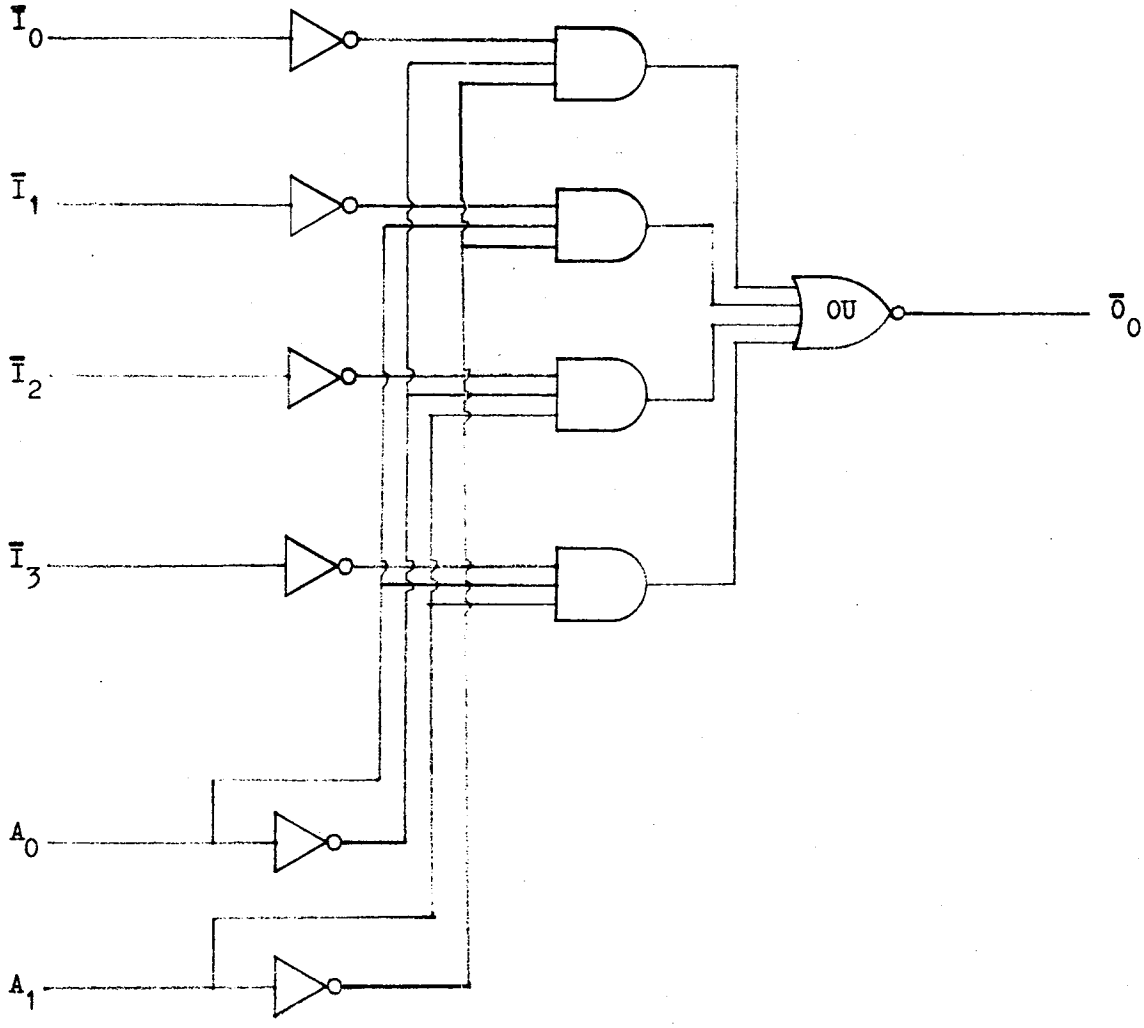
I_0	(I_1, I_2)
I_D	I_D
I_K	I_K
I_D	I_K
I_K	I_D

les deux premiers sont équivalents et sans intérêt puisque précisément ils effacent la distinction recherchée.

Soit maintenant le cas où $\bar{I}_0$ est une donnée traitée sous commande de (I_1, I_2). Il y a quatre commandes possibles qui indiquent l'<u>adresse</u> de routage du bit $\bar{I}_0$ parmi les quatre voies de multiplexage possibles (0, 1, 2, 3).
Le circuit est alors interprété comme un DEMULTIPLEXEUR noté S^{-1}.
Son inverse est un MULTIPLEXEUR S dont le schéma suit.

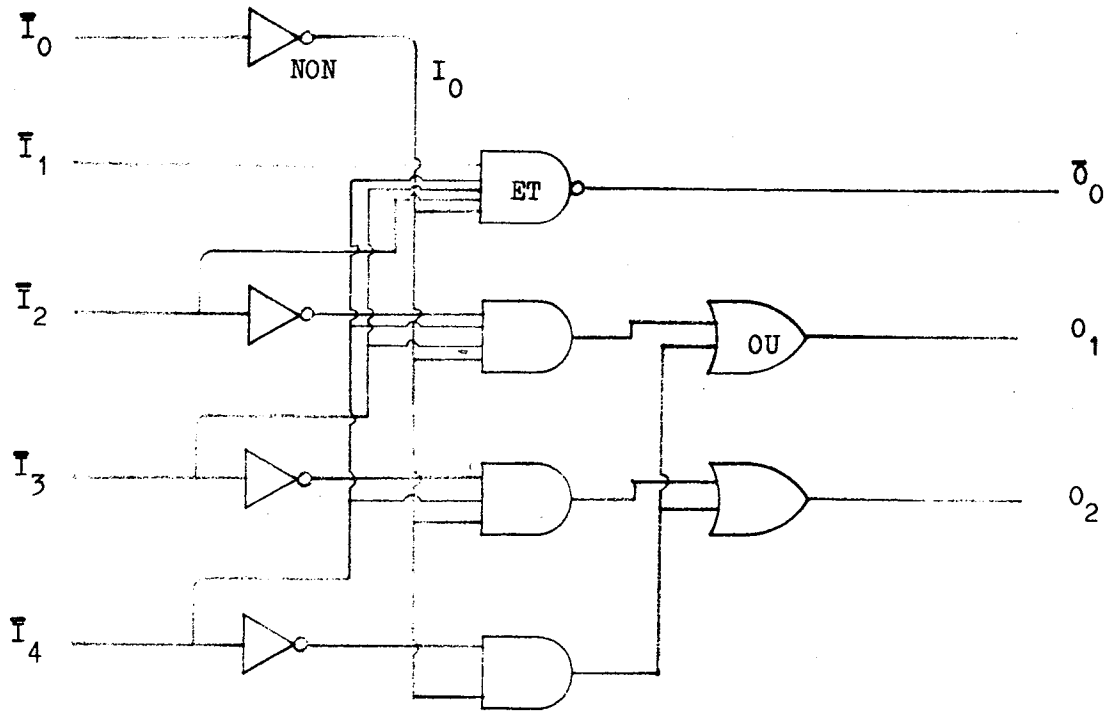
<u>Note</u> : A l'introduction d'un terme nouveau par une définition ce terme est souligné. La notation associée est le symbole $\triangleq$ d'égalité par définition à oppos à ceux d'équivalence ($\equiv$), d'égalité par déduction ($=$) et d'assignation de valeu ($::=$). Le symbole $\oplus$ signifie selon le contexte OU-EXCLUSIF ou somme directe.

Figure I.2. Circuit logique du multiplexeur S.



Soit maintenant le cas où $\bar{I}_0$ dans le circuit de la figure I.1 est une commande opérant sur la donnée (I_1, I_2). La commande unique est une activation ("enable") de transcodage prenant une représentation compacte pondérée 2-1 et la transformant en une représentation primitive (1 parmi 4). Le circuit est ainsi interprété comme un DECODEUR noté T^{-1}. Son inverse est un CODEUR T dont le schéma suit.

Figure I.3. Circuit logique du codeur T.
Le signal de plus fort poids ($\bar{I}_4$) reçoit la plus haute priorité.
Le signal $\bar{O}_0$ indique la présence ou l'absence de données d'entrée lorsque I_0



Le circuit de la figure I.1 peut donc être "interprété" de deux manières, soit comme un décodeur T^{-1}, soit comme un démultiplexeur S^{-1}. Concrètement, le logicien sera amené à utiliser différemment les connexions de ce circuit selon la signification qu'il veut lui donner. Or, le propre des théorie des Automates était de se borner à leurs aspects syntaxiques, alors que la différence entre S^{-1} et T^{-1} est sémantique. L'analyse de cette différence requiert la distinction entre deux types d'information (I_K et I_D) ; pour l'instant nous appellerons informellement dualité la relation entre I_K et I_D et donc celle ent le multiplexage S et le codage T (respectivement associés à la gestion du temps et de l'espace).

I.2.2. Sur la distinction entre les systèmes de commande et les opérateurs.

Mais ici ne s'arrêtent pas les limitations gênantes dans la pratique du concept d'automates. Le besoin se fait sentir aussi de distinguer entre les machines séquentielles de commande K implémentant des algorithmes et celles de

transformation des données supportant des opérateurs logiques D. Ces dernières D sont faites d'un certain nombre de registres, c'est-à-dire de séries de bascules (organe de mémorisation de un bit), enregistrant chacune la représentation codée-binaire d'une quantité standard d'information -mot, octet, etc.- Le nombre d'états est donc considérable, mais le raisonnement est effectué au niveau des registres-mêmes dont la structure spatialement itérative facilite la conception. Par contre, les machines séquentielles de commande K sont de structure hautement irrégulière, ce qui amène souvent à leur affecter un organe de mémorisation par état. C'est pour celles-ci que semble avoir été conçue la théorie des automates, mais comme ces deux types d'automates sont amenés à travailler de concert, interconnectés de diverses manières, le besoin apparaît d'une approche permettant d'analyser conjointement les processeurs plus complexes comprenant des K et des D.

Il est raisonnable de considérer que cette distinction sémantique entre commandes et données introduit la frontière entre l'Automatique et l'Informatique.

Cette dissertation a donc pour but de présenter une généralisation du concept d'automate comme modèle de machine séquentielle, puis de démontrer son intérêt :

- d'une part par rapport aux modèles d'Informatique Théorique (TURING, WANG), tels que présentés dans [ARB.69],
- d'autre part par rapport aux besoins technologiques des calculateurs actuels, tels qu'il apparaissent dans [BEL.71].

Des indications de références, insérées dans le corps du texte, renvoient le lecteur vers divers autres développements ne faisant pas l'objet de la présente thèse, mais qui lui sont suffisamment liés pour mériter l'attention d'un lecteur désireux de poursuivre des recherches sur la voie présentée ici.

I.3. INTRODUCTION A LA LOGIQUE SEQUENTIELLE.

L'affectation d'un organe de mémorisation par état ou par sous-ensemble de l'espace d'état d'une machine séquentielle K est appelée "assignation des états". Chaque méthode d'assignation est fondée sur le choix d'un type d'organe de mémorisation et d'un mode de synchronisation ; chaque méthode entraîne alors une certaine structure d'interconnexion. Ce paragraphe est destiné à rappeler ces connaissances de base ainsi que leur pendant pour la structure des registres dans les opérateurs D.

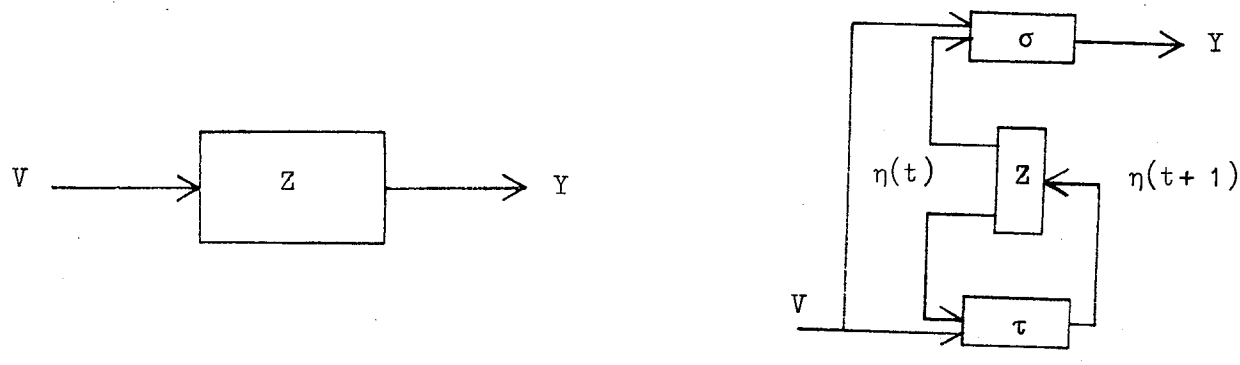
I.3.1. Concepts élémentaires sur les automates.

Un automate est formellement décrit [HAR.66] comme un quintuple $\Sigma \triangleq (V,Z,Y,\tau,\sigma)$ symbolisé par la figure I.4, où V est l'ensemble des entrées, Z l'ensemble des états et Y l'ensemble des sorties. Les fonctions τ et σ sont celle de transition $\tau : V \times Z \rightarrow Z$ et de sortie $\sigma : V \times Z \rightarrow Y$.

Afin d'introduire d'emblée les dispositifs qui permettront d'aborder avec clarté les distinctions fondamentales à effectuer sur les automates de ce point de vue interne, il est utile de considérer la définition de Σ comme une restriction de la définition suivante.

Définition. Un automate est un heptuple $K \triangleq (V,Z,Y,T,\rho,\tau,\sigma)$ où V est l'alphabet des entrées, Z est l'espace d'état, Y l'ensemble des sorties et T le temps identi généralement à l'ensemble $\mathbf{Z}$ des entiers.

Figure I.4. Schéma symbolique d'un automate et forme canonique de HUFFMAN [54].



Un automate est dit fini si les trois ensembles V,Z et Y comprennent un nombre fini d'éléments, ce qui est noté : $|V| = q < \infty$, $|Z| = n < \infty$, $|Y| = p <$ nc

Les trois ensembles sont liés par les fonctions de rythme ρ, de transition τ et de sortie σ qui sont définies comme suit :

$\rho : V \times Z \times T \rightarrow V'$, où V'est un alphabet intermédiaire qui sert à introduire la notion de synchronisation, c'est-à-dire la prise en compte du temps dans l'interaction des éléments de V et de Z. Pour cela V' est défini par $V' \subset V^1$ avec $V^1 \triangleq V \cup \Lambda$ où Λ représente l'élément neutre tel que $\tau(\Lambda,.)$ soit la fonction identité sur Z. Ainsi lorsque certaines conditions sur $V \times Z \times T$ ne sont pas satisfaites, la fonction ρ peut modifier le fonctionnement de l'automate K en inhibant ses entrées ("modifier" par comparaison à l'automate restreint Σ obtenu en remplaçant ρ par la fonction-identité sur V).
Grâce à cela on peut noter $v(t)$ l'élément de V considéré à l'entrée de K à l'instan t, puis $\eta(t)$ l'élément de Z considéré comme l'état de K à l'instant t et $y(t)$ l'élément de Y émis par K à l'instant $t \in T$.

$\rho : V \times Z \times T \rightarrow V'$ définit la synchronisation,
$\tau : V' \times Z \rightarrow Z$ définit l'état suivant : $\eta(t+1) = \tau(v(t),\eta(t))$,
$\sigma : V' \times Z \rightarrow Y$ définit la sortie correspondante.

Il est important de bien percevoir que l'alphabet d'entrée est codé sous forme de signaux q_h parvenant simultanément dans une réalisation électronique sur plusieurs lignes d'entrée. En outre, il est souvent utile de considérer alternativement les entrées de V comme des fonctions sur Z et les états de Z comme des fonctions sur V.

Une perception intuitive de la notion d'état d'un système considéré comme un ensemble de composants interconnectés, peut être obtenue de plusieurs points de vue rigoureusement équivalents :

i) l'état est l'ensemble des informations concernant un système à un instant donné, suffisantes pour résumer son comportement passé et nécessaires pour prédire son comportement futur (point de vue Global).

ii) l'état est l'ensemble des informations concernant l'état des composants participant à la réalisation du système (point de vue Local). Cette seconde définition, récursive dans cette approche formelle, est en fait constructive dans une technologie concrète ; par exemple, l'état d'un système électronique est l'ensemble des tensions en chacun de ses points à un instant donné.

I.3.2. Synchronisation par la fonction de rythme.

Tout mécanisme de synchronisation peut être caractérisé par rapport à deux modes de base : le mode fondamental F et le mode hiérarchisé H. Ils correspondent respectivement aux fonctions de rythme représentées sur la figure I.5 :

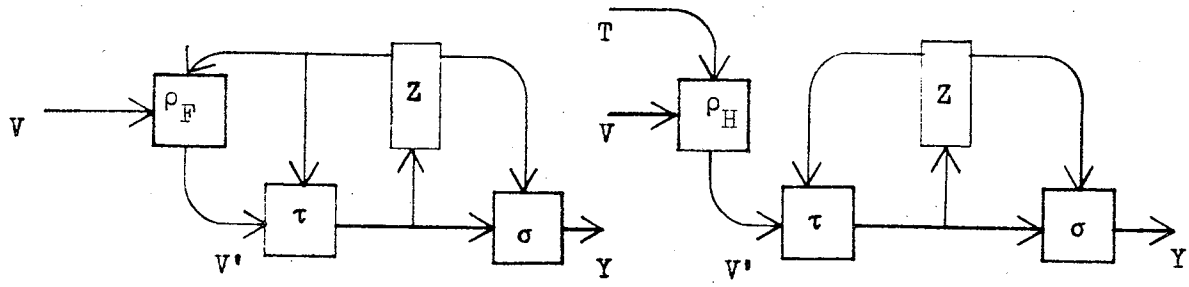


Figure I.5. Les modes F et H.

i) en mode F, ρ_F est la restriction de ρ à $V \times Z$ telle que $\rho_F : V \times Z \to V' \subset V^1$ où, si l'état de l'automate appartient au sous-ensemble Z_S des états dits stables, alors $\rho_F(v,Z_S) \equiv v, \forall v \in V$, tandis que si l'état appartient au complément $\overline{Z}_S \triangleq Z \setminus Z_S$, alors $\rho_F(v,\overline{Z}_S) = \Lambda$, élément neutre tel que $\tau(\Lambda,\eta) \equiv \eta, \forall \eta \in Z_S \subset Z$.

Ce modèle ne peut rendre compte des problèmes d'échantillonnage rencontrés à tout degré de finesse dans l'analyse des machines séquentielles où il faut une échelle de temps T discrète pour toute technologie spécifique, mais ce modèle est ce qu'il y a de commun à toutes les machines séquentielles indépendamment de leurs technologies.

La disparition du temps ici comme entité explicite est compensée par la présence d'une relation d'ordre entre les transitions dans $Z \times Z$ sous contrôle de ce qui permet d'induire la notion d'antériorité d'événements, etc.

Ce mode permet d'organiser le fonctionnement déterministe des machines séquentielles lorsque les signaux d'entrée sont significatifs par niveaux et dits de type L ("level), entraînant des risques d'asynchronisme. Ce problème a été aussi traité en [DEP.1971.e] : "Sur la synchronisation et la fiabilité des automates"

ii) en mode H, ρ_H est la restriction de ρ à $V \times T$ telle que $\rho_H : V \times T \rightarrow V' \subset V^1$ où les signaux d'entrée ne sont significatifs qu'à certains instants définis par une machine séquentielle externe à K, appelée souvent une horloge. Ce mode est employé de préférence lorsque les signaux d'entrée sont de type impulsionnel dit P ("pulse").

Il faut aussi rappeler que physiquement, un mécanisme de synchronisation n'est efficace qu'en probabilité, car le temps est représenté par une "antériorité" entre certains signaux codant les éléments de $V \times Z \times T$ (par exemple entre un signal périodique servant d'horloge et des signaux aléatoires d'entrée) ; or lorsqu'il s'agit de signaux électroniques, les longueurs de fils, la dérive des circuits, etc, ne permettent pas de garantir déterministiquement cette antériorité. Ce thème est développé en [DEP.72] avec le problème de la "Réjection d'Accès".

De manière simplifiée, la synchronisation apparaît maintenant comme une opération logique de OU-EXCLUSIF entre les transitions d'état dans $Z \times Z$ et les changements d'entrée dans $V \times V$. Par la suite, le modèle du quintuple Σ et celui de l'heptuple K peuvent donc être employés équivalemment selon que le problème de la synchronisation est omis ou non.

I.3.3. Bascules de types DATA et TRIGGER et fonctions de transition.

Une bascule est l'automate le plus petit qui soit avec $|Z| = n = 2$. On peut représenter son graphe des états comme indiqué sur la figure I.6.

Définitions. Le graphe G d'un automate K est un ensemble de noeuds en bijection avec les états de K et d'arêtes entre ces noeuds en bijection avec les transitions d'états de K.

Les arêtes sont donc orientées et étiquetées par les entrées commandant les transitions. Selon les cas, il apparaît que les sorties peuvent être associées aux noeuds ou aux arêtes.

Dans l'exemple d'une bascule, les flèches sont étiquetées par les

Si pour chaque état la transition de K est définie pour toute condi
d'entrée de V, le microalgorithme est dit complètement spécifié.

Table des états : le quintuple $\Sigma \triangleq (V,Z,Y,\tau,\sigma)$ peut être représenté tabulairemen
comme indiqué ci-contre.

τ	v_0	v_1	σ
z_0	z_1	z_2	y_0
z_1	z_1	z_2	y_1
z_2	z_2	z_3	y_0
z_3	z_1	z_0	y_1

La forme ici est dite état-sortie car la fonction de sortie σ est indépendante des entrées dans V. La machine de cet exemple sera appelée Σ tout au long de ce texte et ses états $z_i \in Z$.

Les sommets doublement cerclés sur le graphe des états correspondent aux états dont la sortie est y_1.

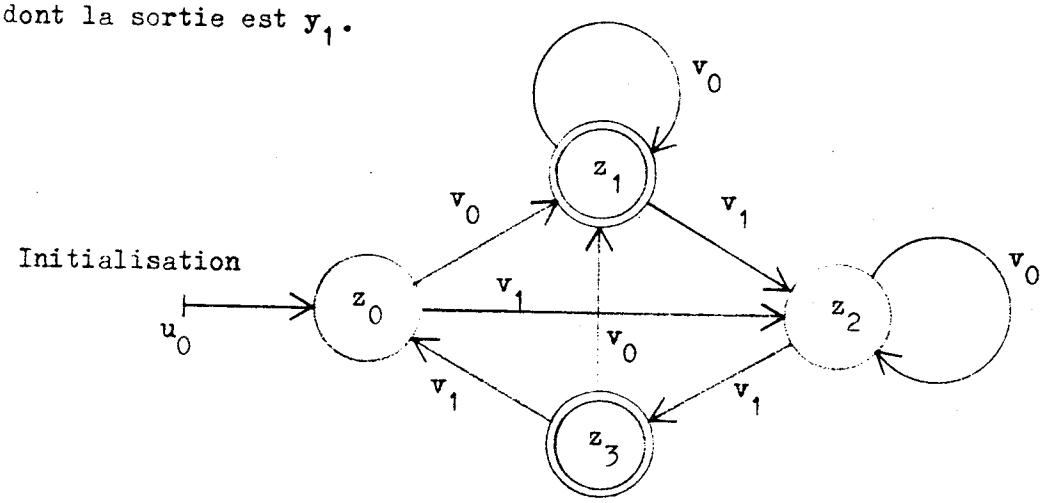


Fig. I.9. Graphe de l'exemple $G(\Sigma)$.

Une assignation des états est dite compacte lorsque le nombre minimal de bascules est employé, c'est-à-dire, pour un algorithme à n états, lorsque le nombre de bascules employé est b tel que

$$b - 1 < \log_2 n \leqslant b \quad \Leftrightarrow \quad 2^{b-1} < n \leqslant 2^b.$$

I.3.4. Fonction de sortie et classification des machines-séquentielles.

On distingue [CAD.59] le type Transition-sortie (Pulse-output : P) tel que la sortie survienne en fonction de la paire (état Z, entrée V) et le type état-sortie (Level-output : L) tel que la sortie survienne en fonction seulement de l'état (Z). Dans le premier cas la sortie dure au plus autant que le signal d'entrée, dans le deuxième temps son niveau logique est maintenu pendant une demi-période d'horloge (H) par une bascule synchrone. Un automate transition-sortie ou état-sortie est dit en forme T ou forme S respectivement.

Transition-Sortie (forme T)		Etat-Sortie (forme S)		
LP	PP	PL	LLC ("clocked")	LL
Générateur [DEP.71.c] Synchrone-Asynchronisé *	Y de la durée de V [MEALY.55] Synchrone	Y de une demi-période [MOORE.56] Asynchrone-Synchronisé*	échantillonnés [GERACE.68]	loquet [HUFFMAN.54] Asynchrone
mode F	mode H		F ∪ H	mode F

Figure I.8. Classification des machines séquentielles.

La classification résultant de l'analyse effectuée dans ce paragraphe non seulement permet de dresser le tableau synthétique de la figure I.8., mais surtout de mettre en évidence un nouveau modèle des machines séquentielles (LP).

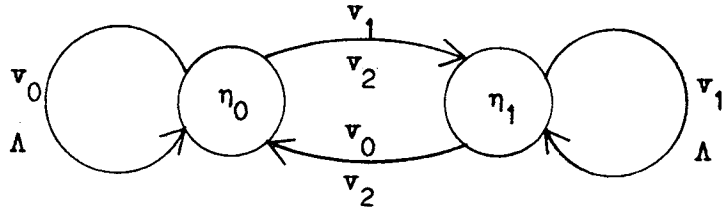
I.3.5. Rappel de diverses techniques d'implémentation d'un automate K.

Soit un automate de commande K destiné à faire exécuter un (micro)-algorithme défini sous forme d'une table des états ou équivalemment sous forme d'un graphe des états G(K), spécifiant une séquence d'actions conditionnelles à effectuer en fonction de valeurs d'entrée. Un modèle d'algorithme est présenté au chapitre III.

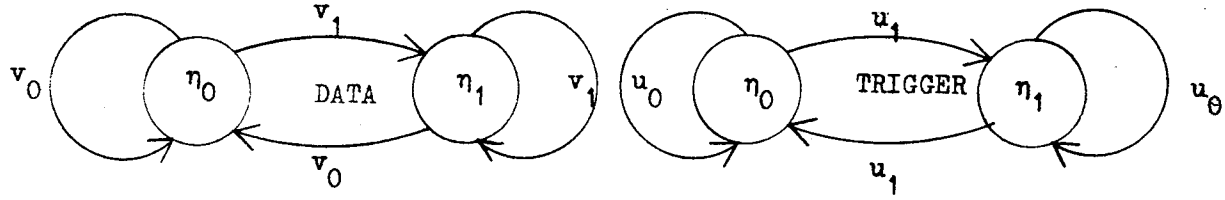
---

*) le premier adjectif porte sur le regime d'entrée (L pour "level", P pour "pulse"), tandis que le second porte sur le fonctionnement interne sans ou avec horloge.

symboles Λ, v_0, v_1, v_2 de l'alphabet d'entrée V le plus complet possible. La bascule est dite <u>complète</u> ou JK.



<u>Figure I.6.</u> Graphe d'une bascule complète (JK).



<u>Figure I.7.</u> Graphes des bascules DATA et TRIGGER.

En effectuant sur V une partition entre les entrées de <u>forçage</u> (RESET) : v_0 et v_1 telles que $\tau(v_i, Z) = \eta_i$, $\forall$ Z, et les entrées de <u>permutation</u> : Λ et v_2 notées respectivement u_0 et $u_1 \in U$ (d'action sur Z inversible), on voit que les fonctions de transition peuvent être classées en type <u>entrée-transition</u> dont l'équation avec Z et $U \equiv \{0,1\}$ peut s'écrire : $\eta(t+1) = \eta(t) \oplus u(t)$, où $\eta \in Z$ et $t \in T$, et en type <u>entrée-état</u> dont l'équation peut s'écrire $\eta(t+1) = v(t)$.

Remarquons qu'une bascule de type DATA n'a pas d'action-identité ou entrée-nulle (ce qu'est u_0 pour le type TRIGGER), mais l'équivalent est obtenu par inhibition du signal d'horloge (Λ).

La <u>table des états</u> d'un automate est une présentation des valeurs de $\tau(v,\eta)$ et $\sigma(v,\eta)$ pour tout $\eta \in Z$ (ligne du tableau) et tout $v \in V'$ (colonne du tableau). Dans le cas d'une bascule :

Z	$\Lambda = u_0$	u_1	v_0	v_1	σ
0	0	1	0	1	0
1	1	0	0	1	1

En pratique les bascules TRIGGER dont les entrées sont de type-permutation ne sont pas adaptées au problème de l'assignation mais plutôt à l'implémentation de compteurs. La figure I.10 montre comment elles peuvent être cependant utilisées pour la réalisation de K. La fonction de rythme ρ est "distribuée" au sein de chacune des bascules composantes M_i.

w_0	w_1	Z
0	0	z_0
0	1	z_1
1	1	z_2
1	0	z_3

Table d'assignation

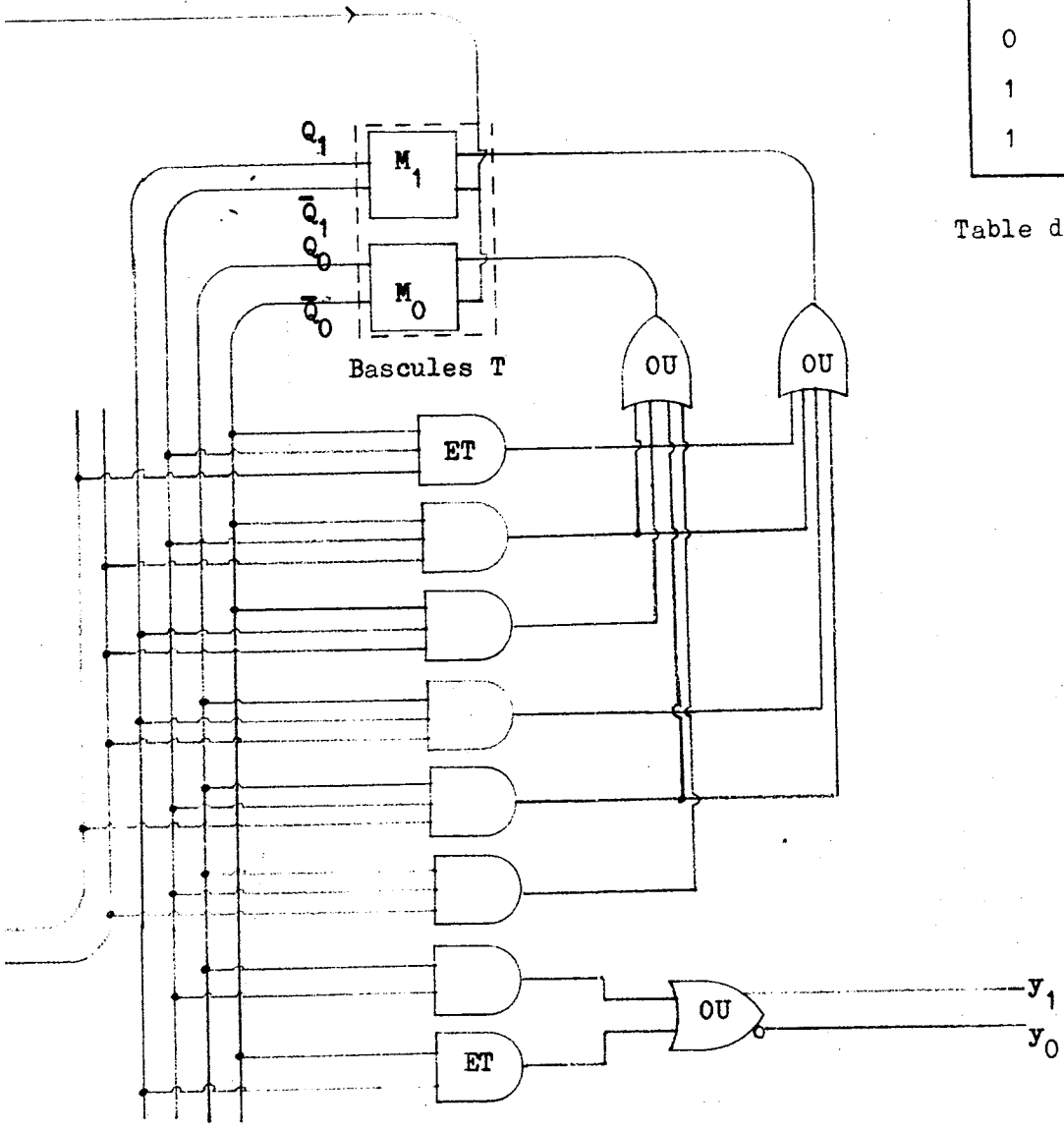
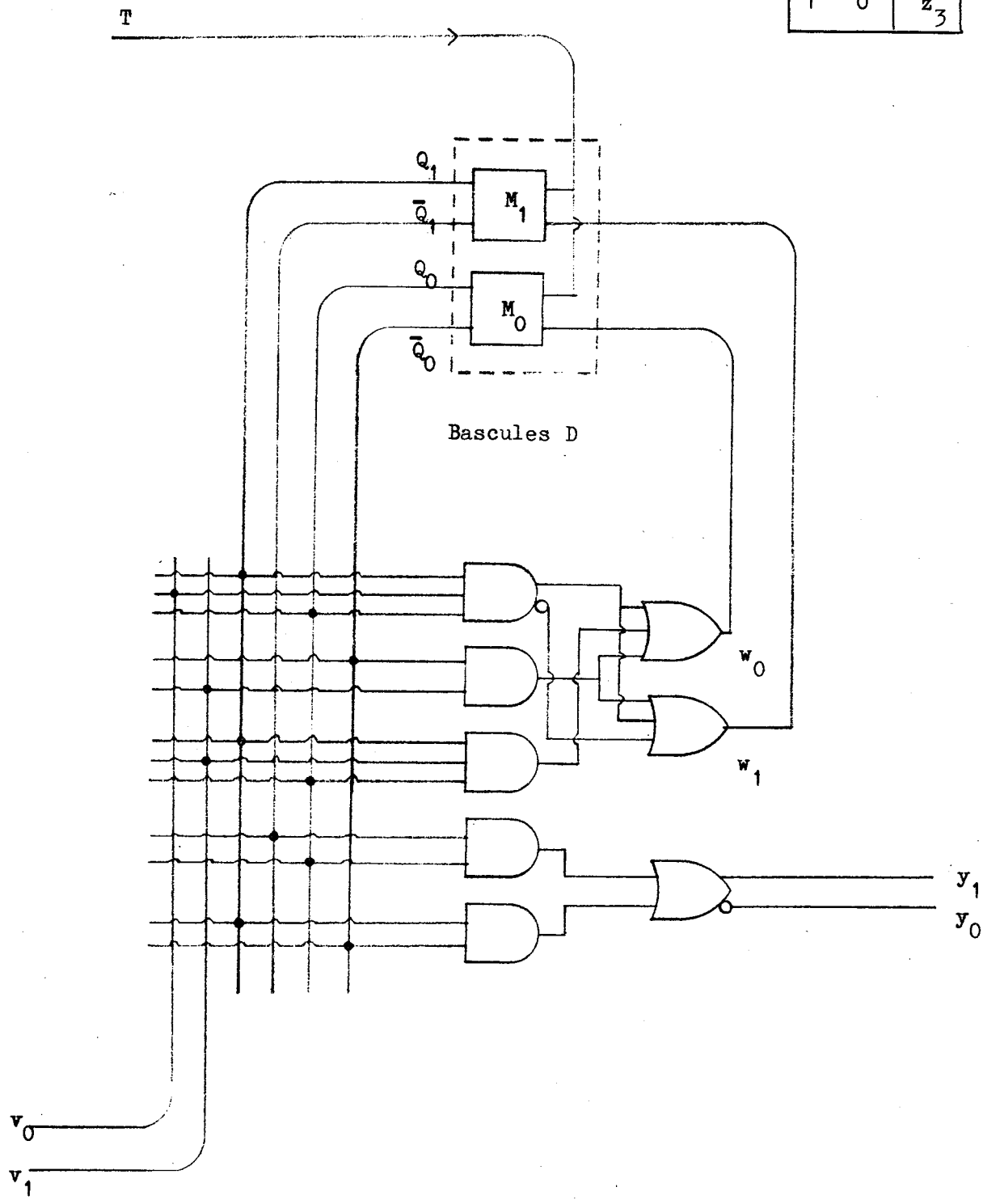


Figure I.10. Solution câblée compacte à bascules T (TRIGGER).

Il est plus naturel pour la même assignation des états d'utiliser des bascules DATA dont les entrées sont de type-forçage comme sur la figure I.11, où l états sont aussi assignés selon le code de GRAY.

Figure I.11. Solution câblée à codage compact.

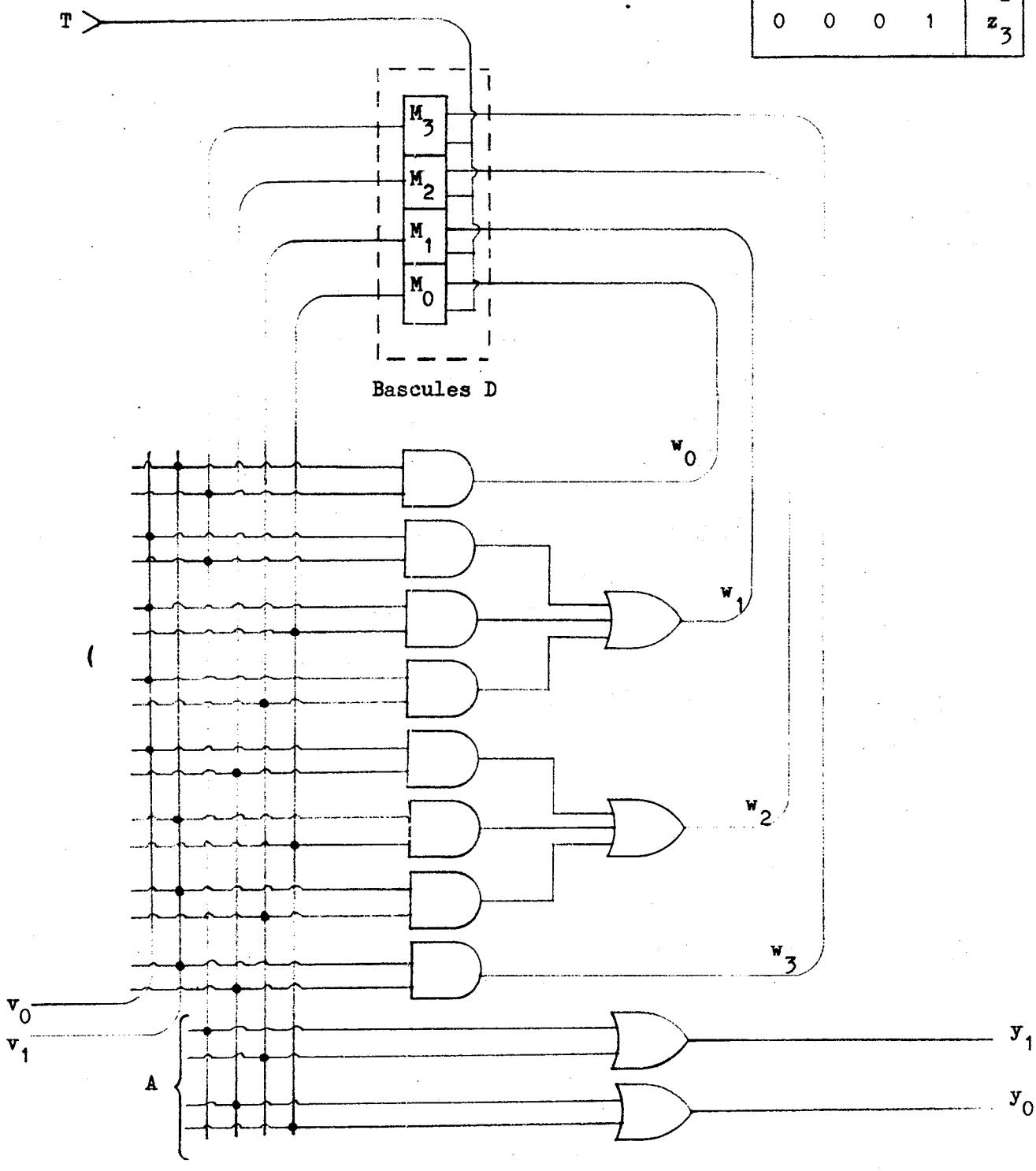
w_0	w_1	Z
0	0	z_0
0	1	z_1
1	1	z_2
1	0	z_3



La figure I.12. présente la technique câblée permettant de rendre plus "lisibles" les fonctions combinatoires d'interconnexion représentant la fonction de transition. Celles-ci sont en forme canonique. L'assignation est dite <u>primitive</u> ou naturelle (ou 1 parmi n) lorsque le nombre de bascules est b = n.

<u>Figure I.12.</u> Solution câblée à codage primitif (1 parmi 4).

w_0	w_1	w_2	w_3	Z
1	0	0	0	z_0
0	1	0	0	z_1
0	0	1	0	z_2
0	0	0	1	z_3

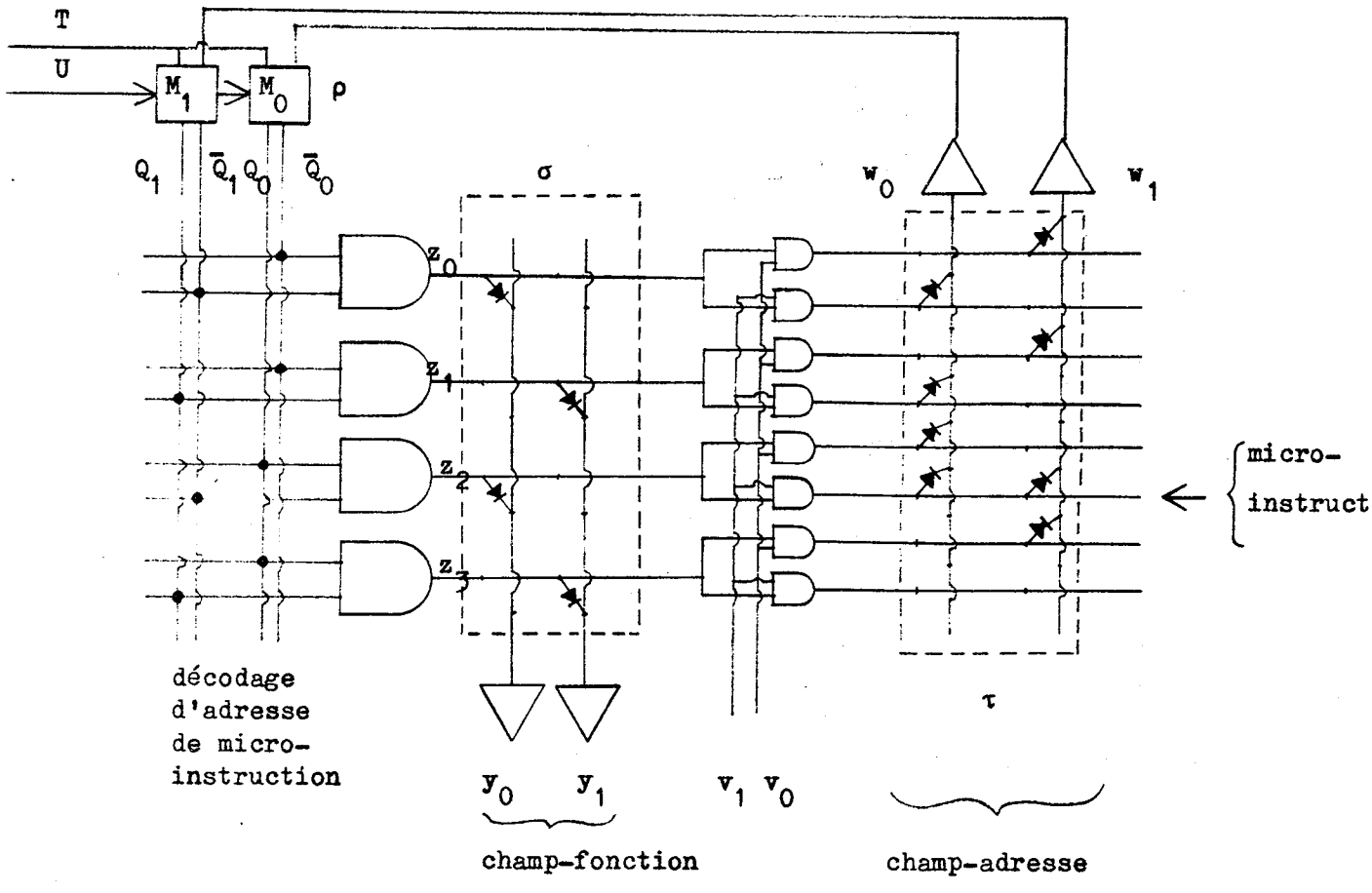


La microprogrammation a été introduite très tôt comme technique de "régularisation" de la structure difficile à câbler sans erreur des fonctions combinatoires de transition et de sortie.

Figure I.13. Solution microprogrammée [WIL. 1951].
Réalisation avec bascules D (DATA) Maître-Esclaves.

Codage des adresses des microinstructions.

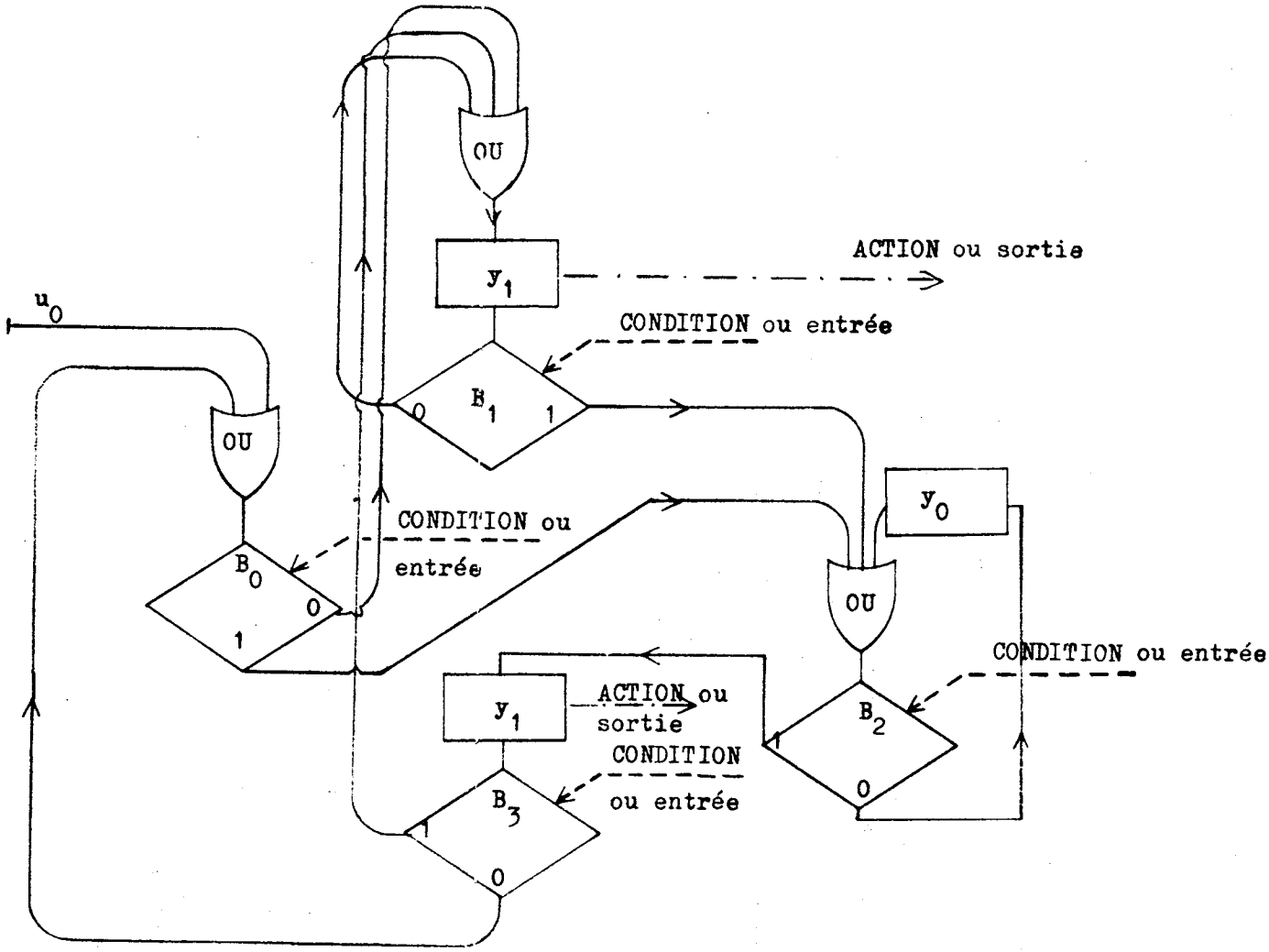
w_0	w_1	Z
0	0	z_0
0	1	z_1
1	0	z_2
1	1	z_3



Note : Ce mode de réalisation en forme état-Sortie est généralement remplacé par une forme Transition-sortie.

La diagrammation a été introduite récemment pour des machines séquentielles relativement petites et lentes. Elle reprend la topologie du graphe des états G(K) avec une assignation naturelle (1 parmi n).

Figure I.14. Solution diagrammée [BEL. 72].
Les divers composants opèrent ici de manière asynchrone (sans horloge) : ce sont le branchement B incorporant une bascule, le OU logique, et l'émetteur de sortie (ACTION).



Il est possible de restructurer topologiquement la solution câblée codage primitif comme indiqué sur la figure I. 15, pour obtenir une implémentati proche de la précédente mais à fonctionnement synchrone.

Figure I.15. Solution déployée.

Cette réalisation est géométriquement proche de la forme diagrammée de la figur I.14 , mais elle est synchrone et correspond à l'emploi de composants standards

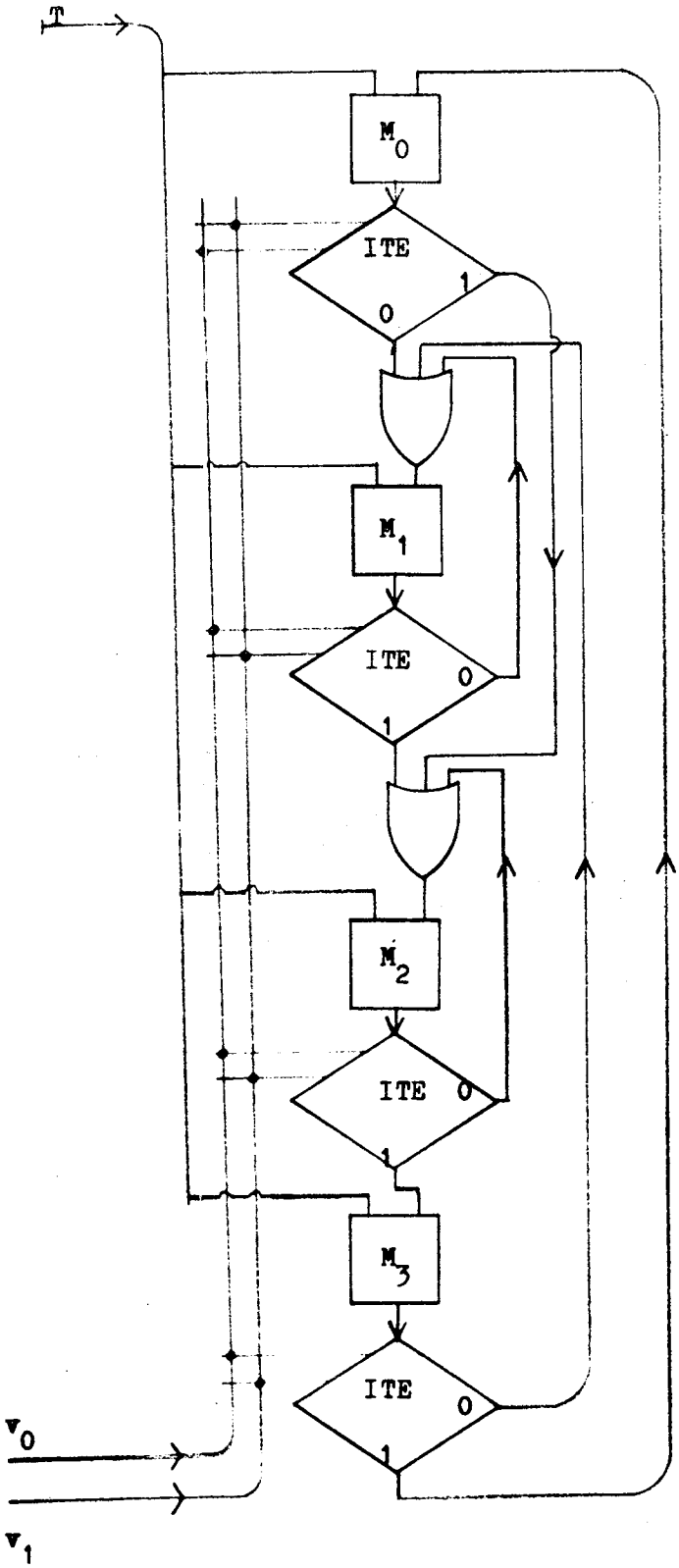
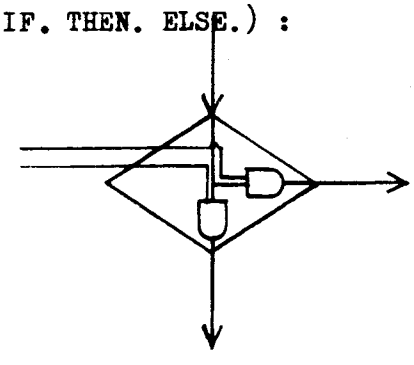


Schéma symbolique du module I

(IF. THEN. ELSE.) :



Enfin à l'aide d'une technique de calcul fondée sur la théorie des "expressions régulières" [WEI. 68], dans le cas d'une machine Σ où $|Y| = p = 2$, une assignation originale permet d'élaborer une solution comme sur la figure I.16, à l'aide d'un seul type de composant appelé module M_i.

Figure I.16. Solution décomposée de WEINER et HOPCROFT.

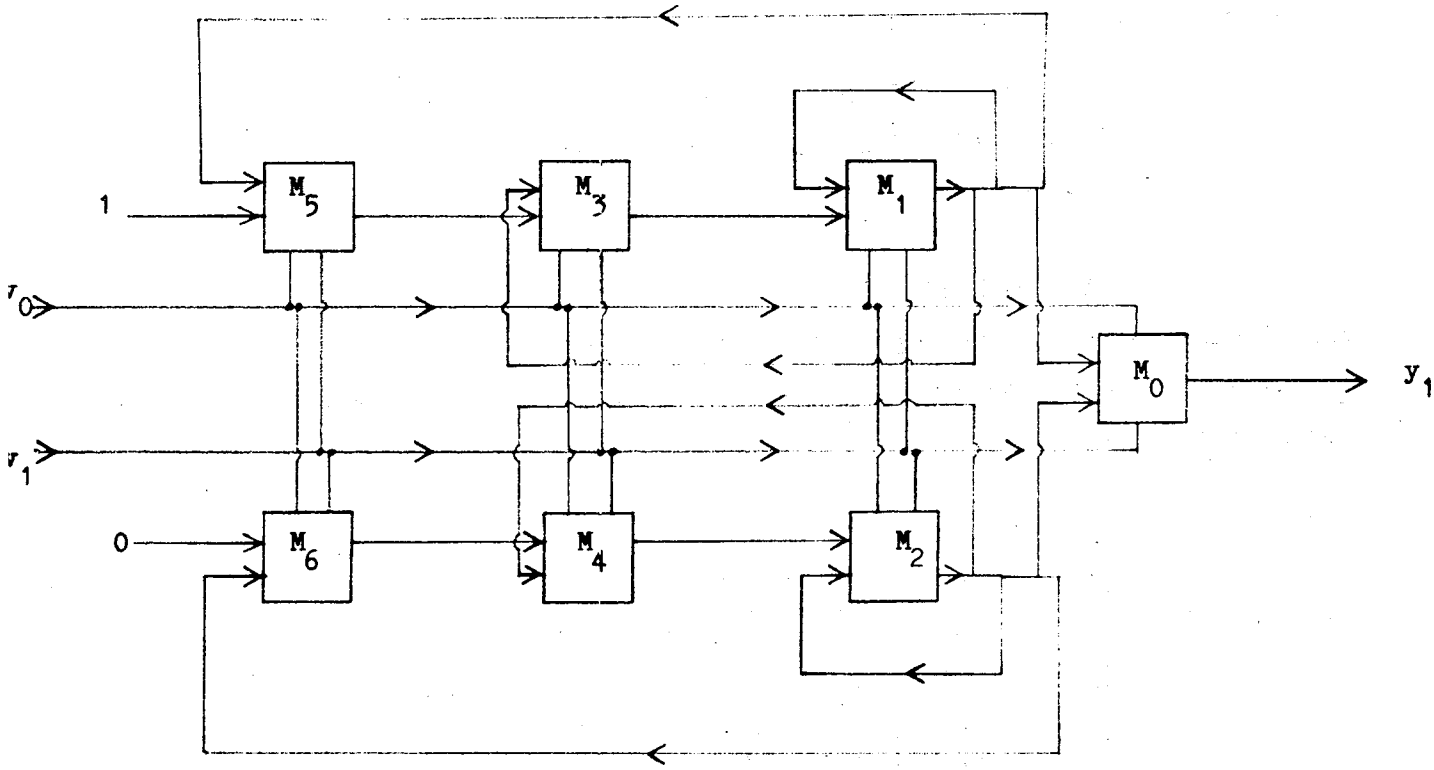
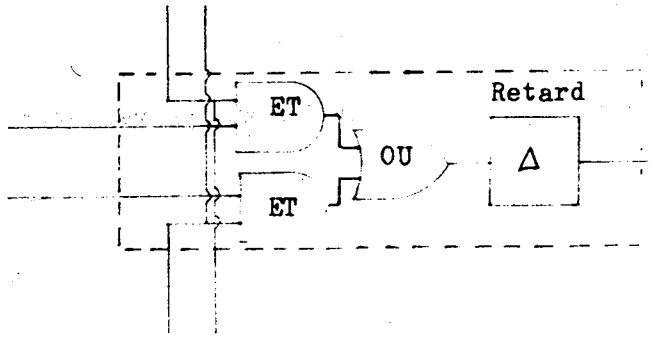


Schéma symbolique d'un module M_i :

D'après le point de vue local de la notion d'état (§.I.3.1) il apparaît que l'espace d'état de Σ est défini par la tension en sortie de ses composants de retard Δ.



I.3.6. Rappel de diverses techniques d'implémentation d'un registre pour un opérateur D.

Un registre est dit général si les bascules qui le composent sont interconnectées de manière à permettre le fonctionnement en compteur-décompteur, en registre à décalage, ainsi que tous les accès d'une unité arithmétique-logique capable d'effectuer l'addition du contenu du registre à celui d'un autre, la mise d'un résultat dans le registre, le test d'égalité à zéro du contenu du registre, la comparaisom du contenu à celui d'un autre registre, etc. En résumé, trois fonctions sont donc nécessaires :

1) l'entrée-sortie sérielle (en décalage gauche ou droit),
2) l'entrée-sortie parallèle (un bit par bascule),
3) l'incrémentation par un.

Les figures I.17, 18 et 19 montrent respectivement comment ces fonction essentielles peuvent être implémentées de manière itérative par interconnexion de bascules et de circuits combinatoires élémentaires (ET, OU et NON-ET $\triangleq$ NET) selon les lois de BOOLE [1854].

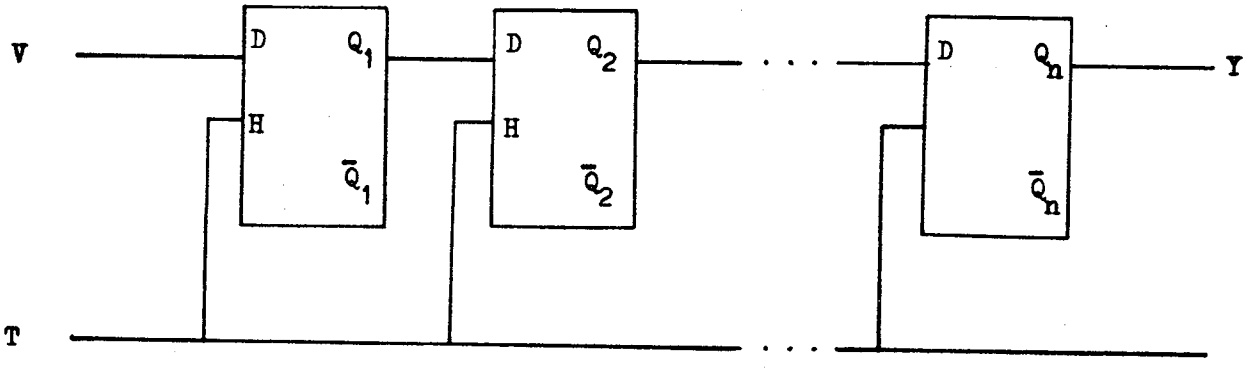


Figure I.17. Bascules formant un registre à décalage M_d.

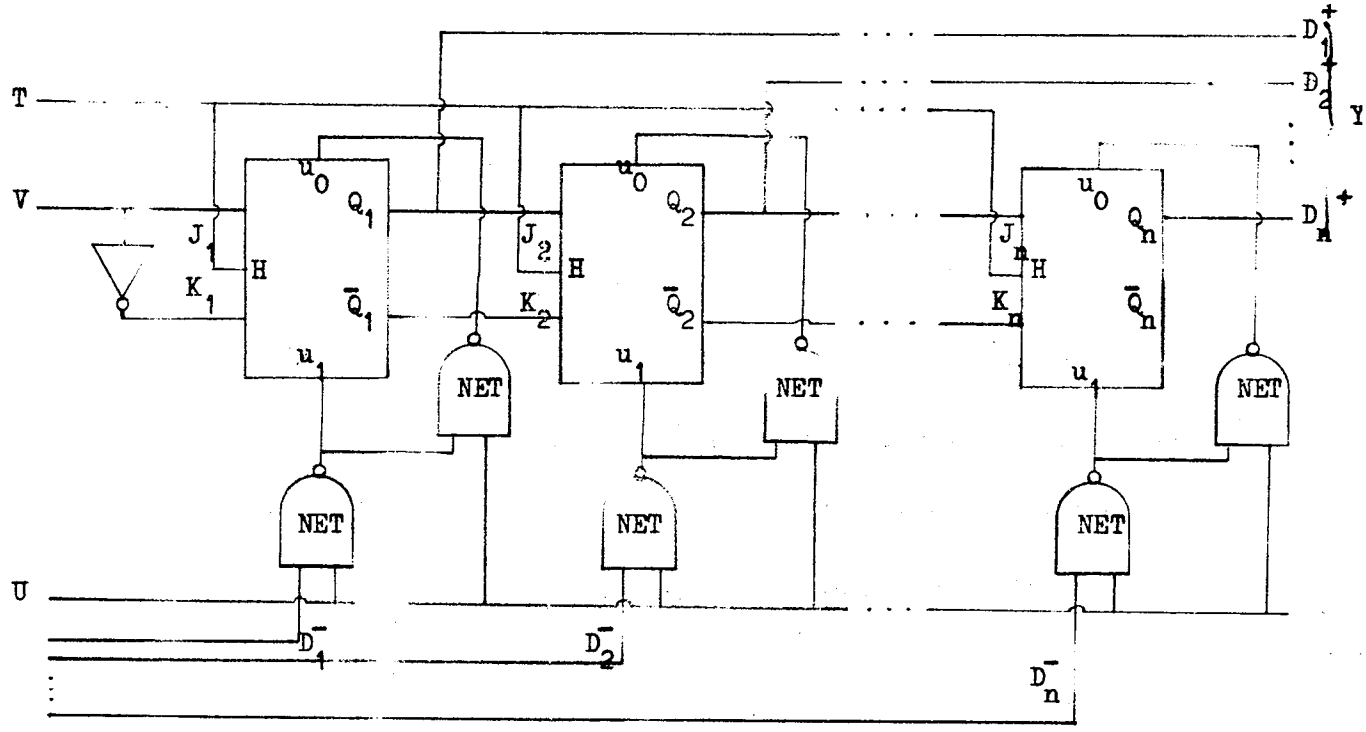


Figure I.18. Registre avec entrées parallèle et sérielle M_{ps}.

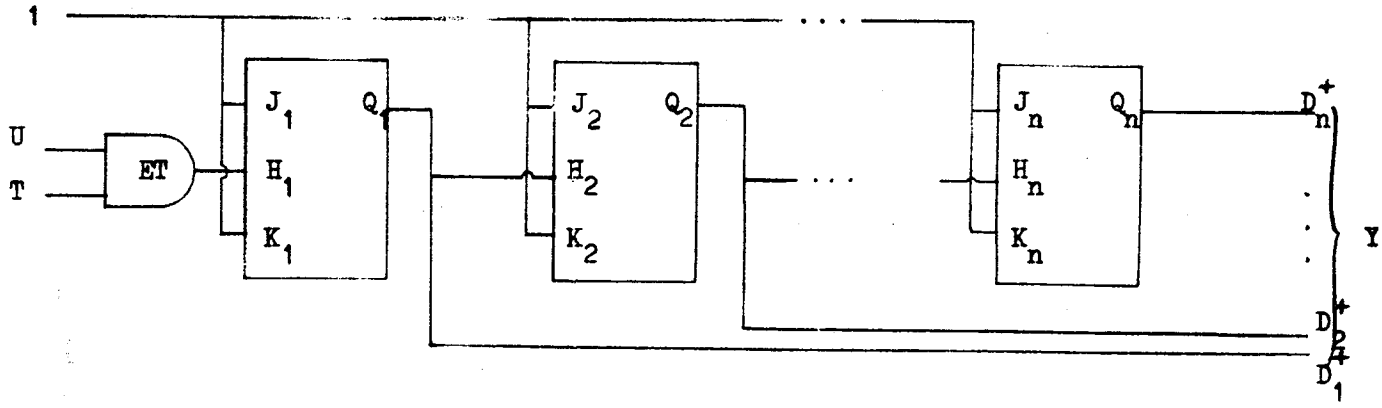


Figure I.19. Compteur binaire à n étages DK.

La valeur $y = \sum_{i=1}^{n} D_i^+ \cdot 2^{i-1}$ est égale au nombre de périodes (T) pendant lesquelles U a eu pour valeur 1. Il est alors possible de considérer le compteur DK non plus comme un opérateur D fournissant le résultat y ∈ Y, mais comme un automate de commande K engendrant les signaux de sortie Y répétitivement.

Ce rappel de diverses techniques d'implémentation d'automates confirme l'existence d'une distinction entre machines séquentielles de commande K et de traitement des données D et par là-même l'intérêt de distinguer entre plusieurs classes de signaux d'entrée comme V et U sur les bascules, et à s'interroger sur l'existence d'un modèle de machine séquentielle adapté aux opérateurs D à structure complexe de registres interconnectés.

CHAPITRE II

QUELQUES PROPRIETES DES POLYMATES

CHAPITRE II

QUELQUES PROPRIETES DES POLYMATES.

Un nouveau modèle de machine séquentielle ou automate est présenté pour séparer les signaux d'entrée en deux classes sémantiquement différentes. Ce modèle est appelé polymate car il généralise la notion d'automate en ce sens qu'il a plusieurs "façons de se mouvoir".

A l'origine de la théorie des automates, se trouvent les premiers efforts de compréhension de la nature des calculateurs ou de l'intelligence humaine [TURING, Mc CULLOUGH et PITTS, VON NEUMANN, CHOMSKY]. Par la suite, divers modèles concrets ont été dégagés [MOORE, MEALY, HUFFMAN, SHANNON et WEAVER, WILKES] afin de permettre d'implémenter avec rigueur les machines séquentielles composant les divers modules d'un système informatique, tandis que des modèles abstraits [SCHUTZENBERGER, MYHILL, GUINSBURG, THATCHER, ARBIB] étaient développés par des mathématiciens désireux de manipuler des structures algébriques riches et nouvelles

Ces deux visions de la théorie des automates n'ont cessé de diverger que dans le domaine de l'analyse syntaxique, où une certaine convergence apparaît occasionnellement. C'est par le biais de la théorie des langages de description des calculateurs [IVERSON, PROCTOR, SCHORR, STABLER, BELL et NEWELL, CHU, PARNAS, SCHLAEPPI, GORMAN et ANDERSON] ainsi que par les problèmes de représentation de la sémantique en général [SCOTT, STRACHEY,...], qu'un certain rapprochement s'ébauche [GERACE, BJØRNER,...] entre les tenants des modèles abstraits et concrets.

Le modèle des polymates est présenté dans plusieurs contextes distincts au long de cette thèse : d'abord par un retour au modèle initial de TURING étendu progressivement jusqu'à une version réaliste de type VON NEUMANN, ensuite par un nouveau départ basé sur le langage PMS de BELL et NEWELL de description des calculateurs. Pour se référer aux auteurs cités, le lecteur peut consulter [ARB.69] et [BEL.71].

II.1. DESCRIPTION INTERNE ET DESCRIPTION EXTERNE.

Un système est usuellement défini avec "un" espace d'entrée. Mais la complexité des concepts de l'Informatique est accrue du fait qu'ils font intervenir plusieurs classes d'entrées essentiellement différentes. (cf. §. I.2,3).

Deux classes peuvent être séparées de manière naturelle si l'on distingue, parmi les entrées agissant sur un système, les <u>événements</u> (ou entrées $U \equiv I_D$ de type argument) et les <u>décisions</u> (ou entrées $V \equiv I_K$ de type fonction).

<u>II.1.1. Définitions formelles</u>.

Un polymate est un octuple $\Pi \triangleq (U,V,Z,Y,T,\rho,\tau,\sigma)$ de description interne o

- U est un espace vectoriel d'entrée dite <u>additive</u> représentant les événements,
- V est l'ensemble fini d'entrée dite <u>multiplicative</u> représentant les décisions,
- Z est <u>l'espace d'état</u> (ensemble des paramètres $\{\eta\}$ suffisant à résumer le passé du polymate pour prédire son futur) et Y est l'espace vectoriel de sortie,
- T est l'espace discret du temps,

et les fonctions ρ,τ et σ sont celles de <u>rythme</u>, de <u>transition</u> et de <u>sortie</u> respectivement, la première permettant de ne prendre en compte qu'un type d'entrée à la fois. Il est possible de ne pas expliciter le temps dans la mesure où il est en bijection avec les éléments de V dans les séquences $\tilde{v}$ définies par la concaténation qui tient lieu de multiplication dans l'ensemble $V^* \triangleq \{\tilde{v}\}$ tel que $\tilde{v} \triangleq v_1 \ldots v_t = v(1)\ldots v(t)$; V^* est appelé l'<u>itéré</u> de V.

Les espaces vectoriels considérés U et Y sont généralement définis sur les réels.

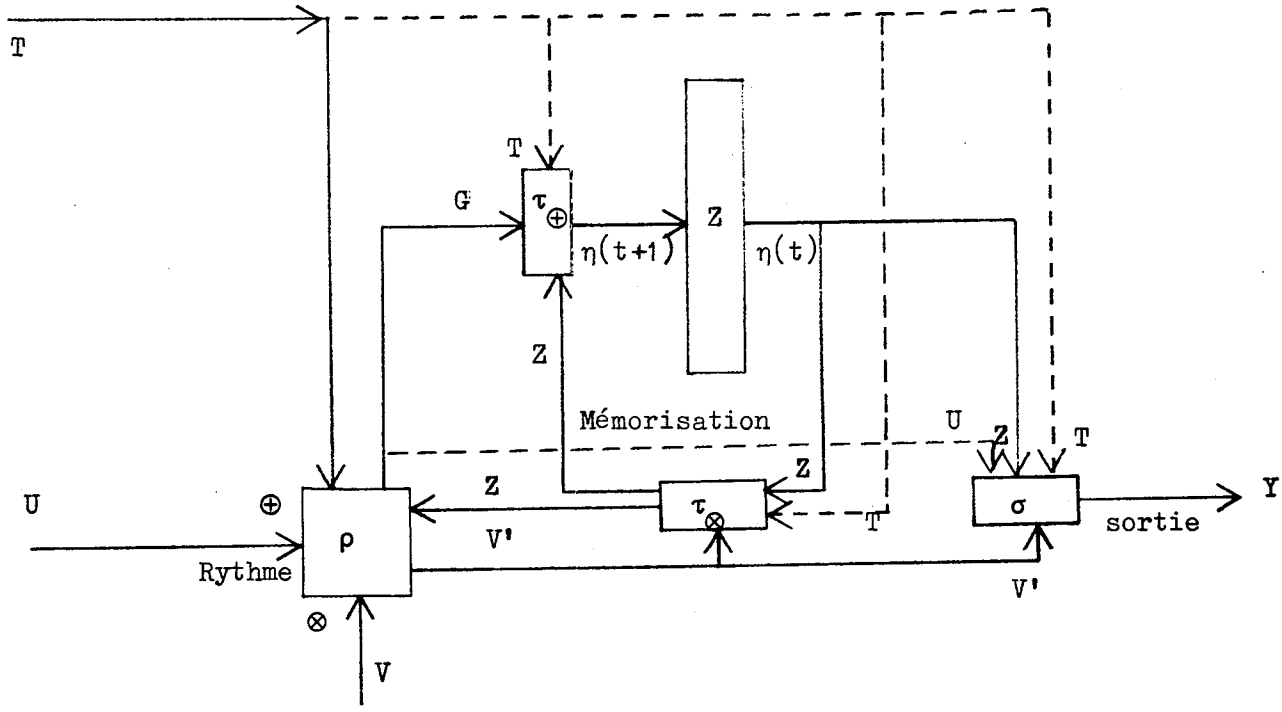
Ainsi $\rho : Z \times U \times V \times T \to G \times V'$ <u>linéaire</u> en U, c'est-à-dire que G est un espace vectoriel avec $\rho(\eta,u_1,v,t) + \rho(\eta,u_2,v,t) = \rho(\eta,u_1+u_2,v,t)$, et $V' \subset V^1 \triangleq V \cup \Lambda$; Λ est l'élément neutre qui permet d'introduire des délais pour synchroniser les actions d'entrée. Par exemple, soit $\rho(\eta,u,v,t) \mapsto (u,\Lambda)$ soit (OU-EXCLUSIF) $\rho(\eta,u,v,t) \mapsto (0,v)$, de sorte que événements et décisions ne puissent interférer aléatoirement. L'essentiel est que les transitions dans $U \times U$ et dans $V \times V$ soient remplacées par des transitions "coordonnées" dans $(G,V') \times (G,V')$

Ensuite $\tau : Z \times G \times V' \times T \to Z$ est la fonction de transition d'états, <u>linéaire</u> en G. Elle peut être factorisée en $\tau_\otimes : Z \times V' \times T$ non-linéaire et $\tau_\oplus : Z \times G \times T$ linéaire.

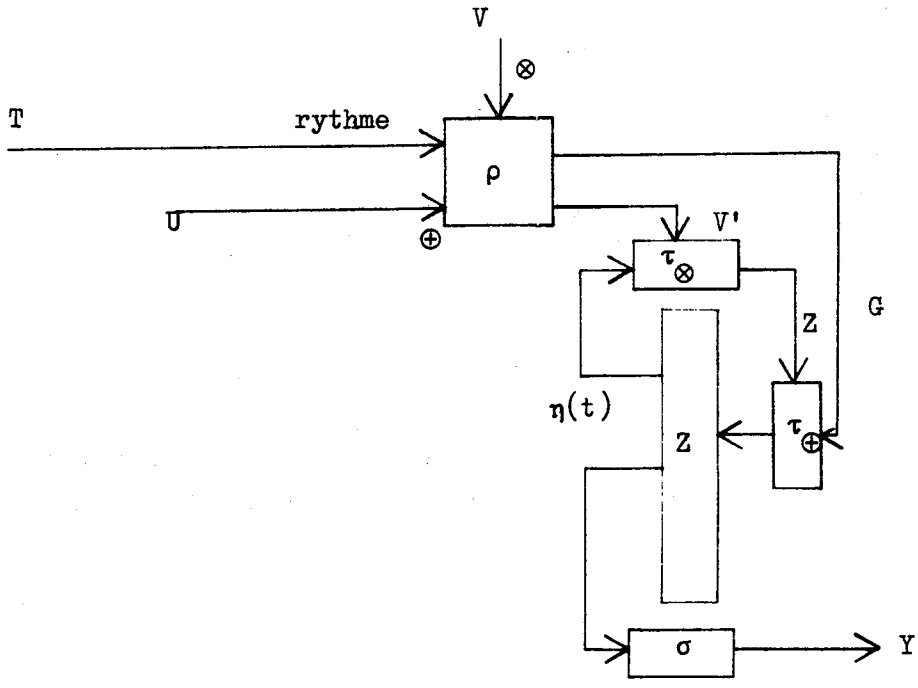
Enfin $\tau : Z \times G \times V' \times T \to Y$ est la fonction de sortie.

A la suite de [HUF.54], la forme canonique des polymates est présentée sur la figure II.1 pour visualiser les fonctions de l'octuple Π.

Figure II.1. Représentations canoniques d'un polymate.



a) Schéma complet (avec factorisation de τ en $\tau_{\oplus}$ et $\tau_{\otimes}$).



b) Schéma rapproché de la forme canonique des automates (Figure I.4).

Si la fonction ρ assume aussi la fonction de synchronisation au sens du paragraphe I.3.1, alors les fonctions τ et σ peuvent être restreintes à :

$$\tau : Z \times G \times V' \to Z \quad \text{et}$$
$$\sigma : Z \times G_{\times} V' \to Y.$$

En général σ peut être indépendant de G et, en forme état-sortie, cette fonction peut se réduire à $\sigma : Z \to Y$.

Un polymate est fini s'il admet une représentation sur un corps fini, et en particulier sur $\{0,1\}$. Puisque la cardinalité de V est finie, $|V| = q < \infty$, il s'ensuit que l'espace d'état est discret et fini : $|Z| = n < \infty$.

II.1.2. Représentations matricielles.

Les fonctions du polymate peuvent être représentées sous forme d'un système d'équations matricielles comme suit :

$$\begin{cases} \eta(t+1) = \eta(t)B_{v(t)} + u(t)C_{v(t),\eta(t)} & \text{représentant les fonctions } \rho \text{ et } \tau, \\ y(t) = \eta(t)A_{v(t)}, & \text{représentant la fonction } \sigma, \end{cases}$$

où, si le polymate est fini, la matrice d'observation A est une matrice d'agrégation sur $\{0,1\}$, monomiale en ligne, ainsi que η et y.

La factorisation de τ en $\tau_{\otimes}$ et $\tau_{\oplus}$ apparaît dans ces équations car $\tau_{\otimes}$ effectue la transformation $\eta(t) \mapsto \eta(t)B_{v(t)} = z(t)$ tandis que $\tau_{\oplus}$ effectue la transformation $z(t) + g(t) = \eta(t+1)$ où $g(t)$ résulte de la transformation $\rho : (\eta(t),u(t),v(t)) \mapsto g(t) = u(t)C_{v(t),\eta(t)}$. Le plus souvent on considère $\rho = \rho_H$, (§.I.3.2), tel que C est indicé seulement par $v(t)$.

On peut considérer un tel système dynamique comme programmable en ce sens que la séquence $\underset{\sim}{v}$ des décisions régule l'effet des événements $u(t)$.

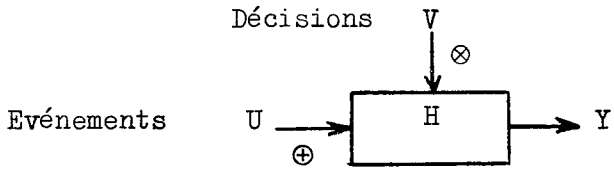


Figure II.2. Schéma fonctionnel d'un polymate.

Cette représentation matricielle satisfait bien la définition de l'octuple Π, où t est l'indice du temps pris dans Z^+ (entiers non négatifs), η est le vecteur ligne 1 × n représentant l'état, u est le vecteur d'entrée additive 1 × m et y le vecteur de sortie 1 × p. Les matrices A, B et C sont celles d'observation (n × p), de dynamique (n × n) et de commande (m × n) respectivement. Elles sont paramétrées par les décisions $v \in V^1$ de sorte que $A_\Lambda = 0$, $B_\Lambda = 1$ et $C_\Lambda = 0$: les paramètres d'évolution v ont donc une action multiplicative par l'intermédiaire de B conformément à la définition de la fonction de transition τ du paragraphe II.1.1.

PROPRIETES II.1. a) Un système linéaire est un polymate, où $|V| = q = 1$, c'est-à-dire à dynamique invariante. [KAL.68].

b) Un automate stochastique généralisé [PAZ.70] est un polymate où A est constant, $\{A_{\tilde{v}} \equiv A, \forall \tilde{v} \in V^*\}$, où C est nul (élimination de l'additivité) et où le temps est spécifié implicitement par l'entrée des éléments de V.

c) Une machine-outil programmable [CAR.69] est un polymate où $A_{\tilde{v}} \equiv I$, $\forall \tilde{v} \in V^*$ et C est nul et où l'indice du temps est pris sur un ensemble fini ordonné, $|T| = s < \infty$.

Le polymate est donc une extension des modèles dynamiques pré-existants. Une action est une paire (événement, décision) :

$$w = (u,v) \in W \triangleq (U,V).$$

THEOREME II.2. Toute action effectue sur l'état du polymate une transformation affine.

Démonstration. Une transformation est dite affine si elle prend la forme $f(x) = x \times \alpha + \beta$. Soit $g(t) \triangleq u(t)C_{v(t)} = \rho(w,t)$ résultant de la fonction de rythme,

il s'ensuit : $$\eta(t+1) = \eta(t)B_{v(t)} + g(t). \qquad (2)$$

Il suffit de noter $B_{v(t)} = h(t)$, de sorte qu'à chaque instant le vecteur d'état η est transformé en $\eta \times h + g$.

C.Q.F.D.

En ce sens un polymate est l'extension affine d'un automate sur l'état duquel une décision effectue une transformation homothétique.

II.1.3. Représentations tensorielles.

Le polymate diffère peu en apparence des autres modèles : il s'en distingue cependant par le type de questions qu'il permet d'aborder et surtout par la présence simultanée des deux classes d'entrée {U et V}. Tandis que la tendanc [ARB.69] était de rapprocher systèmes linéaires et automates, la spécification d ce modèle-là fait ressortir la différence sémantique entre ces deux modèles-ci p la signification soit additive, soit multiplicative, des entrées.

THEOREME II.3.

L'espace d'entrées additives et l'alphabet d'entrées multiplicatives in duisent des décompositions différentes des polymates.

Démonstration. La représentation multidimensionnelle d'un polymate Π est l'ensemb des trois tenseurs $A \triangleq (A_{v_h})$, $B \triangleq (B_{v_h})$, $C \triangleq (C_{v_h})$, $\forall v_h \in V$, le mot tenseur étan pris ici en son acception la plus simple d'ensemble de matrices, tel que $A = (a_i^{jh})$, $B = (b_i^{jh})$ et $C = (c_i^{jh})$ rassemblent les coefficients des fonctions de transition et de sortie de (1). Avec la convention d'Einstein des produits en notation tensorielle $\{\Sigma_j \alpha_{ij}\beta_{ik} \triangleq \alpha_i^j \beta_j^k\}$, l'évolution des composants $\eta^j(t)$ du vecteur d'état est donnée par la représentation équivalente à (1) : (*)

$$\begin{cases} \eta^j(t+1) = \eta^i(t) b_i^{jh} v_h(t) + u^\ell(t)\, c_\ell^{jh} v_h(t), \\ y^j(t) = \eta^i(t)\, a_i^{jh}\, v_h(t), \end{cases} \qquad (3)$$

où le vecteur monomial $v(t) = \delta_k^h$, $\forall k$ tel que $v(t) = v_k \in V$, et δ est le symbole de Kronecker. Cette mise en forme de minterme de V permet de définir les formes disjonctives $v \oplus v'$. Ainsi $b_i^{jh} v_h(t) \equiv (B_{v(t)})_i^j$, matrice de dynamique de (1).

Un système est dit décomposable lorsqu'il est homomorphe d'un système défini par composition à partir d'autres systèmes. Diverses compositions de systèmes dynamiques ont été étudiées algébriquement. Pour les besoins de la distinction sémantique entre classes d'entrée, il suffit ici de considérer les compositions par somme directe sur les entrées additives (et sur l'espace d'état) et par produit direct (cartésien) sur les entrées multiplicatives, puis de défini: $U = \overset{r}{\underset{d=1}{+}} U^{(d)}$ et $V = \overset{r'}{\underset{d=1}{\times}} V^{(d)}$ où les $U^{(d)}$ et $V^{(d)}$ sont les r et r' ensembles composants respectivement. C.Q.F.D. (Ce modèle sert au §.III.4. D'autres types de décomposition requérant plus de sémantique seront étudiés ensuite au chapitre I

(*) Définition. Deux représentations sont équivalentes si l'on peut passer de l'une à l'autre par une transformation inversible.

La représentation (3) est fondamentale aussi pour illustrer géométriquement la dualité au sens de KALMAN [68] pour les systèmes linéaires et la dualité pour les automates [DEP.68], c'est-à-dire une association entre propriétés sur les entrées et les sorties. Dans le premier cas, le vecteur v(t) est réduit à un scalaire et les représentations vectorielles des entrées et sorties duales sont liées par transposition : cette dualité associe les tenseurs A et C^t (transposé de C). Dans le deuxième cas, le tenseur C est nul et la même dualité fait correspondre à A les vecteurs d'état. Ce concept est approfondi et étendu en référence [DEP.74.a].

Or le tenseur B d'un polymate peut être vu de trois manières comme un ensemble de matrices :
(i) $B \triangleq \{b_{\cdot}^{\cdot h}\}$ est le générateur multiplicatif de transition représentant l'alphabet de décision $\{v_h\} = V$; où les indices i et j de b_{ij}^{ih} sont remplacés par des points pour montrer que l'entité considérée est l'ensemble structuré en matrices représentant les v_h.
(ii) $B_i = (b_i^{\cdot\cdot})$ est la strate associée bijectivement à tout état de base $\eta^i(t)$; tandis que les états $\eta(t)$ sont générés additivement par les "strates d'entrée" $(c_\ell^{\cdot\cdot}) \triangleq C_\ell$. Le fondement du concept de dualité tient à ce que du point de vue matriciel le générateur agit de manières bijectives sur les $b_i^{\cdot\cdot}$ et les $b_{\cdot}^{j\cdot}$.
(iii) $B^j = (b_{\cdot}^{j\cdot})$ est la costrate associée dualement à tout coétat de base, représenté par une "costrate de sortie" $A^j \triangleq (a_{\cdot}^{j\cdot})$.

Le chapitre V qui présente le résultat essentiel de cette thèse est entièrement fondé sur ces concepts de strates et costrates (ainsi que le théorème II.8).

THEOREME.II.4. Les fonctions de rythme, de transition et de sortie sont des fonctions de norme généralisées.

Démonstration. La représentation tensorielle (3) et ces réflexions sur la dualité structurale dans les polymates mènent à définir une représentation équivalente, par produits scalaires avec des matrices de norme A^j, B^j, C^j non symétriques et donc dites généralisées :

$$\begin{cases} \eta^j(t) = ||\eta(t), v(t)||_{B^j} + ||u(t), v(t)||_{C^j}, \\ y^j(t) = ||\eta(t), v(t)||_{A^j}. \end{cases} \qquad (4)$$

C.Q.F.D.

Il est remarquable que le terme additif $g^j(t) = u^\ell(t)\, c_\ell^{jh}\, v_h(t)$ fasse jouer des rôles similaires aux évènements u(t) et aux décisions v(t) ; la nature des entrées multiplicatives se distingue cependant par son effet synchrone sur les trois composants g(t), $\eta(t)$ et y(t).

THEOREME.II.5. Les représentations vectorielles des entrées additives u(t) et des entrées multiplicatives v(t) sont invariantes au cours d'un changement de représentation d'un polymate.

Démonstration. Soient Π et Π' deux représentations équivalentes {A,B,C} et {A',B',C'} d'un polymate. Les relations suivantes doivent alors être satisfaites :

$$\begin{cases} u(t)\ (C\ K^{-1})\ (K\ B_{v(t)}\ K^{-1})\ (K\ A) = u(t)C'B'_{v(t)}A', \\ (\eta(t)\ K^{-1})\ (K\ B_{v(t)}\ K^{-1})\ (K\ A) = \eta'(t)\ B'_{v(t)}A', \end{cases} \tag{5}$$

pour qu'il y ait équivalence, pour une matrice K carrée n × n régulière. La représentation avec produits scalaires s'écrit alors :

$$\begin{cases} \eta'^{j}(t) = \| \eta'(t),\ v(t) \|_{B',j} + \| w(t) \|_{C',j}, \\ y^{j}(t) = \| \eta'(t),\ v(t) \|_{A',j} \text{ avec } B' = KBK^{-1}, A' = KA \text{ et } C' = CK^{-1}. \end{cases} \tag{6}$$

C.Q.F.D.

II.1.4. Fonctions de comportement.

Lorsque le polymate est vu de l'extérieur, comme une "boite noire", quatre types de fonctions peuvent être utilisés pour sa description externe : la propriété de linéarité en U implique l'existence d'un état zéro η_0 où le polymate revient au repos (en l'absence de toute entrée) s'il est stable, au même sens que pour les systèmes linéaires. C'est l'absence de cet état d'équilibre qui complique la description externe des automates :

(i) La liste des relations entrée-sortie sur les évènements :

$$f_{\tilde{v}(t)} : U^* \times T \to Y : \tilde{u}(t) \mapsto y(t) = \sigma\{\tilde{\tau}[(\eta_0, \tilde{\rho}(\tilde{u},\tilde{v}),t]\},$$

où $\tilde{\tau}$ et $\tilde{\rho}$ sont les extensions naturelles de τ et ρ à des chaînes d'actions telles que $\tilde{\tau}[\eta_0,\tilde{\rho}(u_1u_2,v_1v_2),t] = \tau\{\tau[\eta_0,\rho(u_1,v_1),t-1],\rho(u_2,v_2),t\}$ d'après la définition en II.1.1. Elle permet d'introduire l'équivalence de Nérode additive pour une chaîne de décisions inconditionnelles $\tilde{v}\tilde{v}$ [KAL.68] permettant le passage de la description interne à la description externe :

$$\tilde{u}_1 \equiv_{\oplus} \tilde{u}_2 \Leftrightarrow f_{\tilde{v}\tilde{v}}(\tilde{u}_1 u) = f_{\tilde{v}\tilde{v}}(\tilde{u}_2 u),\ \forall \tilde{v}\tilde{v} \in V^*,\ \forall u \in U^*.$$

(ii) La fonction séquentielle sur les décisions :

$$f_{\tilde{u}(t)} : V^* \times T \to Y : \tilde{v}(t) \mapsto y(t) = \sigma\{\tau[\eta_0, \rho(\tilde{u}, \tilde{v}), t]\}.$$

Elle permet d'introduire l'équivalence de Nérode multiplicative [ARB.69] pour une chaîne d'événements connus $\tilde{u}u$, et ainsi d'automatiser l'identification d'une description externe pour réaliser une description interne équivalente [BAN.74] :

$$\tilde{v}_1 \equiv_{\otimes} \tilde{v}_2 \Leftrightarrow f_{\tilde{u}u}(\tilde{v}_1 v) = f_{\tilde{u}u}(\tilde{v}_2 v), \ \forall v \in V^* , \ \forall \tilde{u}u \in U^*.$$

(iii) La fonction d'activité :

$$f : (U,V)^* \times T \to Y : w(t) \triangleq [u(t), v(t)] \mapsto y(t).$$

Elle permet d'introduire l'équivalence de Nérode généralisée sur les actions $\tilde{w}_1 \equiv \tilde{w}_2 \Leftrightarrow f(\tilde{w}_1 w) = f(\tilde{w}_2 w), \ \forall w, \tilde{w}_1, \tilde{w}_2 \in W^* = (U,V)^*$.

(iv) La fonction de comportement : une expérience étant un triplet $[y(t)|u(t),v(t)] = x(t)$, l'existence d'une fonction d'activité $f[w(t)] = y(t)$ peut s'exprimer [DEP.68] sous la forme :

$$\mu : (Y|U,V)^* \times T \to [0,1] : x(t) \mapsto \begin{cases} 1 \text{ si } \{f[w(t)]|\ w(t)\} = x(t), \\ \text{sinon } 0. \end{cases}$$

Le lecteur trouvera en [BAN.74.e] une application de ce concept dans le cas où u(t) sert à procurer une séquence de rentrée ("homing") pour un automate stochastique.

La fonction d'activité d'un polymate est liée à la représentation interne par les relations suivantes :

$$\begin{cases} y(t) = \eta(t)\, A_{v(t)} = [\eta(t-1)B_{v(t-1)} + u(t-1)C_{v(t-1)}]\, A_{v(t)}, \\ y(t) = \eta_0 \prod_{j=0}^{t-1} B_{v(j)} A_{v(t)} + \sum_{k=1}^{t-1} u(k) C_{v(k)} \prod_{\ell=k+1}^{t-1} B_{v(\ell)} A_{v(t)}. \end{cases} \tag{7}$$

II.1.5. Application à la modélisation des bascules.

Soit un automate (cf.§.I.3.3) JK = (V,Z,Y,τ,σ) où $Z = \{\eta_0, \eta_1\}$ avec $V = Z^Z$ et qui admet la représentation matricielle suivante.

$\eta_0 = (1,0) = y_0$ et $\eta_1 = (0,1) = y_1$ avec $A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$.

$B_{u_0} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = B_\Lambda$, $B_{v_0} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}$, $B_{v_1} = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$, $B_{u_1} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$.

α)Addition en mode H.

Si maintenant on définit que l'addition sur $\{0,1\}$ s'effectue modulo 2, il est possible de représenter $U = \{u_0,u_1\}$ séparément de $V = \{\Lambda,v_0,v_1\}$ comme suit :

$$C_\Lambda = [1,1],\ C_{v_0} = C_{v_1} \equiv [0,0],$$

$u_0 = 0$ et $u_1 = 1$.

On peut maintenant séparer les parties additive et multiplicative de JK en deux :

-un système linéaire dit TRIGGER $T = (U,Z,Y,\tau_\oplus,\sigma)$
-un automate dit DATA $D = (V,Z,Y,\tau_\otimes,\sigma)$

L'ensemble forme un polymate $(U,V,Z,Y,\rho,\tau,\sigma)$ dont la représentation gagne en compacité par rapport à l'automate JK mais surtout porte une sémantique particulière.

Il importe à ce point, par comparaison à [HAR.66], de noter que U est le générateur d'un groupe additif abélien tandis que V est le générateur d'un semigroupe. **La fonction de U est de <u>permuter</u> les états.**

Par ailleurs la définition de la fonction de rythme ρ (mode H : hiérarchisé ou avec horloge) donnée par le choix des C_{v_i} entraîne que les actions de V sont prioritaires sur celles de U (puisque seul Λ^i n'inhibe pas U).

β)Addition en mode F.

On pourait cependant imaginer un autre polymate pour JK en imposant (comme en I.3.2) que ρ soit fonction de l'espace d'état Z (figure I.5) et que η_0 soit l'état de repos :

$$B_{u_0} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},\ B_{u_1} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},\ v_0 = 0,\ v_1 = 1,\ C_{\eta_0} = [1\ \ 1],$$

et $C_{\eta_1} = [0\ \ 0]$ permettant de rendre (par réinformation : mode F) les actions multiplicatives prioritaires sur les actions additives **dont la fonction consiste à <u>forcer</u> un état (initial).**

Note. Au lieu de considérer V* comme un langage pour étudier les polymates, il est possible d'utiliser la fonction de comportement μ et de considérer $(Y|U,V)^*$ comm une série formelle de puissance en variables non commutatives, c'est-à-dire une fonction de X^* dans $[0,1]$ avec $X = (Y|U,V)$:

$$\mu = \sum_{t=0}^{\infty} \langle \vec{x}(t), \mu > \vec{x}(t).$$

De même que les monoïdes procurent aux automates un modèle algébrique adéquat, et que les K-modules [KAL.68] jouent ce rôle pour les systèmes linéaires (Théorème II.1), de même le modèle algébrique associé aux polymates à commande C et agrégation A constantes est celui des algèbres associatives libres [PAZ.70].

II.2. MACHINES NON-LINEAIRES, STOCHASTIQUES OU ASYNCHRONES.

(i) Un polymate est linéaire en U si la fonction de transition $\tau : Z \times T \times U \times V^1 \to Z$ est linéaire en U. Un système $f(u,x)$ est <u>non-linéaire</u> si $f(u_1,x) + f(u_2,x) \neq f(u_1+u_2,x)$. Un polymate est dit <u>linéaire</u> s'il est linéaire en V.

(ii) Un polymate est <u>stochastique</u> si son espace d'état H est un polytope convexe.

(iii) Une machine est <u>asynchrone</u> si l'indice des temps ne peut être mis en bijection avec les entiers non négatifs Z^+.

(iv) <u>Un canal de communication</u> sans mémoire est un système associant à toute entrée $u(t)$ la probabilité conditionnelle $p(y|u)$ que sa transmission résulte en la sortie y.

Dans chacun de ces cas, le polymate apporte de nouvelles possibilités pour modéliser des phénomènes réels, et il permet d'introduire la <u>réinformation</u> ("feedback").

II.2.1. Systèmes non-linéaires.

<u>PROPRIETES II.6.</u> Tout système non-linéaire peut être mis sous la forme canonique d'un polymate avec loi de réinformation :

$$\varphi : Z \times U \to V.$$

Cette propriété "évidente" présente des avantages divers qui dépendent étroitement de la classe de non-linéarité considérée ; elle est ici illustrée sur l'exemple de l'équation de Van Der Pol (échelle continue pour le temps).

<u>Exemple</u> : Soit $y'' - \varepsilon(1 - y^2)\, y' + y = 0$.

Sous forme matricielle :

$$[y'y''] = [yy'] \cdot \begin{bmatrix} 0 & -1 \\ +1 & \varepsilon(1 - y^2) \end{bmatrix}$$

Soient $\eta^1 = y$ et $\eta^2 = y'$ les composants du vecteur d'état. Le système de Van Der peut être mis sous la forme du polymate suivant : $\eta' = \eta \cdot (F_\Lambda + F_\varphi)$, où $F_\Lambda = \begin{bmatrix} 0 & -1 \\ +1 & \varepsilon \end{bmatrix}$ correspond à la partie linéaire du polymate et se présente sou forme canonique, tandis que $F_\varphi = \begin{bmatrix} 0 & 0 \\ 0 & -\varepsilon(\eta^1)^2 \end{bmatrix}$ correspond à la partie non-linéaire. Cet exemple illustre un cas particulier autonome $[g(t) \equiv 0]$.

II.2.2. Les polymates stochastiques.

Il s'agit ici des applications au domaine des Transmissions de l'Inform

Le polymate stochastique fini est l'extension affine de l'automate stoc tique ; leurs états sont des distributions de probabilité sur un support Z fini d'états déterministes $\{\eta^1,\ldots,\eta^n\} \Rightarrow \eta = [p(\eta^1),\ldots,p(\eta^n)] \in [0,1]^Z$,

avec $\sum_{i=1}^{n} p(\eta^i) = 1$ et $p(\eta^i) \in [0,1]$. Ses équations de mouvement sont (cf. Théorème II.2) :

$$\begin{cases} \eta(t+1) = \eta(t)B_{v(t)} + u(t)C_{v(t)}, \\ y(t) = \eta(t)A_{v(t)}, \end{cases} \qquad (8)$$

où les matrices $A_{v(t)}$ sont Markoviennes : $\{a_i^{jh} \geq 0$ avec $A_v e_p = e_n$ pour des vecteu e_k colonnes de 1 de hauteur k $\}$, et où $\{\eta(t)e_n = 1, u(t)e_m = 1, y(t)e_p = 1\}$ et la transformation τ est affine : $\{B_v e_n = \omega_v e_n$, $C_v e_n = (1 - \omega_v)e_m$ pour $\omega_v \in [0,1], \forall v$

THEOREME.II.7. Si $\omega_v = 0$, le polymate est sans mémoire et si $C_v \equiv C$, $\forall v \in V$, c'e un canal de communication sans mémoire [SHA.49].

Si $\omega_v = 1$, le polymate est un automate stochastique réduit (en forme R) si sa matrice d'observation est une matrice d'agrégation invariante, [DEP.71.e.] : $A_v \equiv A, \forall v \in V$, et $a_i^{jh} \in \{0,1\}$, c'est une machine de Moore-Markov (en forme S : Etat/Sortie), c'est-à-dire une chaîne de Markov hétérogène observable par agrégation [DEP.68]. Le paramètre ω_v est le coefficient de pondération des décisio et évènements dans une action stochastique.

C.Q.F.D.

THEOREME. II.8. Tout polymate stochastique fini admet pour forme canonique une forme S (observable par une agrégation A') même si sa matrice d'observation dépend de v ∈

Démonstration. Il suffit de montrer qu'il existe toujours un changement de base K_v (Théorème II.5) généralisé, tel que $K_v A' = A_v$ où A' est une matrice d'agrégation et K_v admet une inverse généralisée à droite K^D, ∀ v ∈ V, tel que le polymate obten soit équivalent et stochastique [FAU.74.f.].

(i) Si cette propriété est vraie pour un automate stochastique fini, elle demeure pour un polymate stochastique fini à condition que K^D soit une matrice d'agrégation en effet, le passage de l'automate au polymate se fait en remplaçant $\{B_v e_v = e_n\}$ par $\{B_v e_n = \omega_n e_n$ et $C_v e_n = (1 - \omega_v) e_m\}$. Or l'existence d'une forme canonique équivalente requiert seulement que le changement de base préserve le caractère stochastiq de la représentation : $B_v e_n = \omega_v e_n \Rightarrow K_v B'_v K^D e_n = \omega_v e_n$. Puisque K^D est une matrice d'agrégation $K_v B'_v K^D e_n = K_v B'_v e_n$, et puisque K_v est nécessairement une matrice de dispersion [DEP.68] : $K_v B'_v e_{n'} = B_v e_n = \omega_n K_v e_{n'}$.

(ii) Pour montrer que cette propriété est vraie pour un automate, il suffit de prouver les équivalences suivantes entre formes d'automates :

α) il existe un algorithme pour passer d'un automate Σ_R en forme R à un automate similaire Σ_T en forme T (Machine de Mealy-Carlyle: transition/sortie), où les expériences x(t) sont représentées par les matrices de probabilités conditionnelles $p(y', \eta' | v, \eta)$.

β) il existe un algorithme de conversion inversible pour passer d'un automate Σ_T à un automate similaire Σ_S en forme S (Machine de Moore-Markov : état/sortie).

γ) il existe une transformation pour réduire la forme Σ_S à la forme Σ_R, réduite, si et seulement si une condition de réduction est satisfaite.

(iii) Soit Z l'espace d'état de Σ_R et Q celui de Σ_S. Soit Y l'alphabet de sortie commun, avec $|Y| = p$. Il suffit alors de prendre pour A' la matrice d'agrégation $[I_1 \ldots I_n]^t$, où les n matrices d'identité I_j sont de dimension $p \times p$, pour K^D la matrice d'agrégation $[N_1 \ldots N_n]^t$, où les n matrices de restauration N_j ont pour seule colonne non nulle la j-ème égale à e_p, et pour K_v la matrice $[M_1 \ldots M_n^j]$, où les n matrices de dispersion M_j ont pour seule ligne non nulle la j-ème égale à la j-ème ligne de A_v.

Alors $K_v K^D = I_n, K^D K_v = J_{n'}(v)$ est un idempotent, soluble entre les B'_v et A', et $K_v A' = A_v$, tel que si λ strates de B' sont égales et correspondent à la i-ème stra de B, alors les costrates de mêmes indices sont multiples dans les rapports $a_i^1(v) \ldots a_i^\lambda(v)$: c'est la condition de réduction. C.Q.F.D.

COROLLAIRE.a) Les états similaires sont liés par les relations $\eta K_v = q$ et $qK^D = \eta$ avec $K_v B'_v K^D = B_v$ et $K_v A' = A_v$; de même $C'_v K^D = C_v$, avec $q \in Q$ et $\eta \in Z$.

b)La transformation (γ) fait correspondre à la forme minimale de Σ_S la forme minimale de Σ_R si et seulement si, pour tout composant nul $a_i^j(v)$ de la matrice d'observation A_v, les $[(i-1)p + j]$-èmes ligne de K^D et colonne de K_v sont supprimées.

c) $n \leqslant n' \leqslant np$ [Voir DEP.68].

Exemple. Soit $\Sigma_R \triangleq \left(B_v = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/3 & 0 & 2/3 \\ 1/4 & 1/4 & 1/2 \end{bmatrix}, \; A = \begin{bmatrix} 1/2 & 1/2 \\ 0 & 1 \\ 2/3 & 1/3 \end{bmatrix} \right)$, c'est

un automate stochastique réduit minimal. Les coefficients des tenseurs B et A représentent les probabilités conditionnelles de transition :

$$b_i^j = p(\eta_j | \eta_i, v) \text{ et de sortie : } a_i^h = p(y_n | \eta_i).$$

l'algorithme (α) l'associe à $\Sigma_T = \left(x^{(1)} = \begin{bmatrix} 1/4 & 0 & 0 \\ 1/6 & 0 & 4/9 \\ 1/8 & 0 & 1/3 \end{bmatrix}, \right.$

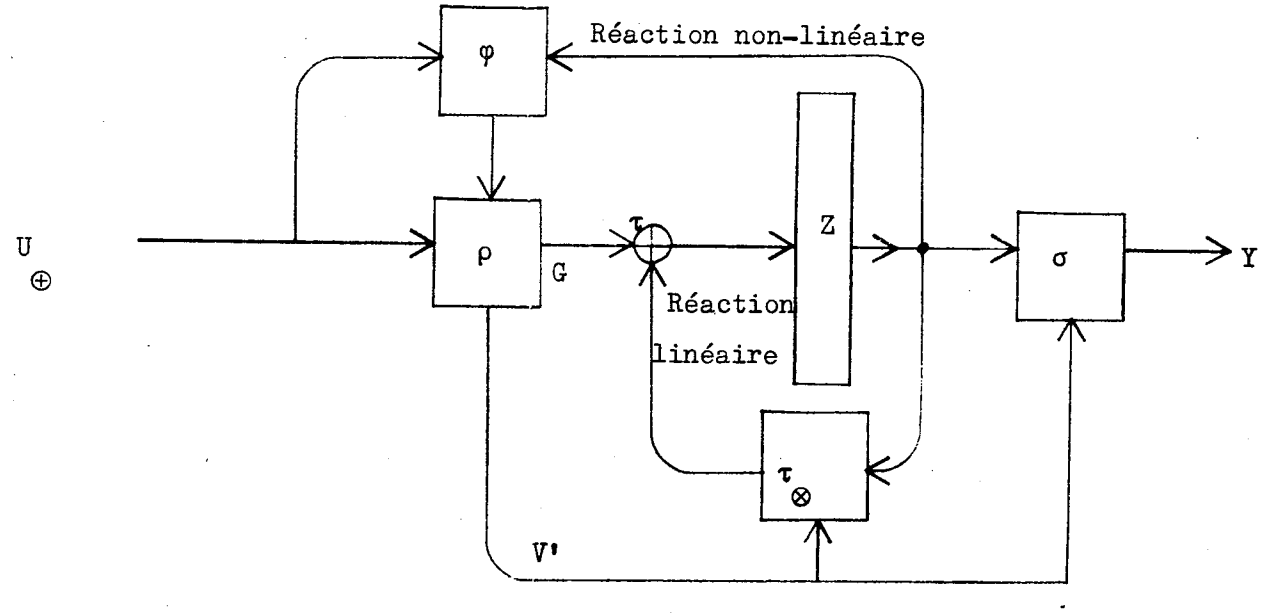
$\left. x^{(2)} = \begin{bmatrix} 1/4 & 1/2 & 0 \\ 1/6 & 0 & 2/9 \\ 1/8 & 1/4 & 1/6 \end{bmatrix} \right)$, similaire, avec $X = (Y|V)$; c'est un automate stochasti

minimal de Shannon. Les coefficients du tenseur X représentent les probabilités $x_i^{j(h)} = p(\eta_j, y_h | \eta_i, v) = p(\eta_j | \eta_i, v) p(y_h | \eta_i)$. L'algorithme de conversion (β) inversible l'associe à :

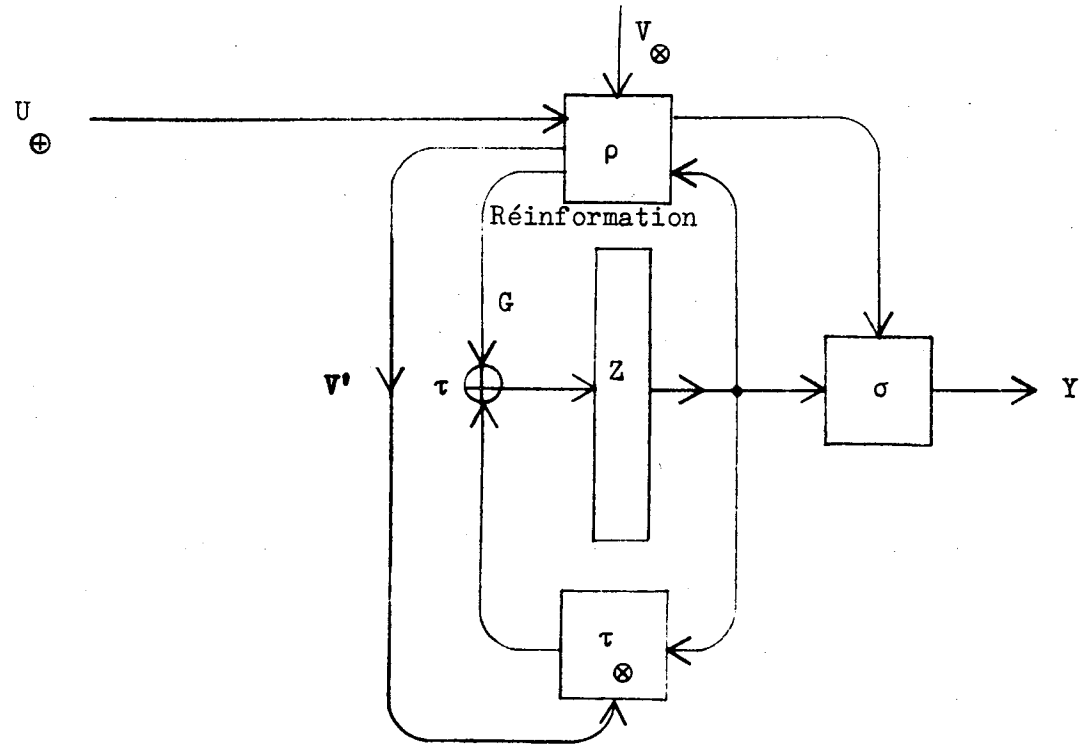
$$\Sigma_S = \left\{ B'_v = \left[\begin{array}{cc:c:cc} 1/4 & 1/4 & 1/2 & 0 & 0 \\ 1/4 & 1/4 & 1/2 & 0 & 0 \\ \hdashline 1/6 & 1/6 & 0 & 4/3 & 2/9 \\ 1/8 & 1/8 & 1/4 & 1/3 & 1/6 \\ 1/8 & 1/8 & 1/4 & 1/3 & 1/6 \end{array}\right], \; A' = \left[\begin{array}{cc} 1 & 0 \\ 0 & 1 \\ \hdashline 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{array}\right] \right\}$$

Figure II.3. Réinformation sur les polymates.

a)Pour la non-linéarité.



b)Pour la synchronisation (mode F).



similaire, avec $Q \subset Z \times Y$; c'est un automate stochastique minimal de Markov.

La transformation (γ) de réduction est donnée par le changement de base K =

$$\left[\begin{array}{cc|c|cc} 1/2 & 1/2 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 2/3 & 1/3 \end{array}\right] \quad \text{avec } K^D = \left[\begin{array}{c|c|c} 1 & 0 & 0 \\ 1 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 1 \\ 0 & 0 & 1 \end{array}\right]$$

Une forme simplifiée de ces transformations est utilisée plus loin au paragraphe V.2.2 dans le cas déterministe. Le lecteur trouvera en [DEP.73.a] des applications de ce modèle stochastique à l'analyse et à la synthèse des cana de communication.

II.2.3. Machines asynchrones.

Les modèles de machines synchrones peuvent être utilisés pour les syst asynchrones si la fonction de rythme ρ est employée à bloquer ou accepter les en trées selon que l'état interne est stable [†] ou non, c'est-à-dire si la transition précédente $\{\eta(t) \to \tau[\eta(t), w(t)]\}$ est achevée. Soit Z_S l'ensemble des états stab alors : $\rho : U \times V \times Z_S \times T \to U \times V'$. (†) Stable comme en I.3.2 et non comme en II.1.4.

Il convient en général de considérer que l'entrée additive U est de ty niveau ("level" : L) tandis que l'entrée multiplicative V est de type impulsion ("pulse" : P) ; cela permet de réaliser la fragmentation d'un événement d'entrée $u(t)$ en une séquence d'événements de sortie $\{y(t), y(t + \Delta), \ldots, y(t + \lambda\Delta)\}$, c'es à-dire la conversion des mots $\tilde{u}$ d'un langage U* en phrases plus explicites d'un langage Y* moins compact, en fonction des variables conditionnelles $v \in V$.

Il est possible aussi d'inverser les rôles des décisions et des événements ; selon que l'entrée multiplicative V représente les instructions ou les données pour un système de calcul, la programmation d'un polymate est dite dirigé par le programme ("program driven") ou par les données ("data driven"). Un regain d'intérêt se fait sentir pour l'étude des machines asynchrones car elles permette d'aborder la modélisation non seulement du matériel mais aussi du logiciel ; le lecteur intéressé par des résultats dans ce domaine peut consulter le paragraphe IV.6.2 puis la référence [DEP.74.a].

II.3. IDENTIFICATION ET TRAITEMENT EN TEMPS REEL.

Un système linéaire Σ a une entrée additive $U(t)$ induisant la sortie $Y(t)$, lorsque l'état initial $\eta(0)$ est l'état de repos, selon la relation :

$$y(t) = \sum_{j=0}^{t-1} u(j)CB^{t-j-1}A, \qquad (9)$$

tandis qu'un automate a un alphabet d'entrée multiplicatif $V(t)$ induisant la sortie $y(t) = \eta(i) \prod_{j=1}^{t-1} B_{v(j)}A$ où $\eta(j)$ représente l'état au temps j.

Or la déconvolution [DEP.70.c] est la transformation des données d'entrée-sortie du système linéaire Σ induisant comme indiqué sur la figure II.4. :

$$T(i) = \sum_{j=0}^{i-1} \delta(j)CB^{i-j-1}A, \qquad (10)$$

où $\delta(j)$ représente une impulsion unité (de Dirac), de sorte que :

$$T(i) = \eta(1)B^{i-1}A, \qquad (11)$$

où $\eta(1) \triangleq \delta(j)C$ est l'état initial de l'automate à une seule décision dont $T(i)$ est la sortie.

Par conséquent, la déconvolution est la transformation des expériences (relations Entrée-Sortie) d'un système linéaire qui permet de l'étudier comme un automate, en ne considérant que le monoïde multiplicatif de sa dynamique et le débarrassant de la linéarité, c'est-à-dire des entrées additives U. Elle est dite en temps réel si $T(i)$ est calculé dès que $u(i)$, $v(i)$ et $y(i)$ sont fournis.

L'identification d'un polymate Π à alphabet d'entrée multiplicatif V de cardinalité q consiste à évaluer les réponses impulsionnelles de Π pour tout un programme donné $\tilde{v} \in V^*$, c'est-à-dire à évaluer la fonction d'activité $f(0,v)$ pour tous les préfixes v de $\tilde{v}$, c'est-à-dire les sous-chaînes v telles que $\tilde{v} = vv'$.

THEOREME. II.9. Soit $T[i|\tilde{v}v(i)]$ la réponse impulsionnelle au temps i du polymate dirigé par le programme $\tilde{v} = v(0)\ldots v(i-1)$, et soit $y[i|\tilde{v}v(i)]$ la sortie correspondante pour une séquence d'événements $\tilde{u} = u(0)\ldots u(i-1)$; alors la réponse impulsionnelle peut être obtenue par la formule de déconvolution :

$$T[i|\tilde{v}v(i)] = y[i|\tilde{v}v(i)]/u(0) - \sum_{j=1}^{i-1} T[j|\tilde{v}_{]j}]v(i)]u(i-j)/u(0) \qquad (12)$$

où $\tilde{v}_{]j]}$ est le j-ème suffixe de $\tilde{v}$, c'est à dire la sous-chaîne des $(j-1)$ dernières décisions de $\tilde{v}$.

Démonstration. Il suffit de démontrer ce résultat par itération au moyen de la for (7). D'abord $y(0) = 0$ et $y[1|\, v(1)] = u(0)C_{v(0)}A_{v(1)}$ entraînent que

$$T[1|\, v(0)\, v(1)] = y[1|\, v(0)\, v(1)]/u(0) = C_{v(0)}A_{v(1)}.$$

Puis

$$y[2|\, v(0)\, v(1)\, v(2)] = u(0)\, C_{v(0)}B_{v(1)}A_{v(2)} + u(1)C_{v(1)}A_{v(2)},$$

donc

$$T[2|v(0)v(1)v(2)] = \{y[2|v(0)v(1)v(2)] - T[1|v(1)v(2)]u(1)\}\ /u(0),$$

et ainsi de suite :

$$T[3|\tilde{vv}(3)] = \{y[3|\tilde{vv}(3)] - T[2|v_{]2]}v(3)]u(1) - T[1|v(3)]u(2)\}\ /u(0).$$

C.Q.F.D.

Il apparaît que l'identification d'un polymate dont $|V| = q \geq 2$ ne peut généralement être effectuée en temps réel, car pour déconvoler $T[i\,|\tilde{vv}\,(i)]$ toutes les pré-expériences $y[j \mid \tilde{v}_{]j]}v(i)]$ pour $j \in \{1, \ldots, i-1\}$ s'avèrent nécessaires. Ainsi la déconvolution requiert la connaissance non seulement d'une branche $\tilde{vv}(i)$ mais d'un sous-arbre $\{\tilde{v}_{]j]}v(i)\}$ de l'arbre des expériences, en génér

THEOREME. II.10. Dans le cas d'une expérience $\tilde{vv}\,(i)$ périodique, la déconvolution peut être effectuée en temps réel après la première période $\tilde{v}_p$.

Démonstration. Soit $\tilde{vv}\,(i) = (\tilde{v}_p)^k \tilde{v}_h v(i)$ avec $i = pk + h + 1$ et $p = |\tilde{v}_p|$ (longueur de la chaîne) et $|\tilde{v}_h| = h$.

Une fois que les pré-expériences requises pour la déconvolution de $T(p\,|\tilde{v}_p)$ ont été effectuées, le polymate peut être considéré comme un système linéaire à une seule variable de décision $\tilde{v}_p$ au moyen du changement de variable d'événement :

$$\tilde{u}_p = \sum_{k=1}^{p} u(k)\, T[p \mid (\tilde{v}_p)_{]k]}]. \qquad (13)$$

Un algorithme pour l'identification en temps réel peut alors être appliqué suivi de la fragmentation de chaque période en séquences de longueur p.

C.Q.F.D.

Le modèle d'automate créé pour l'Informatique s'est vu surtout appliquer en Automatique ; un sort comparable pouvait être réservé au modèle de polymate dans la mesure où la première application rencontrée (présentée sur la figure II.4.) s'appliquait à une technique d'identification de système linéaire en temps réel. Le lecteur intéressé par cet aspect peut le consulter en référence [DEP.70.c].

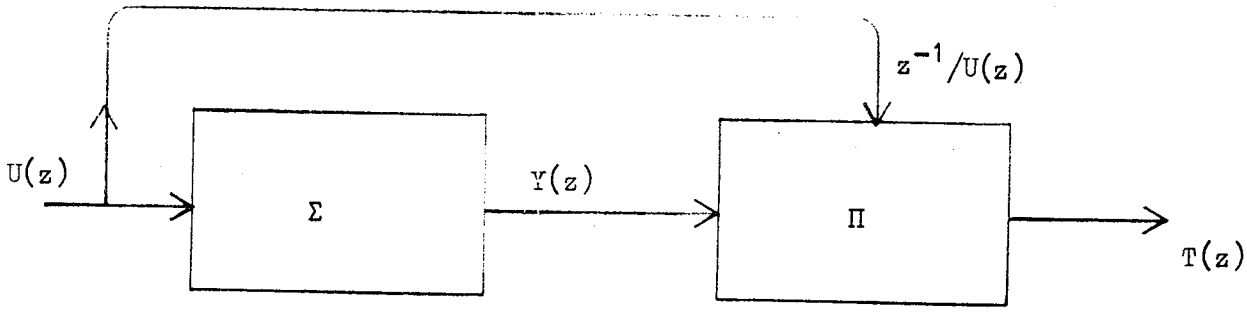


Figure II.4. Le polymate Π en cascade pour déconvolution en temps réel.

Ce chapitre est la démonstration que l'outillage mathématique efficacement mis en oeuvre en théorie du Contrôle par exemple (matrices, groupes, etc.) s'avère bien adapté à la manipulation des structures Informatiques une fois celles-ci formalisées à l'aide du modèle de polymate.

Mais ce modèle n'a été appliqué pour l'instant qu'au "niveau fin" des machines séquentielles ; les deux chapitres suivants ont donc pour but de le porter au "niveau-système", d'abord du point de vue algorithmique et architectural, puis du point de vue fonctionnel et topologique (PMS).

CHAPITRE III

DE LA MACHINE DE TURING A CELLE DE VON NEUMANN

CHAPITRE III

DE LA MACHINE DE TURING POLYADIQUE A CELLE DE VON NEUMANN

Le chapitre précédent introduit un modèle nouveau, le polymate, selon la démarche des "Mathématiques Appliquées", c'est-à-dire par un raisonnement abstrait sur des modèles préexistants. Ce paragraphe effectue le mouvement inverse selon la démarche de la "Théorie des Systèmes" : partant du phénomène dynamique objet de notre étude, le Système Informatique, il en extrait les caractéristiques essentielles pour en dégager une gamme de modèles où se retrouve le polymate. Cette démarche-ci est la seule démonstration de l'adéquation du modèle au problème. Les paragraphes et chapitres ultérieurs tiennent lieu de démonstration de l'utilité du modèle et de la méthodologie qu'il a aidé à développer.

III.1. LANGAGE VERTICAL ET LANGAGE HORIZONTAL.

Un langage de programmation L peut être défini (de manière dite externe) comme l'ensemble des programmes P réalisables sur une catégorie donnée de machines C.

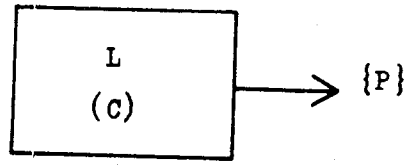


Figure III.1. Langage vu comme une "source" de programmes.

Noter que les termes "machine", "programme" et "réalisable" sont employés ici au sens intuitif et que l'élaboration d'un système axiomatique rigoureux complet n'est pas l'objet de cette dissertation. Plus précisément, cette thèse considère comme acquis cet ensemble de notions de base présenté dans [ARB.69] : "Theories of Abstract Automata" et dans [BEL.71] : "Computer Structures : Readings and examples", qui incluent respectivement les descriptions des modèles de TURING et de VON NEUMANN.

Par contre, le lecteur intéressé par une approche formelle des problèmes abordés dans ce chapitre les trouvera analysés en [DEP.74.a] : "Duality and Hierarchy in Data Processors".

La programmation donc est l'Art de spécifier un algorithme en terme d'opérations élémentaires à exécuter séquentiellement. Elle est la généralisation aux fonctions récursives de la "réalisation" des fonctions séquentielles [RAN.58 BAN.74], mais la possibilité pour un algorithme de mémoriser les entrées et sorti antérieures rend inefficace l'application d'une technique comme celle de l'équiva lence de NERODE [58]. C'est-à-dire que le passage d'une liste de relations entr sortie (relations externes) à une liste d'actions conditionnelles (modèle interne n'est pas automatisable. Cependant, la formalisation du procédé de programmation d'un algorithme depuis avant même l'avènement des calculateurs, est disponible s la forme d'une machine abstraite simple [TUR.1936]. Il s'agit d'un triplet $C \triangleq (K,D,M)$ où K est un automate de commande, D est une unité de lecture-écriture et M est un ensemble de cases-mémoire en nombre potentiellement infini, disposées linéairement (en forme de bande), chaque case pouvant enregistrer un symbole d'un alphabet fini V. A tout instant D est placé sur une case de M et, sous contrôle d K, peut lire le symbole enregistré sur cette case, écrire un nouveau symbole à la place et aussi se déplacer d'une case dans un sens ou dans l'autre. L'automate de commande K a pour alphabet d'entrée V, et pour fonction de sortie la combinais des ordres d'écriture et de déplacement d de D ; de sorte que l'ensemble de sorti est $Y \triangleq (V, d)$. L'automate-opérateur D compose avec K un processeur P = (K,D) comme indiqué sur la figure III.2.

L'algorithme de K est donc défini par sa table des états à laquelle est donnée la forme équivalente d'une liste de quintuples

$$(\eta, v : v', d, \eta') \qquad (14)$$

exprimant que si K reçoit le symbole v en l'état η, il fait inscrire le symbole v' sur M, déplacer D de d et transite à l'état η'.

Définition.

Un langage horizontal (ou fonctionnel) comprend des programmes ayant la structure de listes non ordonnées de règles spécifiées par des paires (Condition → Action).

La machine de TURING, qui est définie par son automate avec une liste non ordonnée de quintuples pour décrire un algorithme, correspond à cette notion ; c'est un langage proche d'une implantation cablée ou microprogrammée mais qui se prête mal à la formalisation des programmes d'utilisateurs de machines. C'est

pourquoi WANG [57] a introduit une variante de ce modèle introduisant la notion de programme séquentiel telle qu'elle était apparue entretemps avec les calculateurs électroniques. Cette variante a la même structure, mais l'algorithme de K est décrit à l'aide des six instructions suivantes et prend la forme d'une liste ordonnée de ces instructions.

Définitions.

1) Une étiquette est un nombre ordinal (par exemple une adresse) permettant de fixer l'ordre d'exécution des actions dans un programme.

2) Le code d'instructions U de la machine de WANG est l'ensemble des six instructions suivantes :

I_D { e : ordre donné à D d'effacer le symbole lu en M, (écrire 0)
m : ordre donné à D de marquer un symbole sur M, (écrire 1)

d { + : ordre de déplacement d'une case vers la droite,
- : ordre de déplacement d'une case en sens inverse,

I_K { t(n) : instruction de branchement conditionnel ; K doit exécuter l'instruction suivante si D lit le symbole v_0 sur M, sinon K doit exécuter celle d'étiquette n.
s : instruction d'arrêt de K.

3) Ces six instructions forment trois paires : instructions d'assignation (I_D) de valeur et d'adresse (d), et instructions de commande (I_K), respectivement

Définition.

Un langage vertical (ou procéduriel) comprend des programmes qui ont la structure d'une liste ordonnée d'instructions spécifiées par des triplets :
-(étiquette, accès d'opérande, opération) pour les instructions d'assignation (ou d'exécution I_D).
-(étiquette, condition, transition conditionnelle vers une étiquette) pour les instructions de contrôle (ou de séquencement I_K). Dans tous les cas autres que celui d'un branchement conditionnel dont la condition est satisfaite, la transition se fait séquentiellement vers l'étiquette suivant celle de l'instruction en cours.

Dans la pratique [WIL.51] les langages horizontaux sont surtout appliqués aux microalgorithmes, et les langages verticaux aux algorithmes.

La machine de WANG correspond à cette notion qui est plus "naturelle pour les programmes d'utilisation de machines informatiques.

III.2. ADRESSAGE ET MACHINES UNIVERSELLES POLYADIQUES.

Le résultat essentiel de la thèse de TURING est qu'il existe une machine universelle au sens où, trouvant sur la bande de mémoire M la descriptio $\bar{u}$ d'un programme quelconque P_u et la séquence $\tilde{v}$ des données initiales, cette machine effectue le traitement $P_u(\tilde{v})$. Puis WANG a démontré que les six instruction de sa machine [1957] forment un code de programmation U universel, c'est-à-dire que tout programme de TURING $\bar{u}$ peut être remplacé par un programme de WANG $\tilde{u} \in U$ où U* est l'itéré de U.

Le but de ce paragraphe est de montrer que la distinction entre deux types d'entrée, considérée (§. I.2) comme la notion importante à ajouter à la notion d'automate, trouve son fondement dans la nature même des machines universelles à programme enregistré $\bar{u}$.

En différentiant (dualité) le code d'instruction U de la représentati des données V telle que $\tilde{v} \in V^*$, puis en introduisant une hiérarchie d'algorithme enregistrés (verticaux) et implantés (horizontaux) tout en spécifiant les mécanismes de synchronisation des divers composants, nous présentons dans le théorème suivant une machine algébrique C formalisant de manière réaliste la structure des calculateurs de type VON NEUMANN. Le lecteur peut consulter en [DEP.74.d] la synthèse de ces concepts et leurs applications à la conception des Interfaces dans les systèmes Informatiques.

Définition.

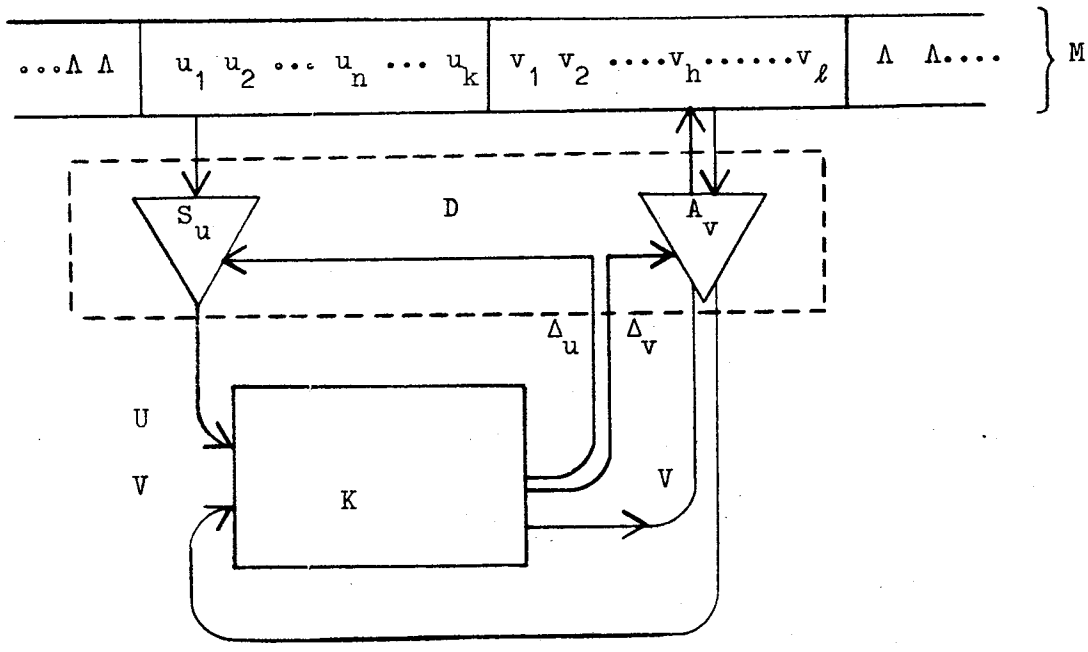
Une machine de TURING polyadique C est composée d'un ensemble de m machines de TURING non-universelles K_i, chacune étant initialisable par une instruc u_i de l'alphabet de commande U (code de programmation), $i = 1,\ldots,m$.

Chaque composante K_i réalise le microalgorithme P_i à partir d'un état initial unique x_i et l'ensemble K est doté d'un état de repos x_0.

THEOREME III.1.

Tout programme $P_u^{\sim}$ de machine de WANG peut être exécuté par une machine polyadique C universelle comprenant un mécanisme d'accès S_u pour les instructions, un mécanisme d'accès A_v pour les données sur M, et une unité de commande K effectuant l'interprétation et la synchronisation des accès.

Figure III.2. Machine C à programme enregistré.



Notes :

1) Puisque K a pour espace d'état le produit Cartésien des espaces d'état $\{\eta_i\}$ composants K_i, ses états seront ici dénotés x_j.

2) Le sélecteur d'instruction S_u disparaît dans le cas d'une machine $C \triangleq (K,D,M)$ non universelle avec un seul $K_i \equiv K$; m = 1.

3) Du moment que M est potentiellement infinie, l'alphabet est toujours $V' \triangleq V \cup \Lambda$ où Λ représente "un blanc".

4) La notion de programme $P_u(\tilde{v})$ est la généralisation par récursion de la notion de fonction d'activité du paragraphe II.1.4. iii.

Démonstration. Rappelons que les six instructions de WANG sont l'écriture de zéro (e), le marquage de un(m), le décalage positif d'adresse (+), son inverse (- le branchement à l'étiquette n sur lecture de 1 (t) et l'arrêt (s).

Quant aux règles de TURING, ce sont des quintuples (x,v:v',d,x') tels que si l'état de K est x et la tête de lecture est sur v,alors la tête d'écriture inscr v', les mécanismes d'accès se déplacent de d avec $d \in \{-1,0,+1\} = \Delta$ et K transite vers l'état x'.

Chacune des instructions d'une machine de WANG peut être simulée par l'un des microprogrammes d'une machine de TURING. Mais pour que celle-ci soit universell il faut que la machine C soit indépendante du programme $P_u^{\sim}$. Il faut donc que C mémorise l'étiquette n de l'instruction en cours d'exécution. Il suffit pour cela de doter le mécanisme d'accès S_u d'un registre SI appelé compteur ordinal dont le contenu soit n et de doter l'unité de commande K d'une fonction de sortie gérant la synchronisation entre les deux niveaux de programme par le biais du mécanisme d'acc S_u.

Pour un programme $P_u^{\sim} = u_1\ u_2 \ldots u_n \ldots u_k$, l'unité de commande K procède com suit : au début le contenu de SI est n = 1, de sorte que K reçoit l'instruction u_1 à l'état x_0, ce qui le fait transiter à l'état x_1 d'activation de K_1. Puis K exécute le micro-programme de TURING correspondant au champ-fonction du programme de WANG et met à jour SI, soit en incrémentant son contenu de (+1), soit en y força une nouvelle valeur entière positive $n \in N$, c'est à dire que K exécute un quintuple (X,V:V, (Δ,N),X) pour chacune des instructions de WANG comme suit :

$$
\begin{aligned}
e &: (x_e,0:0,(0,+1),x_0) \text{ ou } (x_e,1:0,(0,+1),x_0),\\
m &: (x_m,0:1,(0,+1),x_0) \text{ ou } (x_m,1:1,(0,+1),x_0),\\
+ &: (x_+,0:0,(+1,+1),x_0) \text{ ou } (x_+,1:1,(+1,+1),x_0),\\
- &: (x_-,0:0,(-1,+1),x_0) \text{ ou } (x_-,1:1,(-1,+1),x_0),\\
t(n) &: (x_t,0:0,(0,+1),x_0) \text{ ou } (x_t,1:1,(0,n),x_0),\\
s &: (x_s,0:0,(0,\infty),x_0), \text{ ou } (x_s,1:1,(0,\infty),x_0),
\end{aligned}
\qquad (15)
$$

où la paire (Δ,N) représente le déplacement des mécanismes d'accès A_v et S_u respectivement.

Noter que N est soit une adresse absolue (n, ou ∞ qui ici symbolise l'adresse absolue -1), soit une adresse relative (l'incrément +1). Cette définition donne à SI la fonction de remise à zéro de l'automate de commande K.

La machine C ayant exécuté l'instruction s de WANG termine sur le blanc Λ immédiatement à gauche du programme $P_u^{\sim}$ sur la bande M, mais après toute autre instruction, pour un programme correctement écrit, C se retrouve à l'état x_0, reçoit l'entrée suivante (u_2) et procède itérativement.

C.Q.F.D.

La longueur maximale k des programmes exécutables par une telle machine est limitée par la valeur du nombre maximal représentable dans SI pour N. Il est théoriquement possible de se défaire de cette limitation en programmant K de manière plus complexe de sorte que le contenu n de SI soit enregistré sur la bande de mémoire M qui prend alors le format de la figure

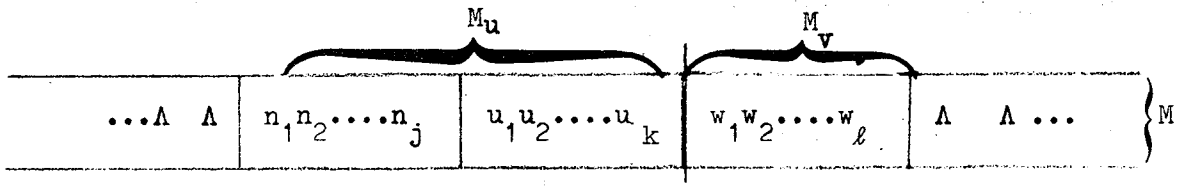


Figure III.3. Bande pour la machine polyadique C.

Inversement, il est plus pratique d'utiliser le même mécanisme d'accès que S_u pour les enregistrements sur M_u et M_v et de remplacer l'adressage autorelatif A_v par un adressage absolu au moyen d'un registre RA d'adressage de la mémoire pour les données. Ce registre est initialisé avec l'adresse h de v_1 puis modifié progressivement ($h' = h + d, \forall\, d \in \Delta$).

Par opposition au cas des automates ne réalisant que des fonctions séquentielles, [RAN.58], il faut noter que c'est la présence de mécanismes d'adressage des symboles dans les séquences d'entrée qui donne à un automate la puissance d'exécuter des algorithmes réalisant des fonctions récursives [TUR.36].

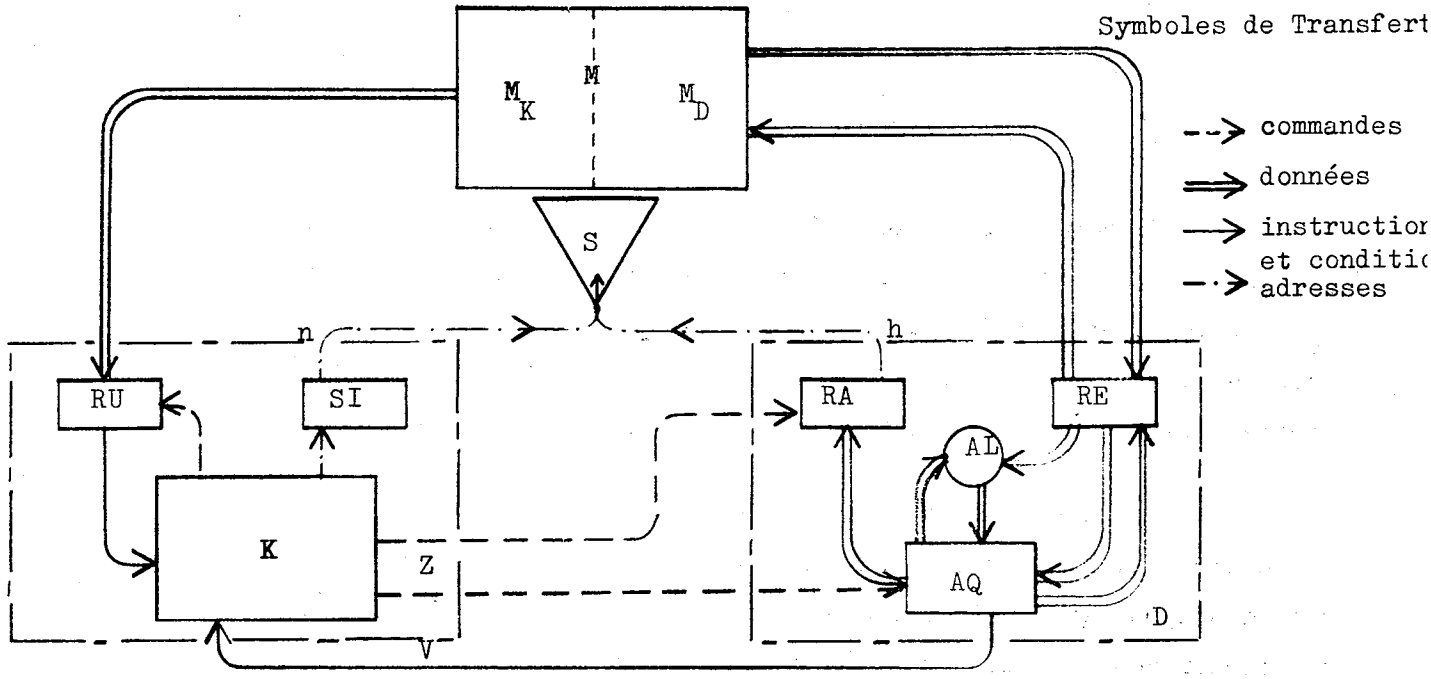
III.3. ARCHITECTURE SYMBOLIQUE D'UN CALCULATEUR ACTUEL.

La machine C du théorème de hiérarchisation est une machine-à-bits, alors que les calculateurs sont des machines-à-mots (qui sont en général des multiples d'octets, mots de 8 bits). Le modèle algébrique C peut être généralisé pour remédier à cela comme suit : la mémoire M est un ensemble ordonné de cellules capables de contenir chacune un mot à codage binaire ; le mécanisme d'accès est un circuit de décodage S sélectionnant une cellule adressée par le contenu n du registre SI pour les instructions, et le contenu h du registre RA pour les données.

De plus, l'instruction en cours d'exécution est mémorisée dans un registre RU et un registre accumulateur AQ est connecté au travers d'un opérateur arithmétique et logique AL (d'addition et de décalage, au moins) à un registre RE d'échange avec la mémoire, de sorte que l'unité de commande K fait effectuer des transferts [soit de mémoire à registre ou de registre à mémoire (entre RE et la cellule en M dont l'adresse est en RA ou SI), soit de registre à registre (entre RE, RA et AQ ou RU) (figure III.4)] à une unité de traitement D = {AQ,AL,RA,

(16)

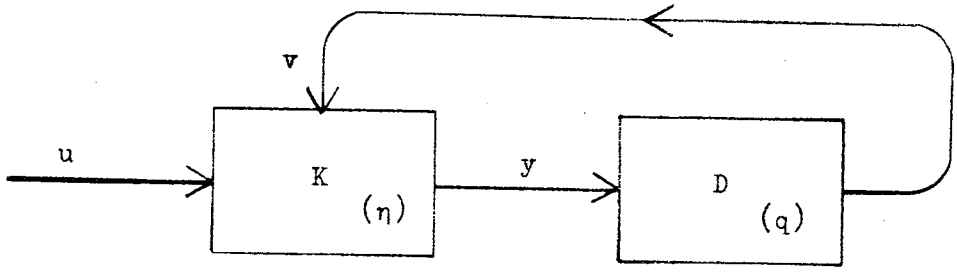
Figure III.4. Machine-à-mots à programme enregistré.



C'est usuellement la même unité physique de mémoire qui contient le programme, dans l'ensemble de cellules non nécessairement contigües M_K, et les données, dans l'ensemble M_D.

L'unité de commande K (incluant RU et SI) est dite connectée en tourbillon sur l'unité de traitement des données D = {AQ,AL,RE,RA} pour former le processeur P = (K,D).

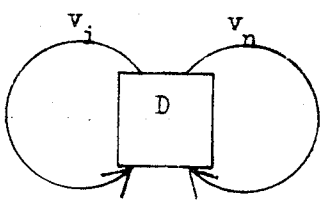
Figure III.5. Interconnexion remarquable de polymates : le tourbillon d'un process



THEOREME III.2.

Pour une machine de TURING binaire, D est réduit à une cellule unique avec un transfert identité(i)et un transfert inverseur(n)

Démonstration.



$\forall \eta \in H$, les quintuples $(\eta,y:y',d,\eta')$ avec $y,y' \in Y = \{0,1\}$ peuvent être réalisés par une bascule TRIGGER telle que l'action identité (v_i) est appliquée si $y = y'$ et l'action permutation (v_n) si $y \neq y'$. C.Q.F.D.

Figure III.6. L'unité-opérateur D d'une machine de TURING.

Nous retrouvons ainsi le besoin exprimé au §. I.3.6 d'un modèle de machine séquentielle D formalisant la notion de transfert de registres. Mais D étant dégénéré dans une machine de TURING, l'étude formelle de ce genre de machine séquentielle n'a pas été stimulé adéquatement.

Ce théorème va nous permettre de présenter sous un jour nouveau un théorème de SHANNON [1956] sur une certaine compensation entre K et D. Il n'est pas question ici d'analyser la nature de cette dualité (le lecteur peut consulter pour cela le document [DEP.74.a]), mais plutôt de lui donner un sens (une sémantiq grâce à la démarche suivie jusqu'ici.

THEOREME DE DUALITE.III.3.

Pour tout processeur P = (K,D) il existe un processeur P* équivalent, c'est-à-dire exécutant les mêmes algorithmes à un transcodage près des commandes et des données, mais tel que ce processeur P*, soit ait une seule bascule d'état da K, soit ait une seule bascule de registre dans D.

Démonstration.

Le théorème de SHANNON indique qu'une machine de TURING universelle peut être construite soit avec une bande mémoire M et un automate K ayant deux états η_0 et η_1 seulement, soit avec un automate K et une bande-mémoire M portant deux symboles v_0 et v_1 seulement. Il suffit alors de remarquer que dans le premie cas une seule bascule suffit à mémoriser l'état de K ; par contre dans ce cas, le nombre de symboles dans l'alphabet V étant accru, le registre RE (ou AQ) en D doi comprendre plus de bascules. Dans le second cas, une seule bascule suffit à transférer le symbole en D dans RE (ou AQ) ; par contre le nombre d'états de K sera accru et son implémentation requerra un plus grand nombre de bascules. C.Q.F.

Définitions.

1) Un transfert peut être translationnel (M ←RE) ou transformationnel (dans AL).

2) De même que pour les programmes, il y a trois représentations pour une opération élémentaire à deux opérandes (cf. §. II.2.3.) :

a. La méthode "dirigée par l'opération" qui déclenche le traitement des opérandes, enregistrés antérieurement et formant l'état initial d'un automate : (RE); (AQ).

b. La méthode "dirigée par les données" qui sont séparées en un opérande (AQ) chargé à l'avance et un opérateur (RE) agissant selon une opération définie par la structure interne (AL).

c. La méthode fonctionnelle où les opérandes sont envoyés chiffre à chiffre dans un automate qui émet séquentiellement les chiffres du résultat (traitement sériel).

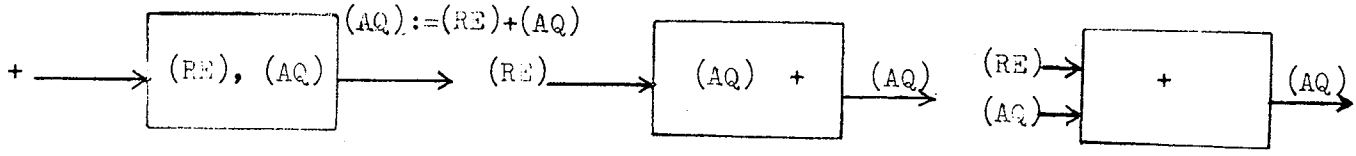


Figure III.5. Les trois modes de calcul (le contenu d'un registre R est noté (R)).

THEOREME III.4.

L'unité de traitement D peut être représentée par un polymate, selon la méthode dirigée par les données (b).

Démonstration.

Comme indiqué sur la figure III.6 , l'état du polymate η est le contenu du registre accumulateur AQ.

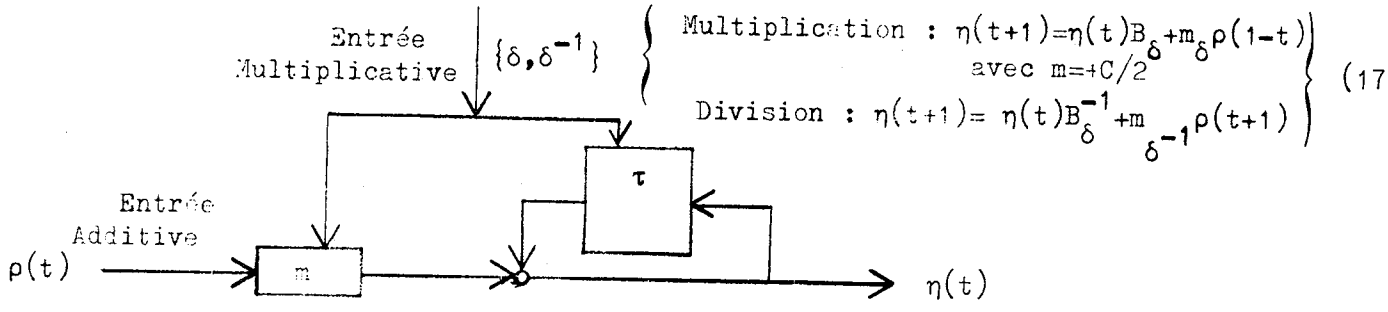


Figure III.6. Le polymate accumulateur D.

Les microinstructions dans K définissent l'alphabet Z qui contient en particulier l'opérateur de décalage à droite δ pour la multiplication et son inverse pour la division.

Pour un mot de données $\rho = [\rho(0), \rho(1)...\rho(l)]$ où $\rho(0)$ représente le signe et $\rho(i)$ le bit de poids 2^{-i}, l'action de δ est donnée par $\rho B_\delta = [\rho(0), 0, \rho(1), ...\rho(l-1)]$ et sa représentation matricielle est

$$B_\delta = \left[\begin{array}{c|cccc} 1 & 0 & 0 & \cdots & 0 \\ \hline 0 & 0 & 1 & \cdots & 0 \\ 0 & 0 & 0 & \ddots & 0 \\ \vdots & \vdots & \vdots & \ddots & \\ \vdots & \vdots & \vdots & \ddots & 1 \\ 0 & 0 & 0 & 0 & 0 \end{array}\right] \tag{16}$$

(18)

C.Q.F.D.

Cette approche de la représentation des machines à transfert de regist n'a cependant pas toute la généralité voulue car elle tient à la présence d'un additionneur pour la formalisation en polymate. Le lecteur peut consulter en [DEP.70.a] une application intensive de cette approche qui en révèle les limitatio ce qui amène à poursuivre la recherche d'un modèle mieux adapté que nous trouveron ici au chapitre suivant (§.IV.5).

III.4. UNITE DE COMMANDE ET CALCUL PARALLELE.

III.4.1. Le tourbillon.

La fonction de transition τ d'une unité de commande K d'un calculateur est destinée à fragmenter chaque instruction (u) en une séquence de micro-ordres exécutés par l'unité arithmétique D ; elle peut être spécifiée par des diagrammes de temps, par un organigramme ou par un système [IAN.58] de <u>formules de transfert</u> :

$$\tau(\eta_i^u, v_j) = \eta_j^u \ \forall j \in 1,...,|V| \ \Leftrightarrow$$

$$\eta_i^u \to v_{i,1}\eta_1^u + \cdots + v_{i,j}\eta_j^u + \cdots + v_{i,n}\eta_n^u, \tag{19}$$

où chaque η_i^u représente l'état de l'unité en codage primitif (1 parmi n) et par conséquent l'état de la i-ème bascule d'une réalisation cablée de K. L'unité de commande peut être considérée comme un polymate asynchrone fini synchronisé par réinformation (mode F : "feedback") à partir de l'unité d'exécution D. Plus générale ment, chacun des η_i peut représenter l'état d'un processus élémentaire d'un système de procédures enregistrées initialisables par u et où les $v_{i,j}$ sont des fonctions de prédicats (décisions) pour instructions conditionnelles.

Exemple : Soient deux instructions u_1 et u_2 à fragmenter en micro-instructions :

$u_1 \rightarrow (\eta_2^{u_1}\ \eta_1^{u_1}\ \eta_2^{u_1})$ et $u_2 \rightarrow (\eta_3^{u_2}\ \eta_2^{u_2}\ \eta_1^{u_2}\ \eta_3^{u_2})$, suivant une séquence de décisions $v_1\ v_2$ ou $v_1 v_2 v_3$.

Le polymate K est instantanément observable ($A = I_5$) et à commande invariante ($C_v = C$, $\forall v \in V$). Soit

$$= \{\eta_1^{u_1}, \eta_2^{u_1}, \eta_1^{u_2}, \eta_2^{u_2}, \eta_3^{u_2}\}$$

$u_1 = (1, -1)$
$u_2 = (-1, 1)$

$$C = \left[\begin{array}{c|cccc} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{array}\right] \text{ et } B_{v_1} = \left[\begin{array}{c|cccc} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{array}\right]$$

$\eta_2^{u_1} = (01000)$ et $\eta_1^{u_2} = (00001)$

$$B_{v_2} = \left[\begin{array}{c|cccc} 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{array}\right], \quad B_{v_3} = \left[\begin{array}{c|cccc} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{array}\right]$$

Les fonctions de condition $v_{i,j} = v_{ij}(q_1 \ldots, q_h, \ldots, q_k)$ sont définies par les variables booléennes q_h indépendantes issues du code de conditions des unités arithmétiques D associées au traitement de l'instruction par le polymate K tel que $\tau(\eta_i^u, q_h) = \eta_{ih}^u$ avec $u \equiv \eta_0^u$. Ces fonctions de condition satisfont les axiomes probabilistes sur les prédicats q_h :

$$v_{ij} v_{ig} = \delta_{jg} \quad \text{et} \quad \sum_{j=1}^{n} v_{ij} = 1 \text{ où } \delta_{ig} \text{ est le symbole de Kronecker.} \qquad (20)$$

L'organigramme de l'instruction u est alors obtenu en concaténant les symboles de lancement des opérateurs η_i^u et les prédicats q_h qui déterminent l'orientation des ruptures de séquence (branchement). Ensuite, il faut factoriser les q_h au sein des fonctions de condition v_{ij} puis les compléter par les adresses de branchement (étiquettes de jonction $\downarrow_h$). Le branchement vers h si q_i est faux est noté $q_i \downarrow_h$

Cette technique [DJA.68] revient à effectuer une décomposition booléenne des vecteurs v(t) du théorème II.3, en fonction des prédicats q_h venant de l'unité d'exécution par réinformation, ce qui permet d'obtenir sous forme linéaire une version souple du séquencement des opérations de l'unité cablée K. Cette démarche est précisément celle de la microprogrammation verticale amenant à traiter séquentiellement les diverses variables d'une expression booléenne.

Exemples. Il suffit de remplacer une expression $q_i q_j \oplus \bar{q}_i q_k$ par $q_i \uparrow_h q_j \oplus \downarrow_h q_k$.

$$\eta_i \to \bar{q}_1 q_2 \eta_1 \oplus \bar{q}_1 \bar{q}_2 q_3 \eta_2 \oplus \bar{q}_1 \bar{q}_2 \bar{q}_3 \eta_3 \oplus q_1 \eta_4,$$

$$\Leftrightarrow \eta_i \to \bar{q}_1 [q_2 \eta_1 \oplus \bar{q}_2 (q_3 \eta_2 \oplus \bar{q}_3 \eta_3)] \oplus q_1 \eta_4,$$

$$\Leftrightarrow \eta_i \to \bar{q}_1 \uparrow_5 q_2 \uparrow_6 \eta_1 \oplus \downarrow_6 q_3 \uparrow_7 \eta_2 \oplus \downarrow_7 \eta_3 \oplus \downarrow_5 \eta_4.$$

III.4.2. Extension possibiliste pour le calcul parallèle.

La technique de linéarisation ci-dessus ne permet de représenter qu'un algorithme séquentiel. Même si chaque sortie de K commande plusieurs transferts simultanés en D le traitement est séquentiel. Le passage du modèle de traitement séquentiel au polymate de commande de traitements collatéraux se fait en considérant que le codage des états n'est plus primitif (sa représentation vectorielle n'est plus monomiale), mais que plusieurs processus élémentaires fonctionnent concurremment, d sorte que le polymate K est possibiliste, c'est-à-dire qu'à un état initial la fonction de transition τ fait correspondre un état complexe dans 2^Z et non Z (le vecteu d'état $\eta(t+1)$ est polynomial). Il y a alors indépendance entre les transferts parallèles en D. Dès lors une représentation spatiale (à trois dimensions) permet de disposer les chemins simultanés sur des plans parallèles contenant chacun les graphes des tâches séquentielles. Une formalisation par règles grammaticales est présentée en [DEP.74.a,d].

Mais la généralisation du formalisme de linéarisation [DJA.68] des organigrammes requiert l'introduction de symboles étiquetés ∂ de diffusion et $\int$ de fusion jouant le rôle de parenthèses de synchronisation (TIME BEGIN, TIME END en mode H) pour définir des blocs de temps. Ces symboles sont usuellement appelés FORK et JOIN. Le AND introduit par WIRTH [66] pour paralléliser ALGOL est alors remplacé par : $\langle \partial_i \ldots + \ldots \int_i \rangle \equiv \langle ; \ldots \text{ AND } \ldots ; \rangle$. Le QUIT est représenté par le symbole ∞ d'in-fini. Des applications de ces concepts sont présentées en [DEP.74.a et d]. Cette généralisation du formalisme de linéarisation permet d'en assurer le rapprochement avec les primitives du logiciel d'une part, les représentations matricielles d'autre part (et donc avec la microprogrammation horizontale

En effet, l'état x d'un processus de calcul étant représenté par le vecte d'état de D, si le processus contient des phases exécutables en parallèle, alors l'emploi du calcul parallèle correspond à la décomposition en produit direct de x. [DEP.69]. Ce problème est développé plus loin (§.V.1).

Considérons un processus comme l'exécution d'une séquence de décisions.

Exemple : Soit le processus inconditionnel (sans réinformation) w'ww". Soit w décomposable en phases parallèles v_1 et v_2. La formule linéarisée s'écrit alors $x \to xw'\partial_3 v_1 + v_2 \int_3 w"$. Le processus agit sur les entrées multiplicatives de D.

$$\text{Soit } B_w = B_{v_1} \times B_{v_2} = \begin{bmatrix} \alpha_1 & \beta_1 \\ \gamma_1 & \delta_1 \end{bmatrix} \times \begin{bmatrix} \alpha_2 & \beta_2 \\ \gamma_2 & \delta_2 \end{bmatrix} = \begin{bmatrix} \alpha_1\alpha_2 & \alpha_1\beta_2 & \beta_1\alpha_2 & \beta_1\beta_2 \\ \alpha_1\gamma_2 & \alpha_1\delta_2 & \beta_1\gamma_2 & \beta_1\delta_2 \\ \gamma_1\alpha_2 & \gamma_1\beta_2 & \delta_1\alpha_2 & \delta_1\beta_2 \\ \gamma_1\gamma_2 & \gamma_1\delta_2 & \delta_1\gamma_2 & \delta_1\delta_2 \end{bmatrix}$$

PROPRIETES. III.5.

Pour tout processus décomposable, il vient :

i)Commutativité : $\exists P \Rightarrow B_{v_1} \times B_{v_2} = P(B_{v_2} \times B_{v_1})P^{-1}$. (21)

ii)Action sur place : x est décomposable en x_1 et x_2 avec $x = x_1 \times x_2$ et $x\,w = x_1 v_1 \times x_2 v_2$. (22)

iii)Transposition : $(B_{v_1} \times B_{v_2})^t = B^t_{v_1} \times B^t_{v_2}$. (23)

iv) Suite à la définition introduite au paragraphe II.2.3. :

Fragmentation : $B_w = \bar{B}_{v_1} \bar{B}_{v_2}$ avec $\bar{B}_{v_1} = (B_{v_1} \times I)$ et $\bar{B}_{v_2} = (I \times B_{v_2})$ (24)

Cette dernière propriété s'interprète ainsi : deux phases de calcul parallèles peuvent toujours être exécutées successivement ; mais elle révèle aussi une relation entre la fragmentation et la décomposition. Le lecteur peut consulter en [DEP.69] une application de cette représentation matricielle de la décomposition pour le découpage du matériel implémentant un automate de commande, et en [DEP.70.b et 71. a et b] son application systématique à l'accélération et à la parallélisation des algorithmes de Transformation de Fourier Discrète.

Mais ce modèle matriciel manque encore trop de sémantique pour être aisément utilisable en toute généralité ; c'est pourquoi nous allons l'enrichir au chapitre suivant des primitives architecturales de PMS avant d'approfondir aux paragraphes IV.4 et IV.5 les problèmes de parallélisme et de décomposition. Ensuite il sera efficace au chapitre V de se limiter à la représentation matricielle pour atteindre l'objectif d'une nouvelle conception des unités de commande K.

III.5. INTRODUCTION DU COMPLEXE DE TURING.

Le raisonnement du paragraphe III.3 (figure III.4) nous a conduit à un schéma réaliste de processeur central d'un calculateur actuel simple. Avec la mémoire et les liaisons d'entrée-sortie, P forme une unité centrale :

$$C := (P, M, L_i, L_e) \qquad (25)$$

Pour avoir un système complet, il faut ajouter un pupitre T permettant la mise en marche et l'arrêt du système ou la remise à zéro des registres centraux, le chargement en mémoire et le lancement des programmes, et il faut ajouter des périphériques E (mémoires externes ou transcodeurs de liaison avec l'extérieur) et leurs automates-contrôleurs (figure III.2).

Nous avons ainsi reconstitué progressivement un calculateur de fonctionnement équivalent à celui décrit en 1946 par J. VON NEUMANN et qui caractérise encore les principes de conception des systèmes actuels [BEL.71].

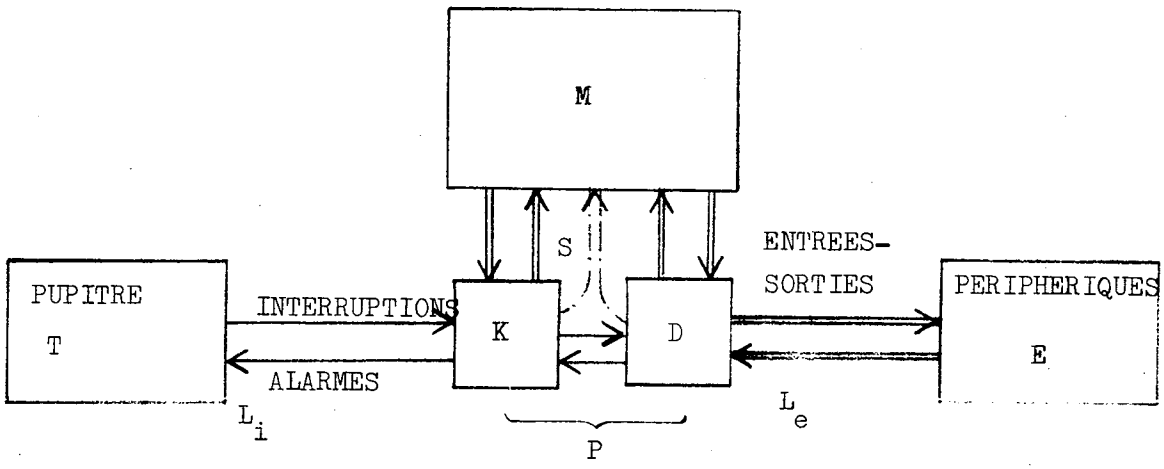


Figure III.7. Schéma synoptique d'un système de traitement de l'information.

Inhérente à cette structure est la notion de centralisation du contrôle géré exclusivement par K ; cependant les diverses unités K,D,M,E... sont chacune des machines séquentielles et toutes sont susceptibles de fonctionner (c'est-à-dire d'effectuer des transitions d'état) simultanément.

Or le formalisme de la machine de TURING permet de dégager quelques fonctions primitives d'un système de calcul à monoprogrammation séquentielle exclusivement :

i) A l'écriture des données initiales sur la mémoire M, puis à la lecture du résultat sur M à l'arrêt final, correspondent les fonctions d'échange sur une liaison L avec l'environnement E, ou encore d'entrées-sorties. Ces questions sont approfondies dans une étude des liens publiée en [DEP.72].

ii) A la notion d'initialisation du traitement correspond celle d'interruption-pupitre pour le chargement initial, ou encore de console-opérateur T avec alarmes de signalisation d'arrêt.

iii) Aux changements d'état de l'unité de commande K correspondent les instructions de séquencement I_K du processeur central P.

iv) A l'écriture des données intermédiaires et finales sur M correspondent les instructions d'assignation I_D de l'unité de traitement des données D.

v) Aux déplacements de la tête de lecture-écriture correspondent les fonctions d'adressage ou de sélection S.

Cependant pour prendre en compte la possibilité de fonctionnements simultanés, c'est-à-dire, la coopération d'algorithmes décentralisant plus ou moins le contrôle du calculateur C, il est nécessaire d'étendre les modèles précédents, selc les fondements posés au paragraphe III.4.2.

L'idée de généraliser la machine de Turing par l'adjonction de plusieurs têtes de lecture-écriture pour une bande de mémoire M puis de plusieurs bandes de mémoire pour un automate de commande [ARB.69] ne permet pas cependant de formaliser ce qu'est effectivement le calcul parallèle. L'illusion vient-elle de ce que l'on parle de "tête" de lecture au lieu de Sélecteur alors que la TETE est en fait le polymate de commande K ? C'est pourquoi nous introduisons comme système réellement polycéphale la notion de "complexe de TURING" N fait de plusieurs automates de commande communiquant par leurs zones communes de bande de mémoire et coopérant en vue de la solution d'un même problème, comme indiqué sur la figure III.8, par exemple, avec :

$$N = (M_P, P_C, M_C, P_E, M_E) \qquad (26)$$

Cette généralisation introduit un autre aspect de la notion de liaison L dans un système de calcul : la synchronisation, ensemble des moyens assurant la collaboration des divers automates du complexe.

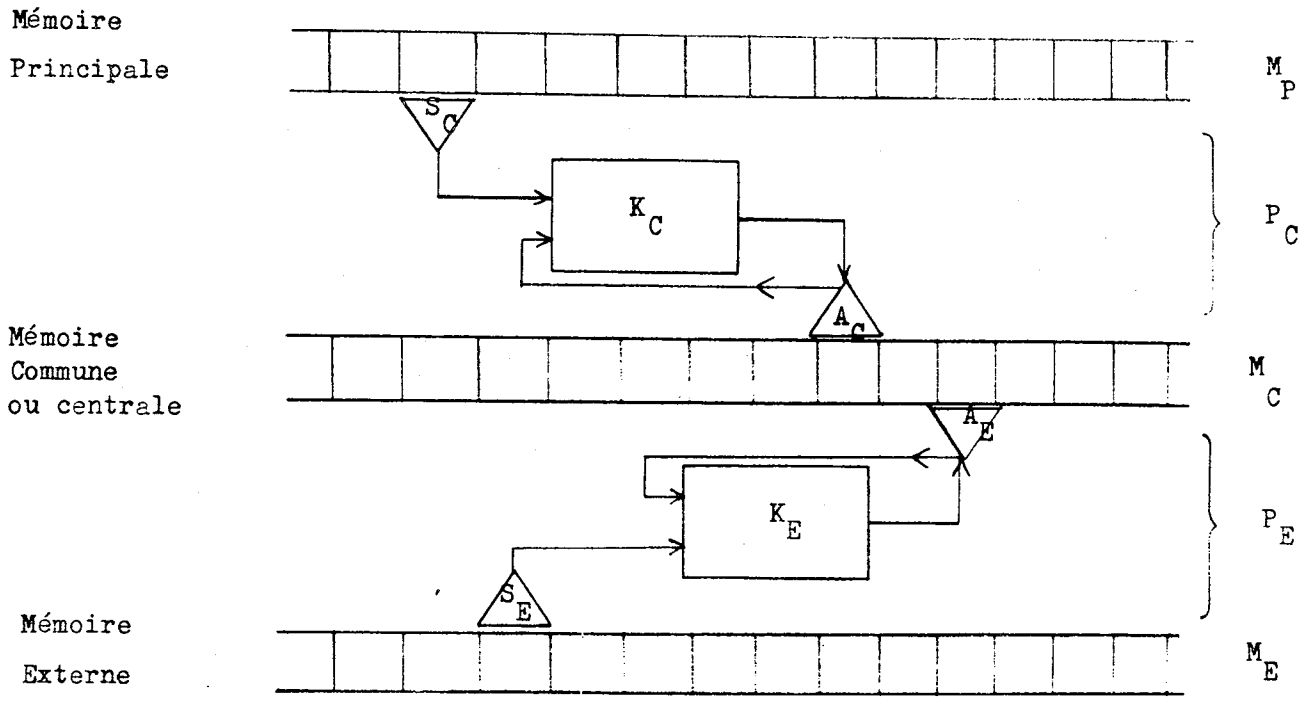


Figure III.8. Complexe de TURING N.

Intuitivement, le complexe de TURING N montre qu'il y a deux niveaux de synchronisation [DEP.72] :

i) le niveau logique d'organisation des programmes en K_C et K_E pour que la coopération logicielle soit cohérente,

ii) le niveau physique de déplacement des organes de lecture-écriture A_C et A_E sur M_C pour que les mécanismes matériels fonctionnement correctement.

CHAPITRE IV

UNE SEMANTIQUE DE PMS FONDEE SUR LES POLYMATES

CHAPITRE IV

UNE SEMANTIQUE DE PMS FONDEE SUR LES POLYMATES

Les tenants de la théorie des Automates, à peine dotés de ses premiers résultats d'ordre mathématique, se sont empressés de la présenter comme la base de la Recherche Informatique. La démarche de cette thèse montre que les concepts antérieurs ne permettaient même pas de poser un certain nombre de questions importantes dans la pratique.

L'architecture des calculateurs ne peut devenir une Science sans l'apport d'une formalisation permettant assez de rigueur pour éviter les déclarations inspirées plus par le sentiment que par la raison. D'autre part, il faut éviter "le complexe du Hoovercraft" amenant à développer un outillage mathématique complexe qui ne parvienne qu'à effleurer la surface du problème.

Une contribution significative est due à BELL et NEWELL [1971] avec le langage PMS qui apporte la rigueur d'une syntaxe élémentaire dans le domaine de la description des structures informatiques. Il s'agit d'une notation que sa mnémonique rend explicite et que nous avons déjà introduite progressivement dans les chapitres précédents (P pour processeur, M pour Mémoire, S pour Sélecteur,...). Ses éléments de base sont appelés "primitives", quoiqu'ils puissent être obtenus par composition à partir des autres. En effet ce langage favorise un affinement gradué de la description en permettant de manière progressive (un effet de "zoom") une description plus ou moins fine de sorte que ses éléments apparaissent comme des primitives relativement à un certain niveau.

Le langage PMS exprime un point de vue caractéristique d'une approche descendante et fonctionnelle, à partir du niveau de la Configuration-système, jusqu'au niveau de transferts de registres (à l'opposé de la méthode ascendante et topologique, à partir du niveau des circuits logiques jusqu'au niveau des paniers en armoires). Son objectif est de permettre la rationalisation des descriptions de systèmes informatiques permettant entre autres la rigueur du raisonnement en Architecture de machines, moyennant la précision de débits, etc. comme en [BEL.72] ou [DEP.73.a], mais aussi le développement de programmes de Conception Assistée.

La machinerie en PMS apparaît comme un réseau de composants interconnectés, qui effectuent chacun quelque opération sur des formats de données spécifiques, mais qui sont connus seulement d'un point de vue externe ("le cahier de charges").

La faiblesse majeure de cette approche est de ne pas fournir -comme le fait pour les machines séquentielles la théorie des automates- d'information sur la dynamique du système sous-jacent qui est en fait un réseau de sous-systèmes interactifs à l'image d'un complexe de TURING.

Ce chapitre IV démontre que le modèle des polymates permet d'ajouter sémantique et dynamique à la syntaxe de PMS et l'effort essentiel porte sur la spécification des interfaces entre primitives (liens implicites) et sur l'analyse des primitives de liens L (explicites). Alors que dans la plupart des structures les interfaces apparaissent seulement comme des connexions et non comme des composants d'un système, ils sont présentés ici comme le lieu privilégié des mécanismes de synchronisation, des procédures de communication et du tamponnement des échanges. Ces problèmes d'ailleurs sont plus en relation avec la théorie de la commande (l'Automatique) qu'avec celle du traitement de l'information (l'Informatique).[DEP.72,73.a].

IV.1. DEFINITION DES PRIMITIVES DE PMS.

1) L (liaisons ou liens) transfère des données d'un emplacement à un autre sans modification. La procédure de transmission a la forme d'un algorithme dont les étapes sont obtenues par entrelacement des actions initialisées alternativement par chacun des deux modules terminaux de la liaison. Cette procédure et les conventions de représentation des données forment l'interface.

2) S (sélecteur, commutateur ou décodeur) change la liaison qui connecte des composants à accès multiplexé afin de réorienter les transferts de données. Il permet l'adressage d'autres primitives.

3) K (commande, contrôle, séquencement) actionne l'exécution des opérations d'un composant dans le système en réponse aux diverses conditions d'entrée et en fonction de son propre état interne.

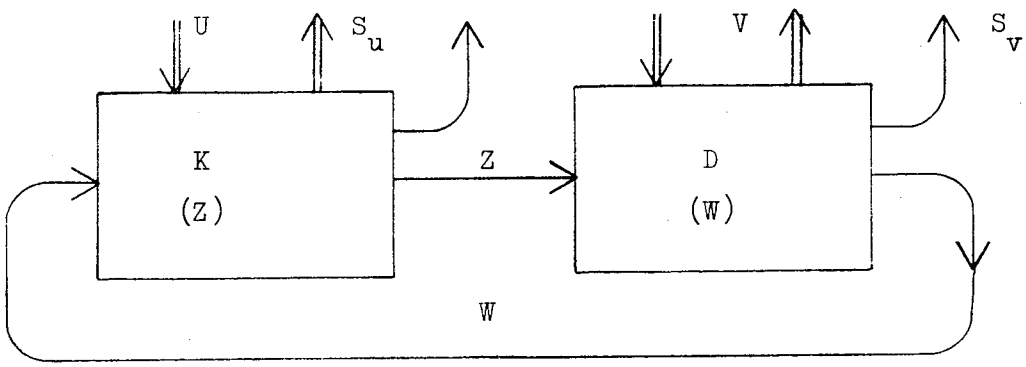
4) D (Opérateurs sur les Données, unité d'exécution) est le composant qui produit une nouvelle information dans une certaine représentation codée à partir d'un ensemble de données d'entrée.

5) M (mémoire) emmagasine les données sans modification au cours du temps. La mission essentielle de cette primitive "passive" est d'offrir une grande capacité d'enregistrement (elle est dite passive en ce sens qu'elle est incapable d'exécuter un algorithme).

6) T (transducteur, transcodeur ou modulateur) change le format des données, à partir d'une représentation non-numérique (analogique, externe au système numérique) ou à partir d'une représentation numérique (codage en bits, octets,...) vers une autre (internes au système numérique), mais il ne modifie pas la quantité d'information associée.

7) P (processeur) est un sous-système consistant en un ensemble d'autres composants avec un K qui détermine interprétativement la prochaine opération à requérir afin de pouvoir progresser de manière autonome. Ainsi P = <K,D> où le symbole <,> représente l'interface vertical entre l'automate de commande ordonnant des actions et l'automate d'exécution signifiant dualement des conditions. De ce point de vue la différence entre D et T en quantité d'information consiste en ce que D ajoute aux données d'entrée (V*) les informations de commande (Z) émises par K. (cf. §.III.3).

Figure IV.1. Le tourbillon d'un processeur <K,D>.



La mission essentielle de cette primitive "active" est d'offrir une grande vitesse d'exécution (elle est dite active en ce sens qu'elle seule est capable d'exécuter un algorithme).

Un calculateur C est un système incluant au moins un processeur P et une mémoire M contenant le programme des opérations destinées à P. La connexion de plusieurs C forme un réseau N ("network").

C peut être vu comme une boîte noire dont l'environnement E intervient en entrée $E_I : \emptyset \rightarrow V^*$ et en sortie $E_O : W^* \rightarrow \emptyset$, où V est l'alphabet fini d'entrée codant toutes les données (instructions et opérandes), où W est l'alphabet intermédiaire incluant Y celui des résultats (sorties de D) et où $\emptyset$ est l'ensemble vide.

L'environnement $E = (E_I , E_O)$ mémorise les données, avec modifications éventuellement, afin d'être une source et un puits pour le flot au travers ("throughput") du système C. Celui-ci apparaît comme un quadruplet où chaque tiret représente un interface de connexion : (27)

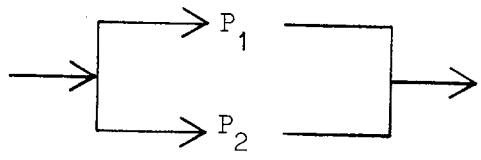
$C ::=$ P-M-L-T (-E) où $P = \langle K_P , D_P \rangle$ et $L = \langle K_L , D_L \rangle$ effectuent des assignations de valeurs, mais D_P représente des opérations transformationnelles comme $A ::= B + \bar{R}$ tandis que D_L représente des opérations translationnelles comme $d_M ::= \delta_T$ où d_M est une variable en M et δ_T est une valeur en T (cf. §.III.3).

Le rôle des liens L dans cette recherche de conception structurée (tant pour les programmes que pour les machines) mérite d'être souligné : parce que D_L est seulement translationnel, l'étude de L est plus simple que celle de P. L'analyse des interfaces et liaisons facilite donc la mise en évidence des principes fondamentaux comme c'est le cas dans l'étude [DEP.72]. De plus chaque fois qu'un L apparaît explicitement dans une équation architecturale en PMS, un interface est spécifié, mais plus souvent un interface doit être défini alors qu'un L n'apparaît pas explicitement comme dans le cas d'une connexion P-M entre un processeur et sa mémoire.

IV.2. INTERCONNEXIONS.

Soit P une primitive qui peut être un Automate ou plus généralement toute boîte noire dotée d'entrées et de sorties et "-" une connexion dont la durée de parcours égale le temps de traversée (ou temps de cycle de P) : il y a sept "schémas" fondamentaux pour la connexion des primitives ; ce sont, du point de vue de PMS sept "formules" mathématiques bidimensionnelles (la durée de parcours est une clause importante dans les cas de rebouclages à cause des risques d'instabilité).

A. En parallèle tel que :



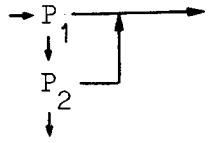
noté $P_1 // P_2$. (28)

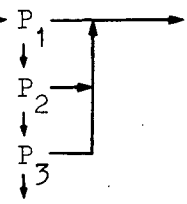
Cette loi de composition est associative et commutative : l'ensemble des primitives forme un semi-groupe abélien par composition. Le montage en parallèle définit une relation d'équivalence (transitive, symétrique et réflexive) : l'ensemble des interfaces induit des classes de primitives parallèles.

B. En série : $\rightarrow P_1 \rightarrow P_2 \rightarrow$ noté $P_1 \rightarrow P_2 \triangleq P_{12}$. Cette loi de composition (29)

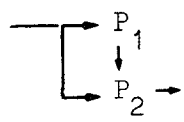
est associative seulement. L'ensemble des primitives forme un monoïde.

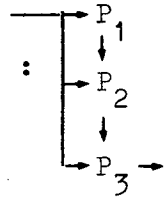
$P_{123} \equiv P_{12} \rightarrow P_3 \equiv P_1 \rightarrow P_{23}$. Le montage en série définit une relation d'ordre.

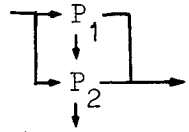
C. En série-pontée :  noté $P_1 \bar{\rightarrow} P_2$. Cette loi de composi- (30)

tion est associative seulement :  . Du point de vue relationnel,

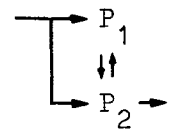
elle est transitive.

D. En cascade :  noté $P_1 \times P_2$. (31)

Cette loi de composition est associative seulement : 

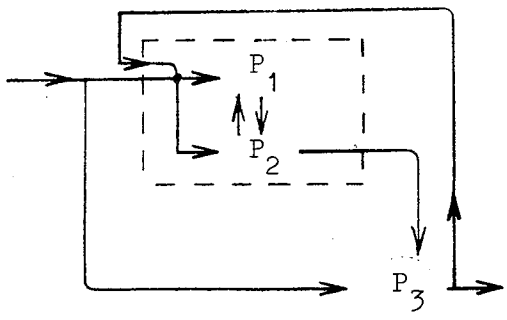
E. En cascade-pontée :  noté $P_1 \bar{\times} P_2$. (32)

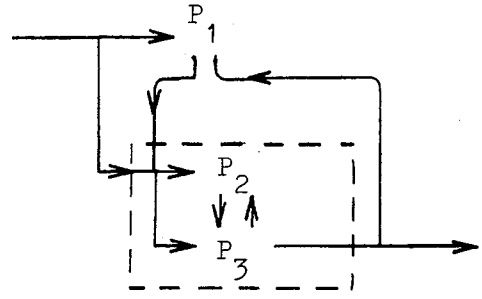
Cette loi est associative seulement. Une application est présentée en [DEP.71.e].

F. En tourbillon asymétrique :  noté $P_1 \overset{*}{-} P_2$. Cette loi (33)

n'est pas associative. En effet soit P_3 un troisième module. Il suffit de comparer les deux formes :

$(P_1 \overset{*}{-} P_2) \overset{*}{-} P_3$ et $P_1 \overset{*}{-} (P2 \overset{*}{-} P3)$. (34)

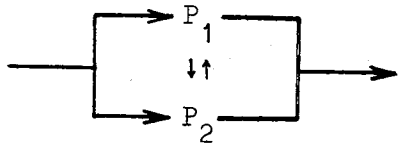




Les deux cas diffèrent par les connexions respectives :

$P_1 \leftarrow P_2$ et $P_2 \leftarrow P_3$ (c'est ici que la clause des temps de traversée est cruciale).

G.En tourbillon:



noté $P_1 * P_2$. (35)

Cette loi est associative et commutative.

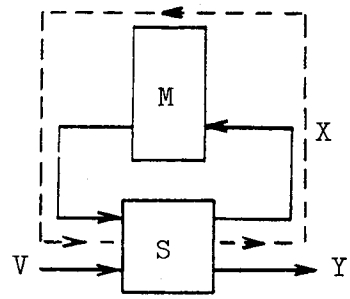
ASSERTION IV.1.

La connexion en tourbillon est une relation d'équivalence si les modules P sont des automates. Dans ce cas la connexion en série est une relatio d'ordre.

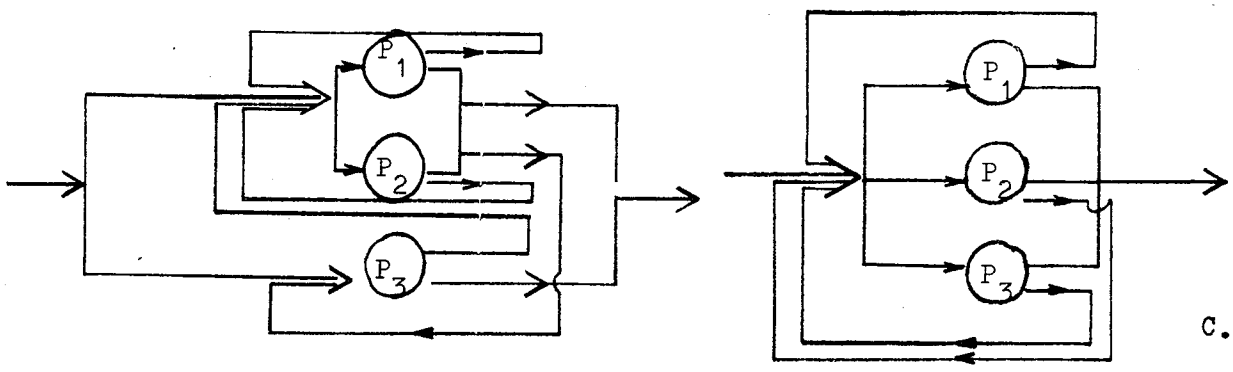
Démonstration :

En effet elle est réflexive puisque tout automate peut être formalisé sous une forme canonique en tourbillon rebouclant ses sorties sur ses entrées. (cf.figure II.1.b).

P::=

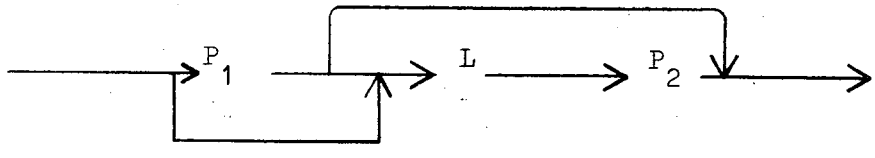


De plus elle est commutative (symétrique) et associative (transitive) comme le montrent les représentations équivalentes suivantes :



C.Q.F.D.

Les interconnexions non parallèles peuvent être encore plus spécialisée si un L est spécifié entre les modules P_1 et P_2. Par exemple :

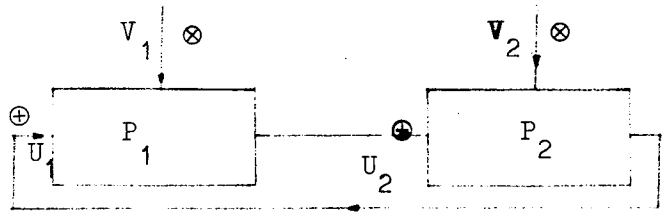


Ici P_1 envoie dans P_2 un signal de synchronisation direct permettant -soit que les transitions de P_1 et P_2 soient entrelacées (temps de cycle et temp morts intercalés),[DEP.71.e],
-soit que la fréquence de transition de l'un des modules soit un multiple de cell de l'autre, faisant de la relation P_1/P_2 dans ce type d'interconnexion une relati maître/esclave. (cf. Propriétés III.5.).

Quand ce ne sont plus seulement des automates mais des polymates qu'i s'agit d'interconnecter -et c'est le cas dès que l'on connecte des bascules JK- la sortie de l'un peut être employée soit comme entrée additive, soit comme entré multiplicative pour l'autre ; on définit alors **quatre types de composition en** tourbillon :

G.1.additive :

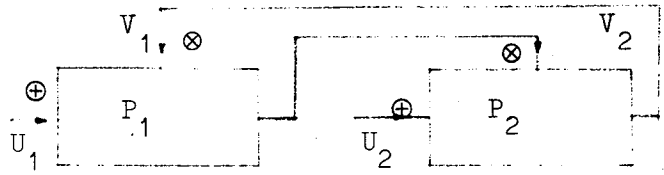
P1 $*_{\oplus}$ P2



(36)

G.2.multiplicative :

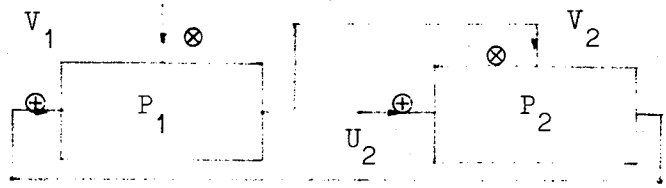
P1 $\overset{*}{\otimes}$ P2



(37)

G.3.mixte :

P1 $\overset{*}{\oplus\ \otimes}$ P2

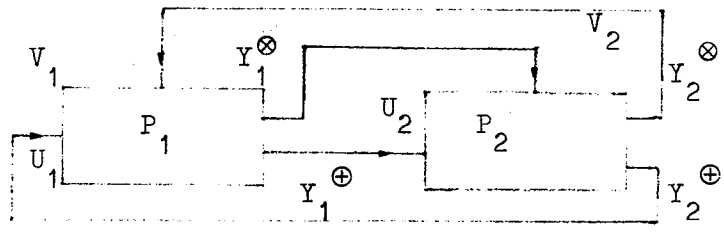


(38)

Comme en général les entrées U sont de type L ("level") et les entrée: V sont de type P ("pulse"), le rôle de la fonction de rythme ρ à l'entrée des polymates est de prendre en compte les problèmes de synchronisation sur interface Il est donc possible de panacher les types précédents en tourbillon si l'on distingue deux sous-ensembles $Y^{\oplus}$ et $Y^{\otimes}$ dans les espaces de sortie :

G.4.double :

P1 ** P2



(39)

Outre les applications de ces types de composition déjà rencontrées aux paragraphes I.3.5 et 6 ou III.3 , nous proposons le résultat suivant directement généralisable à tout registre compteur (avec n ⩾ 2) :

APPLICATION IV.2.

Un registre à décalage à deux bascules monté en compteur binaire diviseur-par-trois est une composition en tourbillon-double de polymates à deux états.

Démonstration .

Il suffit d'analyser le schéma de la figure IV.2.

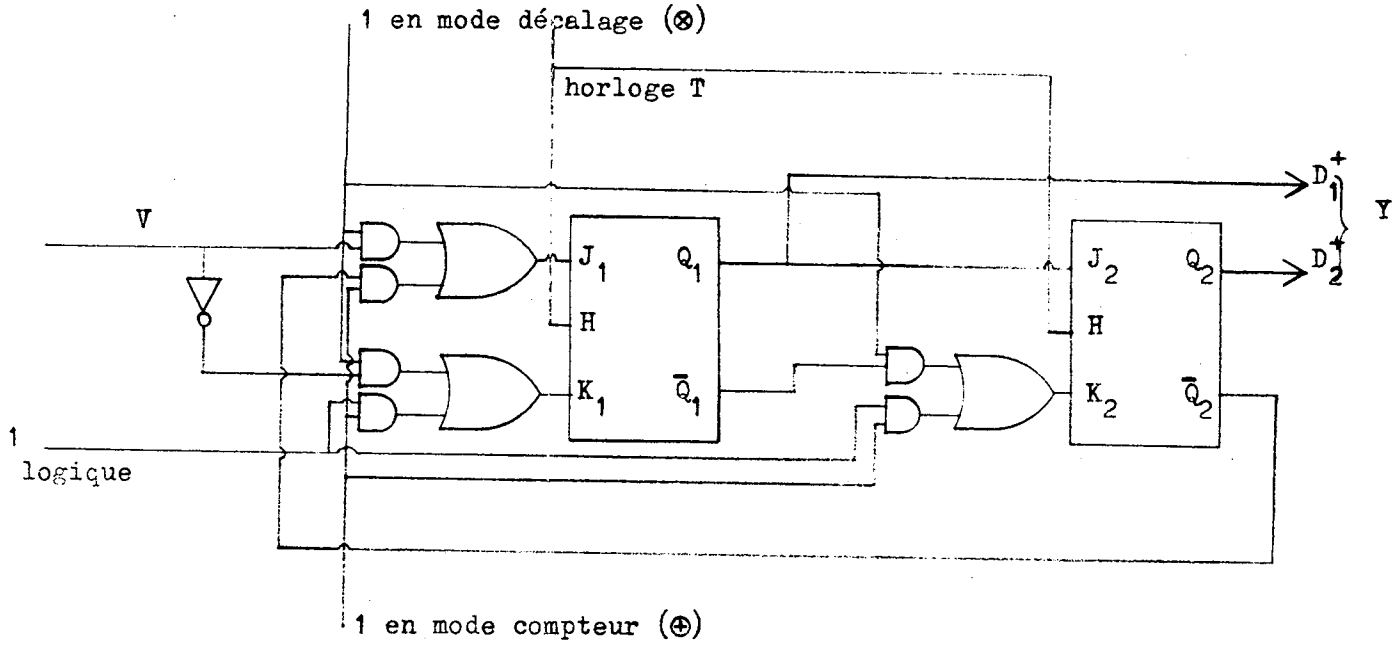


Figure IV.2. Polymate décomposable en tourbillon double.

Au niveau d'interconnexion d'unités plus complexes comme des processeurs P, les liaisons doivent être analysées aussi en fonction de leur mode de synchronisation par extension des propriétés de rythme présentées au paragraphe I.3.2.

Des processeurs interconnectés coopèrent en mode F si leur synchronisation est organisée par réinformation ("handshaking Feedback") ; ils coopèrent en mode H si leur synchronisation est coordonnée par un sélecteur S (une Horloge par exemple).

Le mode H facilite la rapidité de coopération de composants physiquement éloignés, mais il ralentit le traitement à la vitesse du processeur le plus lent. Le mode F facilite la rapidité de coopération de composants rapprochés ainsi que le traitement des erreurs quand les signaux de réinformation sont utilisés pour informer l'émetteur des incidents de transmission sur les liens L correspondants ; mais il introduit des problèmes de courses (critiques éventuellement) dès que des accès simultanés sur des ressources communes, dites sections critiques, sont possibles. Cette notion est donc importante dans l'étude du complexe de TURING introduit au paragraphe III.5.

Exemples.

1) Les entrées-sorties directes d'un calculateur sont effectuées en mode H par scrutation ; en mode F, elles sont gérées par interruptions prioritaires.

2) La composition des automates par produit pseudo-direct [DEP.71.e] requérant l'entrelacement des transitions des composants est en mode F, tandis que la composition par produit direct ou semi-direct [ARB.69], requérant la simultanéité des transitions des composants, est en mode H : l'absence de signal de retour est requise par les théories algébriques des automates, ce qui en explique les limitations pratiques.

3) L'addition (§.II.1.5) en mode F est un forçage et en mode H une permutation sur l'espace d'état d'un polymate.

4) Les télétransmissions sur réseau en mode H sont dites périodiques et font usage de compteurs et horloges de garde locales, tandis qu'en mode F elles sont dites apériodiques et font usage de signaux de retour ("handshaking feedback

IV.3. LES LIENS ET INTERFACES L.

Du point de vue de l'Architecture, L est la primitive essentielle, le ciment d'un système. La Théorie des automates permet d'introduire par ce biai les concepts de méthodes d'accès et de machines séquentielles qui manquent à l'a proche descriptive de PMS.

Définition. Une liaison ou lien est un triplet (V,Y,δ) où δ est une fonction de transfert $\delta : V^* \to Y^*$ (où E^* est l'ensemble des séquences finies obtenues par cor ténation d'éléments d'un ensemble E). Dans les cas simples, où il n'y a pas de transduction séquentielle pour supporter une procédure du dialogue, $\delta : V \to Y$: v introduisant simplement un délai.

La définition logique des procédures de transfert sur un lien est une spécification d'interface séquentiel.

Un lien multipoint est tel que $V = \overset{a}{\underset{i=1}{\times}} V_i$ ou $Y = \overset{b}{\underset{j=1}{\times}} Y_j$. (40

Si $a = 1 = b$, le lien est bipoint entre deux primitives P et P' : $P - L - P'$.

Un lien simplex $\vec{L} = (V,Y,\vec{\delta})$ permet exclusivement des transferts d'une primitive terminale vers l'autre : $P \to L \to P'$.

Un lien demi-duplex permet la transmission dans les deux sens, mais dans un seul à la fois. Il doit être représenté comme un lien à une seule voie physique $P \leftrightarrow L \leftrightarrow P'$.

Un lien vrai-duplex permet la transmission simultanée dans les deux sens Il peut être représenté par une paire de liens $(\vec{L},\overleftarrow{L'})$ de sorte que $P \underset{\leftarrow}{\rightarrow} L \underset{\leftarrow}{\rightarrow} D'$. En fait il peut s'agir d'un dédoublement physique ou d'un multiplexage temporel de la même voie.

Le lecteur intéressé par les applications de ces concepts en trouvera un certain nombre exploitées en [DEP.72 et 73.a]. Il s'agit ici de démontrer l' adéquation du modèle de polymate et des concepts introduits jusqu'ici pour les problèmes architecturaux concrets, pour aboutir au théorème IV.5. du paragraphe IV.6.1.

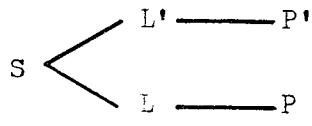
IV.4. LES POLYMATES A TRANSITIONS D'ETATS.

IV.4.1. La primitive S.

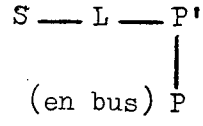
Un sélecteur dans sa forme la plus élémentaire, est un décodeur ou mécanisme d'adressage (U,Y,σ) effectuant un changement de représentation combinatoire de l'espace d'entrée U afin d'émettre un code de sortie primitif ($_1$ parmi n) où $n \leq 2^m$ et $|U| = m$, **ou inversement. Il est noté S(cf. §.I.2.1).**

S peut donc choisir surjectivement un parmi au plus N emplacement physiques pour tout "pointeur" symbolique $u \in U$.

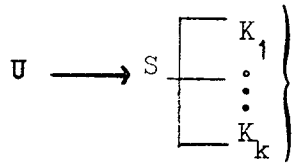
Un lien multipoint requiert une adresse de module et un mécanisme de commutation et de verrouillage de liens bipoints pour les modules connectés, que ce soit en connexion en étoile ou en bus.



(en étoile)



Dès qu'un sélecteur est combiné avec une interconnexion en parallèle d'automates, il en résulte un polymate dont chaque automate traite un microalgorithme spécifique et chaque code d'instruction commande une sélection appropriée.



$$K = \bigoplus_{i=1}^{k} K_i . \qquad (41)$$

IV.4.2. La primitive K.

Comme plusieurs des automates composants peuvent être réalisés de sorte que certaines séquences de micro-instructions ou <u>phases</u> leur soient communes dans le polymate, l'unité de commande K peut être représentée sous une forme canonique où l'espace d'état Z est partitionné en $Z = Z_u \times Z_v$ tel que Z_u mémorise l'instruction sélectionnée et pour l'exécution de laquelle un des automates est en activité. Tout $z_u \in Z_u$ peut être alors considéré comme une adresse de retour pour le micro-programme en cours et être traité comme une donnée dans un registre auxiliaire, tandis que les $z_v \in Z_v$ représentent les instructions d'un micro-sous-programme . C'est le cas dans la machine de Turing polyadique (§. III.2).

Définition formelle.

Soit Z^+ l'ensemble des entiers positifs.

Un polymate à TRANSITIONS D'ETATS est un heptuple $K = (Z,U,W,Y,\pi,\tau,\sigma)$ où Z est l'espace de phases ou de microinstructions, un ensemble linéairement adressable par une application $S_z : Z \to Z^+$ de sorte que l'état est le pointeur vers la microinstruction active, et W est le code de conditions, U est le code d'instructic et Y le code d'ordres, liés par les fonctions d'initialisation $\pi : U \times Z \to Z$, de transition $\tau : W \times Z \to Z$ et de sortie. Cette dernière peut être de deux types : transition-sortie [MEALY] : $\sigma : Z \times W \to Y$ ou état-sortie [MOORE] : $\sigma : Z \to Y$.

L'alphabet d'entrée U peut être interprété (§.II.1.5) :
- soit comme une entité de description hiérarchique dont V assure la fragmentation en une séquence dans Z*, (§.II.2.3), par addition en mode F,
- soit comme une instruction de forçage d'état tel qu'il existe une application injective de U dans Z telle que $\forall u \in U$, $\exists z \in Z$ tel que $u = u_z$ et $\pi(z',u_z) = z$, $\forall z' \in Z$.

Pour pouvoir exprimer dans ce modèle le parallélisme dans la commande il faut l'enrichir des concepts découlant de la notion d'état complexe (§.III.5).

Définitions.

1) L'espace d'état Z est dit complexe s'il peut être partitionné en isolant des états de type FORK (Z_F), JOIN (Z_J) et QUIT (Z_Q), séparant p sous-ensembles d'états partiels tels que rencontrés au paragraphe III.4.2 :
$Z^\dagger = \{Z_F,Z_J,Z_Q\}$ et $Z = Z^\dagger \bigcup_{i=1}^{p} \tilde{Z}_i$ avec $\tilde{Z}_i \cap \tilde{Z}_j = \emptyset$, $\forall i,j \in \{1,\ldots,p\}$, de sorte que la fonction de transition est multivariable : $\tau(Z \setminus Z^\dagger, w) = [\tau_i(\tilde{Z}_i,w)]_{i=1}^{p}$, $\forall w \in W$, (42) où $\setminus$ est le symbole de suppression et où τ_i est la restriction de τ sur $\tilde{Z}_i$.
2) Un sous-ensemble $\tilde{Z}_i$ est actif s'il contient un état successeur d'un état initial.
3) Pour tout $z_F \in Z_F$, $\tau(z_F,w_k) = (\ldots,\tilde{z}_i^k,\ldots)$ avec $w_k \in W$ (43) et $\tilde{z}_i^k \in \tilde{Z}_i$ pour des sous-ensembles $\tilde{Z}_i$ inactifs correspondant à une liste $\mathcal{L}(z_F,w_k)$.
4) Pour tout $z_Q \in Z_Q$, $\tau_i(z,w_j) = z_Q \Rightarrow \tau(z,w_j w) \in Z \setminus \tilde{Z}_i$, c'est-à-dire (44) que z_Q n'a pas de successeur et $\tilde{Z}_i$ n'est plus actif.
5) Pour tout $z_J \in Z_J$, $\exists$ une liste $\mathcal{L}(z_J)$ telle que $\tau_i(z_J,\Lambda)$ n'est défini que si tous les états de la liste sont actifs.

Une telle machine K est non-séquentielle, mais elle est déterministe et synchrone avec parallélisme de degré p.

Quelques définitions.

Selon qu'un système dynamique complexe est présenté en mettant l'emphase sur ses modules composants ou sur ses connexions, on peut utiliser un graphe de fluence ou de précédence associant les modules aux sommets, ou un diagramme de bloc associant les interfaces (données, commandes et résultats) aux sommets. Dans les deux cas les connexions sont orientées dans le sens du temps.

Conformément à ces définitions, il est intéressant d'associer à K un graphe de précédence G(K) où chaque sommet représente un état de Z, et à D un diagramme $\bar{R}(D)$ où chaque sommet représente un élément de mémoire de M. On appelle graphe de UCLA une représentation en graphe [MAR.67] mélangeant données et commande sur les flèches. La technique de MARTIN et ESTRIN de séparation des sommets à logique ET et à logique OU peut maintenant être restreinte aux graphes à précédence temporelle G(K) représentant un polymate à transitions d'états complexes.

THEOREME DES APPELS IMBRIQUES. IV.3.

Un polymate à transition d'états complexes est équivalent à un ensemble de δ polymates séquentiels interconnectés en tourbillon : $\delta \leqslant |Z_F| + 1$.

Démonstration.

Il suffit de remplacer chaque état FORK $z_F^i \subset Z^\dagger$ tel que $\exists\, z_f^i \in \tilde{Z}_i$ et $w_F \in W$ avec $\tau(z_f^i, w_F) = z_F^i$ et $\tau_j(z_F^i, w) = z_g^j \in \tilde{Z}_j$ pour une liste $\mathcal{L}(z_F, w)$ de sous-ensembles cibles $\tilde{Z}_j$ par un état QUIT z_Q^i défini par

$$\tau_i(z_f^i, w_F) = z_Q^i \quad \text{et} \quad \sigma(z_Q^i, w) = \{\sigma(z_F^i, w), \ldots\, u_g^j \ldots\} \tag{45}$$

où chaque u_g^j est une entrée additive des sous-polymates cibles $\tilde{Z}_j$ forçant l'état $z_g^j = \pi(z^i, u_g^i)$. Quant aux états JOIN z_J^i telsque $\exists\ z_j^i \in \tilde{Z}_i$ et $w_J \in W$ avec $z \in Z \setminus \tilde{Z}_i \Rightarrow \tau((z, z_j^i), w_J) = z_J^i$, ils sont remplacés par des états QUIT à sorties additives

$$\sigma(z_j^i, w_J) = u_j^i \text{ et } \sigma(z, w_J) = u \tag{46}$$

où chaque u_j^i est une entrée du sous-polymate cible $\tilde{Z}_i$ tel que $\tau(z_J^i, \Lambda) = z_k^i \in \tilde{Z}_i$ est remplacé par $\pi(\Sigma_{\mathcal{L}(z_J^i)} u_j^i) = z_k^i$. Les états QUIT z_Q^i sont inchangés et correspondent à l'état stable zéro au sens des systèmes linéaires. (47)

L'équivalence obtenue est la suivante : pour toute séquence d'entrée dans $(U \times W)^*$ pour le polymate complexe, la séquence W^* et les séquences d'états actifs dans $Z \setminus Z^\dagger$ sont préservées.

C.Q.F.D.

IV.4.3. Application du concept d'état complexe.

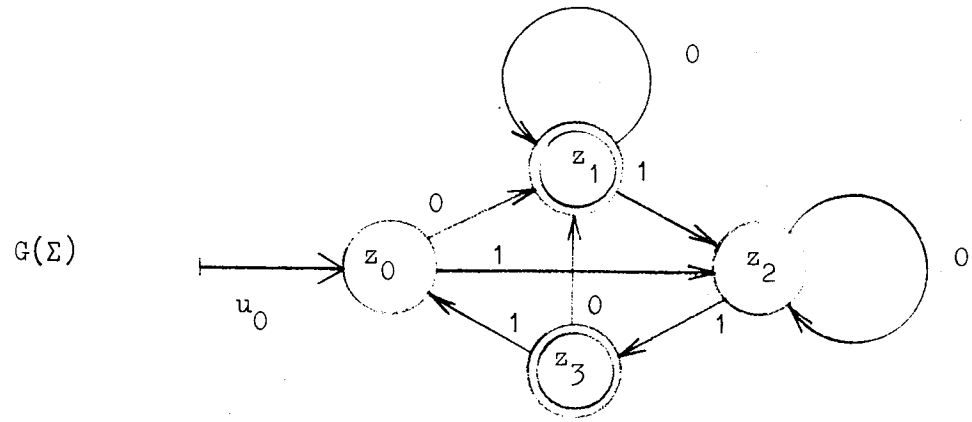
Chacun des polymates séquentiels obtenus ainsi pouvant être considé pour une initialisation donnée par u ∈ U, comme un automate, nous présentons un possibilité d'extension au cas complexe d'un exemple connu d'automate accepteur binaire (§.I.3.5 : figure I.16).

Figure IV.3. Extension complexe de l'exemple de [WEI.68].

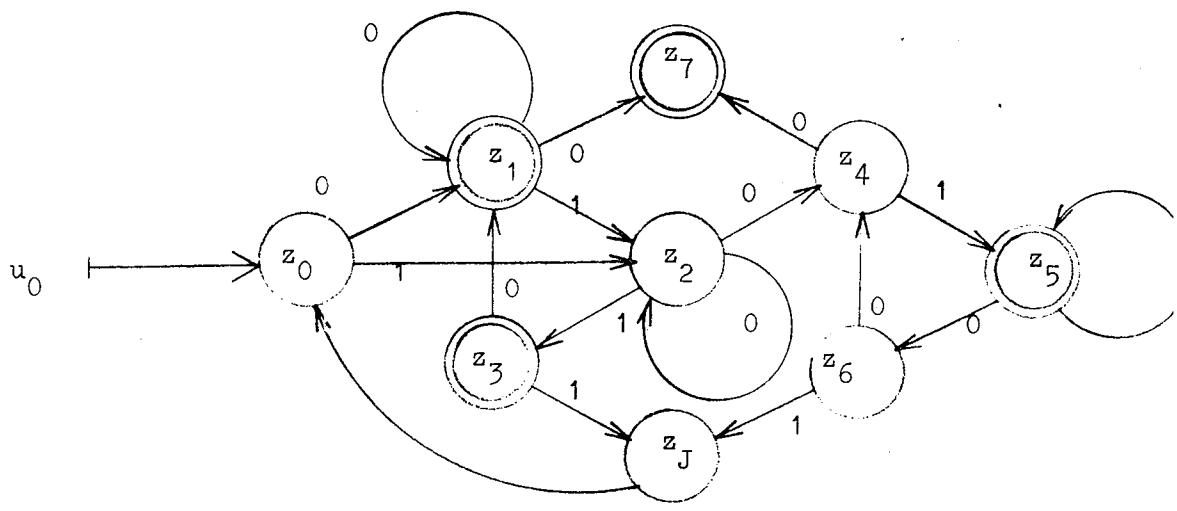
a. Graphe de précédence G(Σ) séquentiel.

Les sommets doublement cerclés correspondent aux états d'acceptation dont la sortie est $y_1 = 1$. Pour les autres, c'est $y_0 = 0$.

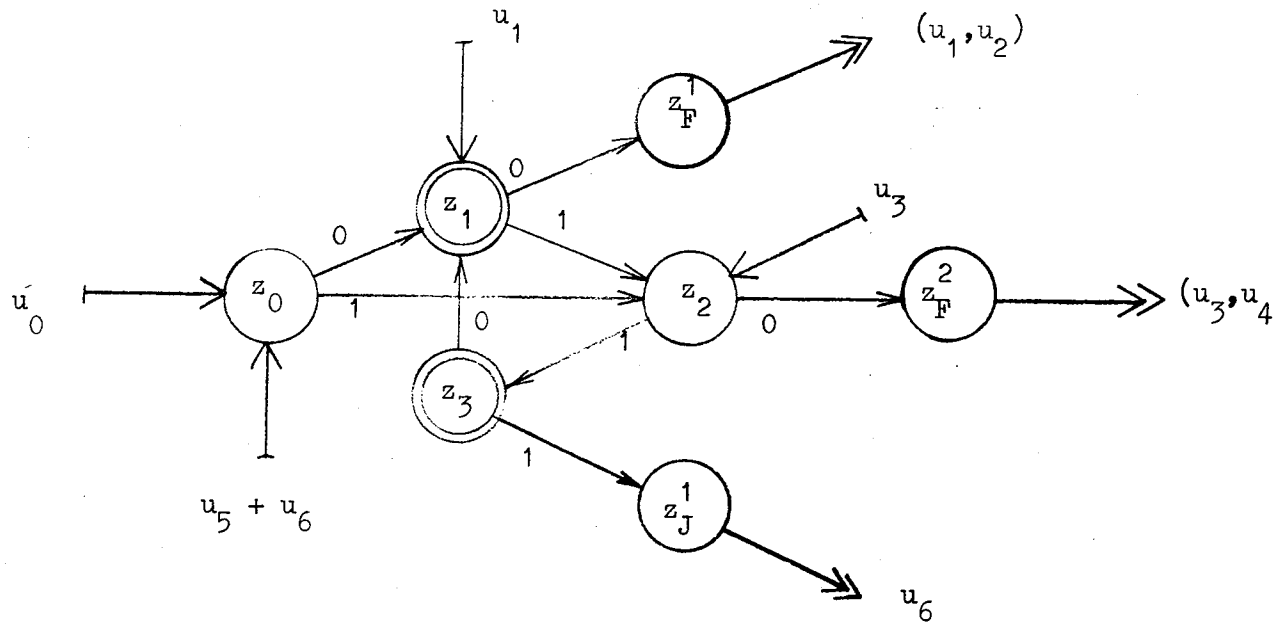
Soit $Z = \{z_0, z_1, z_2, z_3\}$ l'espace d'état et $V = \{0,1\}$ l'alphabet d'entrée.



b. Cas complexe : graphe de précédence G(K).

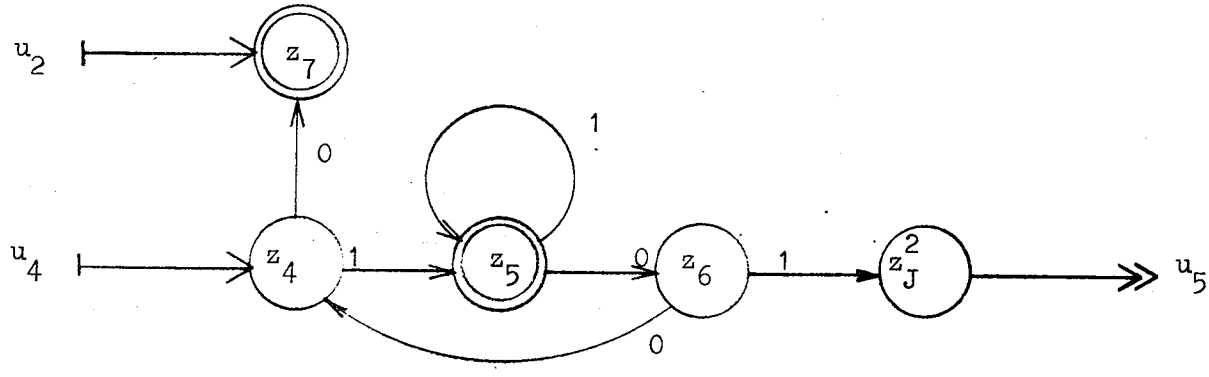


c. Cas complexe décomposé : graphe de précédence $G_1(K)$.



Le symbole ⇉ indique à la fois l'émission d'un signal y de sortie et le retour à un état de référence pour les entrées additives (par exemple ici z_0).

d. Cas complexe décomposé : graphe de précédence $G_2(K)$.



Le théorème des appels imbriqués montre ainsi comment le modèle de polymate permet de définir, par panachage [§. IV .2] des sorties et entrées additives et multiplicatives, un nouveau mode de "composition" de machines séquentielles plus souple que ceux précédemment utilisés [HAR.64, ARB.69]. Ici $|Z_F| = 2$ mais le nombre de sous-polymates de la décomposition est $\delta = 2 < |Z_F| + 1$ (48) parce que les entrées additives qui en émanent agissent sur les mêmes sous-ensembles cibles.

COROLLAIRE DE L'ASSIGNATION PRIMITIVE DES ETATS.IV.4.

Tout polymate à n états peut être décomposé en une cascade double de n polymates à deux états (généralisation de l'Application IV.2).

Démonstration :

Comme indiqué sur la figure IV.4., il suffit d'éclater chaque état z_i du polymate original en un triplet d'états FORK, JOIN et QUIT :

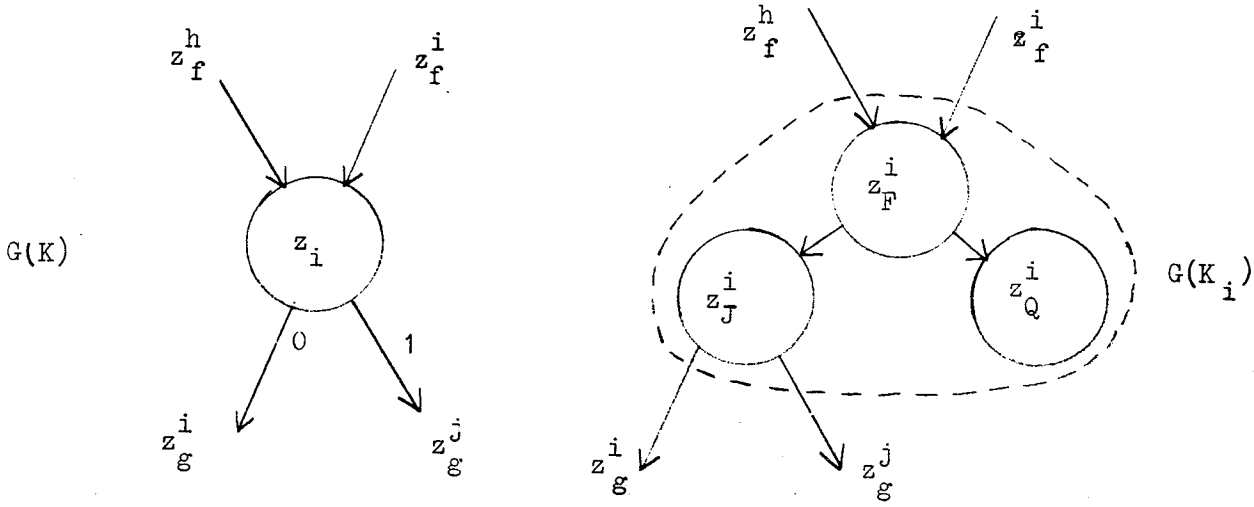


Figure IV.4. Eclatement des états.

puis d'appliquer le théorème des appels imbriqués et d'éliminer les états QUIT n'émettant pas de sortie utile.

Le schéma résultant montre l'interdépendance entre décomposition e assignation des états, car ce corollaire signifie simplement que toute machine séquentielle peut être implémentée par interconnexion en tourbillon double de J automates (ce qui est de pratique courante): **l'assignation est primitive car chaque JK mémorise que l'état qu'il représente est actif ou non.**

C.Q.F.D.

Remarques.

1)Il est remarquable que cette relation entre décomposition et assignation ait échappé à [WEI.68] et que cette différence sémantique ait bloqué le développement de leurs travaux. Nous allons l'exploiter au chapitre V .

2)Chacun des n sous-polymates joue ici le rôle d'un module indépendant avec une logique d'entrée fonction des prédécesseurs de z_i dans G(K) ; cett logique permet aussi de remplacer le type JK des bascules par le type TRIGGER ou le type DATA exclusivement en traitant les entrées additives et multiplicatives manière identique.

3)Noter qu'un état z_k^i (comme ici z_0) peut être réalisé en traitant chaque u comme une entrée multiplicative, ce qui ne préserve pas l'échelle temporelle des transitions et permet d'isoler ici une primitive REQUEST (u_5 et u_6) :

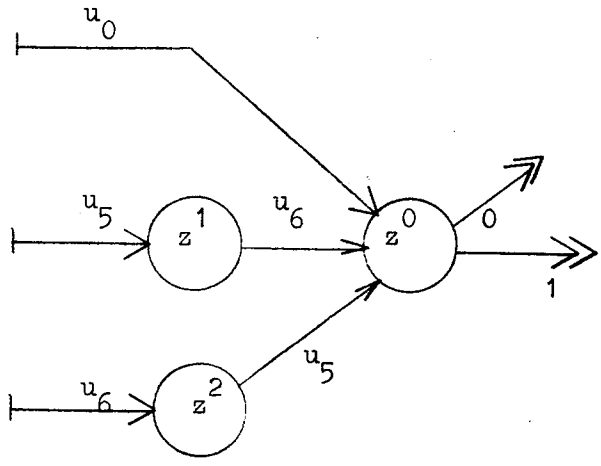
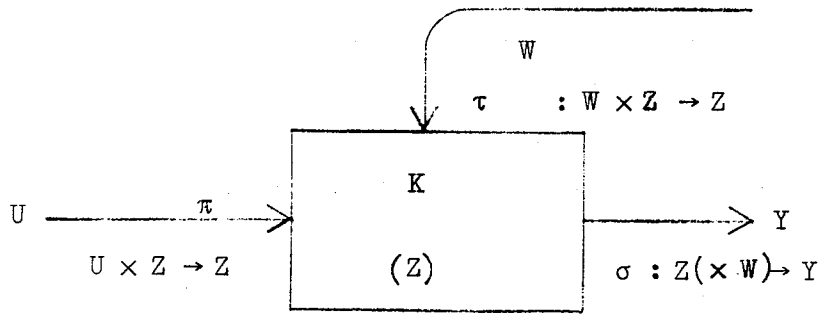
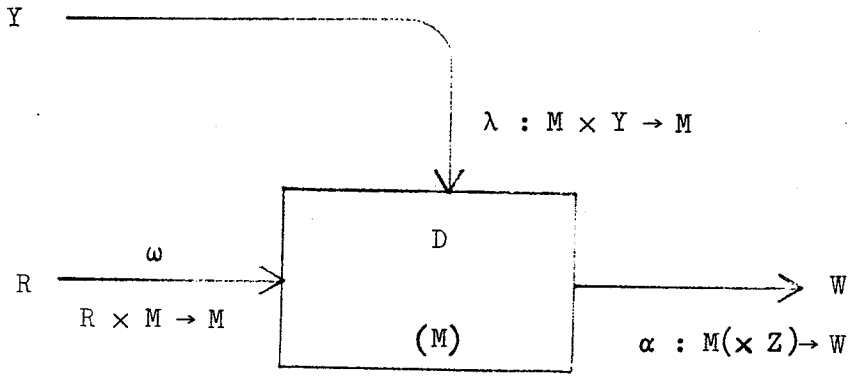


Figure IV.5. Traitement multiplicatif des entrées additives (primitive REQUEST). Les états z^1 et z^2 servent à assurer la commutativité de u_5 et u_6.



a. Polymate K à transition d'états.



b. Polymate D à transfert de registres.

Figure IV.6. Schéma synoptique des polymates duaux.

IV.5. LES POLYMATES A TRANSFERT DE REGISTRES.

IV.5.1. La primitive M.

Un emplacement de mémoire M_i (cellule ou registre) est un automate d'ord 1 formalisable comme un quintuple (V,X,Y,ψ,η) où la fonction de transition ψ et la fonction de sortie η opèrent à l'exclusion l'une de l'autre : les informations entr $(V \to X)$ ou bien sortent $(X \to Y)$ de M_i; ceci est exprimé par l'élément neutre d'entrée $\Lambda \in V$ qui agit sur l'espace d'état X comme l'identité. De plus les trois espaces sont inclus : $V \supseteq X \supseteq Y$ de sorte qu'une partie de l'information soit protégée (inaccessible).

La fonction de transition ψ emmagasine simplement la dernière donnée d'entrée en détruisant l'état présent :

$$\psi : V \times X \to X \quad \text{avec } \psi : \Lambda \times x \mapsto x$$
$$\text{et} \qquad v \times x \mapsto v, \forall v \in (V \setminus \Lambda) \cap X.$$

tandis que la fonction de sortie η fournit l'état ou <u>valeur du contenu</u> :

$$\eta : V \times X \to Y \text{ avec } \eta : \Lambda \times x \mapsto y = x$$
$$\text{et} \qquad v \times x \mapsto y = \Lambda, \forall v \in (V \setminus \Lambda) \cap Y.$$

Une extension formelle de ce modèle est le concept d'emplacement de mémoire-tampon représenté par un automate d'ordre k tel que l'espace d'état mémorise les séquences d'entrée de longueur inférieure ou égale à k :

$$M_i^{(k)} : \psi(v_0 v_1 \ldots v_k) = \psi(v v_1 \ldots v_k), \forall v \in V, \tag{49}$$
$$\text{et} \quad \eta(x, v_1 \ldots v_k v_{k+1}) = x\, v_1, \quad \forall x \in X^k.$$

Une unité de mémoire est l'extension-polymate d'une <u>cellule</u> R_i fournie par un sélecteur d'accès $S : U \to i$ pointant sur un ensemble $\{M_i\}$: $M \triangleq (S, \{M_i\})$. Il est remarquable que ce modèle est une extension de la RESET-machine de ZEIGER [HAR.66] que l'on pourrait baptiser machine-à-<u>forçage</u>. C'est aussi une extension de l'automate DATA du paragraphe II.1.5. Par contre $M_i^{(k)}$ est une machine à décalage d position du contenant tandis que l'automate TRIGGER est à permutation de valeur du contenu.

---

*) <u>Rappel</u> : l'ordre d'un automate est le plus petit indice k tel que, quel que soit l'état initial x(0), pour toute séquence d'entrée v(0)...v(t), l'état final est bi défini si et seulement si $t \geqslant k$, c'est-à-dire $\tau(.,v(0)\ldots v(t)) = x(t)$.

IV.5.2. La primitive D.

Une unité d'exécution D est obtenue par composition en tourbillon d'un opérateur combinatoire (arithmétique et logique) et d'un registre de mémorisation AQ (accumulateur). On emploie le terme de cellule pour R_i si l'adresse u_i est connue explicitement et de registre dans le cas implicite.

Définitions formelles.

1) Un flux [DEP.74.a] est la séquence de valeurs des données initiales et intermédiaires dont le traitement par des opérateurs fournit une valeur(finale) appelée résultat. Un flux est décrit en commençant par son résultat.

2) Un polymate à TRANSFERTS DE REGISTRES est un heptuple $D = (M,R,Y,W,\omega,\lambda,\alpha)$ où

M est l'ensemble des cellules de mémoire, un ensemble doté de contenus ou états par une application $\eta : M \to Z$ où Z est l'ensemble des entiers.

R est l'ensemble des registres susceptibles de contenir un résultat,

W est l'ensemble des valeurs assignables comme arguments aux variables de

Y comme pour K est l'alphabet des actions de transfert commandables,

ω est la fonction d'accès finalisant les transferts en initialisant un flux : $\omega : R(\times M) \to M$ de sorte que la cellule résultant soit active, c'est-à-dire prédécesseur d'un registre résultat dont le contenu est en cours de transfert,

λ est la fonction de liaison spécifiant la source du transfert : $\lambda : M \times Y \to M$ de sorte qu'un ordre y tient lieu d'adresse du prédécesseur de M_i,

et α est la fonction de valuation fournissant le contenu des cellules actives de deux manières : - rémanente (mode niveau : L) : $\alpha : M \to W$ (induisant une partition sur l'espace d'état de chaque registre)

- transitoire (mode impulsionnel : P) : $\alpha : M \times Y \to W$.

Il faut remarquer que pour un registre terminal donné la séquence des transferts est générée par λ en remontant l'écoulement du temps jusqu'à accès aux arguments. La notion de parallèlisme des données est inhérente à celle de registre dans la mesure où l'atome de représentation est le mot et non seulement le bit, mais de plus, certaines liaisons λ correspondent à des opérateurs non monadiques (dual de Z_F) et d'autres permettent une sortance non unitaire (dual de Z_J) ou encore fournissent des valeurs immédiates (dual de Z_Q). Ce fait peut être exprimé avec le concept d'onde (dual d'état complexe) correspondant à un ensemble de registres à transferts équidistants d'un terminal comme suit :

$\lambda : M \times Y \to (M_1,\ldots,M_q) \subset 2^M$ (possibiliste) $\Rightarrow$ l'onde de rang ℓ de $r \in R$ est

$\lambda^\ell(\omega(r),y^\ell)$, $\forall y^\ell \in Y^\ell$. (50)

La figure IV.6. met en évidence la relation étroite (appelée dualité) entre K et D. Il est important d'y remarquer l'inversion du sens des fonctions dans un cas et dans l'autre par rapport au temps. Cette dualité revient comme un leitmotiv au long de cette dissertation mais le lecteur curieux d'en analyser sa nature est invité à se reporter aux références [DEP.71.b et 74.a]. Elle a l'avantage de procurer un modèle nouveau et adapté pour les opérateurs D en réponse à l'interrogation du paragraphe I.3.6 et de nous préparer à l'analyse des résultats du chapitre suivant V.

IV.6. LES TRANSDUCTEURS ET TRADUCTEURS.

IV.6.1. La primitive T.

Un transducteur T est un polymate dont la fonction essentielle est de transformer des séquences. Dans le domaine des signaux analogiques, c'est l'entrée additive de type U qui porte l'information à traiter avec échantillonna Dans le domaine des signaux binaires, c'est l'entrée multiplicative de type V qu porte l'information à transformer.

Le transducteur préserve les longueurs si les séquences d'entrée et de sortie qu'il associe sont de même longueur ; il fragmente les longueurs si à chaque entrée correspond une séquence de longueur non nulle en sortie ; il réduit les longueurs si chaque sortie correspond à une séquence de longueur non nulle en entrée (§.II.2.3).

Arrivés à ce point nous avons démontré implicitement le théorème suiva

THEOREME DE LA SEMANTIQUE DE PMS. IV.5.

Toute primitive de PMS est réalisable par une interconnexion de polymates. C.Q.F.

IV.6.2. PMS et polymates logiciels.

Mais comme le montre l'équivalence des machines de WANG et de TURING, il n'y a pas de raison pour limiter cette approche au domaine matériel ; dans le domaine-logiciel, un système contemporain typique utilise le schéma de programmation suivant :

(K) une section de programme (PSECT) contenant le code exécutable avec adresses translatables mais sans informations qui puissent être altérées en cours d'exécution (procédure pure).

(D) une section de donnée (DSECT) ou plusieurs,

(L) une section de connexion (CSECT) contenant toutes les liaisons et informatio: de contrôle nécessaires à l'édition des liens et au chargement (voire à l'exécution en cas de chargement dynamique),
(S) un module de sélection pour gérer le multiplexage de K sur plusieurs D en cas de réentrance et la synchronisation de plusieurs procédures P = <K,D> en cas de multiprogrammation.

Quelquefois S et K sont regroupés en une CSECT et L et D en une DSECT. Pour les procédures enfin il est naturel de considérer T comme un traducteur.

IV.6.3. Polymates et graphes.

Suite aux définitions du paragraphe IV.4.2. et par opposition au graphe de précédence qui indique la "succession temporelle" des états, un diagramme de dépendance R décrit la "précession spatiale" des registres actifs ; pour lui faire correspondre l'inversion du cours du temps, il suffit de définir le diagramme R(D) par inversion du sens des flèches sur $\bar{R}(D)$.

Le modèle de système de traitement de l'information que nous avons dérivé de PMS au moyen des polymates doit maintenant être comparé au graphe dit de UCLA, [MAR.67]. L'innovation ici consiste en la séparation des instructions -sommets du graphe G(K) pour K et des registres-sommets du réseau ou diagramme $\bar{R}(D)$ pour D, et par suite en la séparation des conditions de branchement-flèches du graphe de K, et des microordres de transfert-flèches du diagramme de D. Cette constatation vaut aussi bien par comparaison au modèle "peu naturel" des réseaux de PETRI.

Il est remarquable que le polymate à transition d'états complexes, comme celui à transfert d'ondes sont une généralisation de réseaux PERT dans lesquels tous les noeuds ont une logique à parallélisme en entrée (JOIN) et en sortie (FORK). Mais G(K) associé à K correspond à l'ordonnancement des opérations, tandis que R(D) associé à D correspond à l'allocation des cellules de données.

On peut alors s'interroger sur le sens d'une assignation de probabilité aux flèches des "graphes duaux" de K et D par comparaison au cas de UCLA. Le gain en généralité apparaît identique à celui obtenu par l'introduction de la notion de variable dans un calcul ; en effet les probabilités de transition dans un algorithme de calcul peuvent maintenant être exprimées indépendamment de données spécifiques, et dualement les probabilités de transfert dans une structure de données peuvent l'être indépendamment de traitements spécifiques.

Remarque. $\bar{R}(D)$ est une généralisation du modèle de KARP. MILLER et WINOGRAD [67] de calcul à récurrence uniforme où l'absence de branchements rend invariantes les relations de récurrence.

Cet ensemble de considérations montre bien que l'exploitation du modèle de polymate à transferts de registres, tant cherché depuis le paragraphe I.3.6 ne devait pas s'arrêter ici. Le lecteur peut trouver en référence [DEP.74.a] la poursuite de l'analyse et des applications de ce concept.

CHAPITRE V

IMPLEMENTATION D'UN POLYMATE DE COMMANDE PAR DUALITE

CHAPITRE V

IMPLEMENTATION D'UN POLYMATE DE COMMANDE PAR DUALITE

Nous avons vu à la fin du chapitre I , que les machines DK qui sont par nature des opérateurs D peuvent être exploitées occasionnellement comme des automates de commande ; inversement, ce chapitre est l'eccasion d'introduire une technique de réalisation de machines de commande (appelées KD) selon une méthode apparentée aux opérateurs D.

V.1. COASSIGNATION D'UN POLYMATE EN FORME ETAT-SORTIE.

L'exemple de Σ pris dans [WEI.68] est poursuivi pour illustrer la démarche (cf. §.IV.4.).

Soit Z en codage primitif (1 parmi n) avec $n = |Z| = 4$ et $p = |Y| = 2$.

$z_0 = (1\ 0\ 0\ 0)$, $z_1 = (0\ 1\ 0\ 0)$, $z_2 = (0\ 0\ 1\ 0)$, $z_3 = (0\ 0\ 0\ 1)$.

Soit Y en codage 1 parmi p :

$y_0 = (1\ 0)$, $y_1 = (0\ 1)$.

Formellement, le modèle des polymates permet d'ajouter l'état de repos $z_\infty = (0\ 0\ 0\ 0)$ auquel correspond la sortie $y_\infty = (0\ 0)$.

Les transitions sont représentées par des multiplications matricielles :

$$\tau(z,v) = z' \quad \Leftrightarrow \quad z\ B_v = z' \tag{51}$$

et les initialisations requièrent une matrice de commande C_Z telle que :

$C_z = [0\ 0\ 0\ 0]$ pour $z \neq z_\infty$ et $C_{z_\infty} = [1\ 0\ 0\ 0]$ avec $u_0 = 0$ et $u_1 = 1$, ce qui permet l'initialisation en z_0 à partir de z_∞ avec u_1 selon l'équation de dynamique des polymates : $z(t+1) = z(t).\ B_{v(t)} + C_{z(t)}\ u(t)$. (52)

La représentation matricielle des entrées dans V^1 pour $V = \{0,1\}$ est alo

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad v_0 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad v_1 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \text{ et } A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

est la matrice d'agrégation en sortie.

La théorie des semigroupes sur les automates [ARB.69] s'est concentrée sur l'étude des classes de congruence $S \triangleq V^*/\equiv_\tau$. (53)

Dans l'exemple, il apparaît que $v_0^2 \equiv_\tau v_0, v_1v_0v_1 \equiv_\tau v_1v_0, v_1^4 \equiv_\tau v_1$ et $v_1^3 v_0 \equiv_\tau v_0$. D'où $S = \{\Lambda, v_0, v_1, v_0v_1, v_1v_0, v_1^2, v_0v_1v_0, v_0v_1^2, v_1^2v_0, v_1^3, v_0v_1^3, v_0v_1^2v_0\}$.

Définitions.

L'équivalence $\equiv_D$ induit l'espace des coétats $W \triangleq V^*/\equiv_D$, qui est l'ensemble des classes d'équivalence sortie-entrée telles que $w \equiv_D w' \Leftrightarrow$ $\sigma(\tau(z_i,w)) = \sigma(\tau(z_i,w'))$, $\forall z_i \in Z$. (54)

Autrement dit, l'ensemble des registres est le dual des classes d'équivalence de Nérode [NER.58] : $M_S \triangleq \{M(y) | \forall y \in Y\}$. (55)

Pour y_1, dans l'exemple précédent Σ, il suffit de considérer les coétats w_j générés (à partir de la droite) par multiplication matricielle à partir de $y_1^t = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ pour chacun des éléments de S :

$$\Lambda A\, y_1^t = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} \triangleq w_0 \; ; \; v_0 A\, y_1^t = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} \triangleq w_1 \; ; \; v_1 A\, y_1^t = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \triangleq w_2 = v_0v_1 A\, y_1^t \; ;$$

$$v_1v_0 A\, y_1^t = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \triangleq w_3 \; ; \; v_1^2 A\, y_1^t = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \triangleq w_4 \; ; \; v_0v_1v_0 A\, y_1^t = w_2 \; ; \; v_0v_1^2 A\, y_1^t = w_1 \; ;$$

$$v_1^2v_0 A\, y_1^t = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} \triangleq w_5 \; ; \; v_1^3 A\, y_1^t = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \triangleq w_6 \; ; \; v_0v_1^3 A\, y_1^t = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \triangleq \emptyset \; ; \; v_0v_1^2v_0 A\, y_1^t = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \triangleq Z.$$

Par souci de clarté, nous allons d'abord suivre l'obtention du résultat de cette approche avant de formuler et démontrer le théorème dans le cas général.

Le polymate à transfert de registre D(K) requiert donc au plus $|W| = d$ registres formant sa mémoire M_S. Si l'on considère à part les monostables ($\emptyset$ et Z), dans le cas de Σ il faut $d = 8$ registres ayant seulement à mémoriser le fait qu'ils sont ou non actifs : leur contenu est une valeur logique (1 ou 0) et chacun peut donc être implémenté à l'aide d'une seule bascule à forçage (DATA). Dans un tel diagramme, un rectangle représente un registre M_k dont le contenu w_k est le codage de l'un des coétats actifs d'un polymate à transfert de registres.

Définition : Soit $D = (M_S, \tau^{-1}(y_1), V, W, \omega, \lambda, \alpha)$ tel que défini au §.IV.5.

La fonction de liaison se déduit de τ par translation gauche comme suit :

$$v_i \tilde{v}_j A y_1^t = w_i \text{ et } \tilde{v}_j A y_1^t = w_j \text{ avec } w_i = \alpha(M_i) \text{ et } w_j = \alpha(M_j)$$

$$\Rightarrow \lambda : M \times V \to M : (M_i, v_i) \mapsto M_j. \qquad (56)$$

Comme dans [KAL.68], à un état non-observable correspond un coétat (une assignation) non-atteignable : ceci est une des expressions les plus rigoureuses du concept de dualité (§.II.1.3).

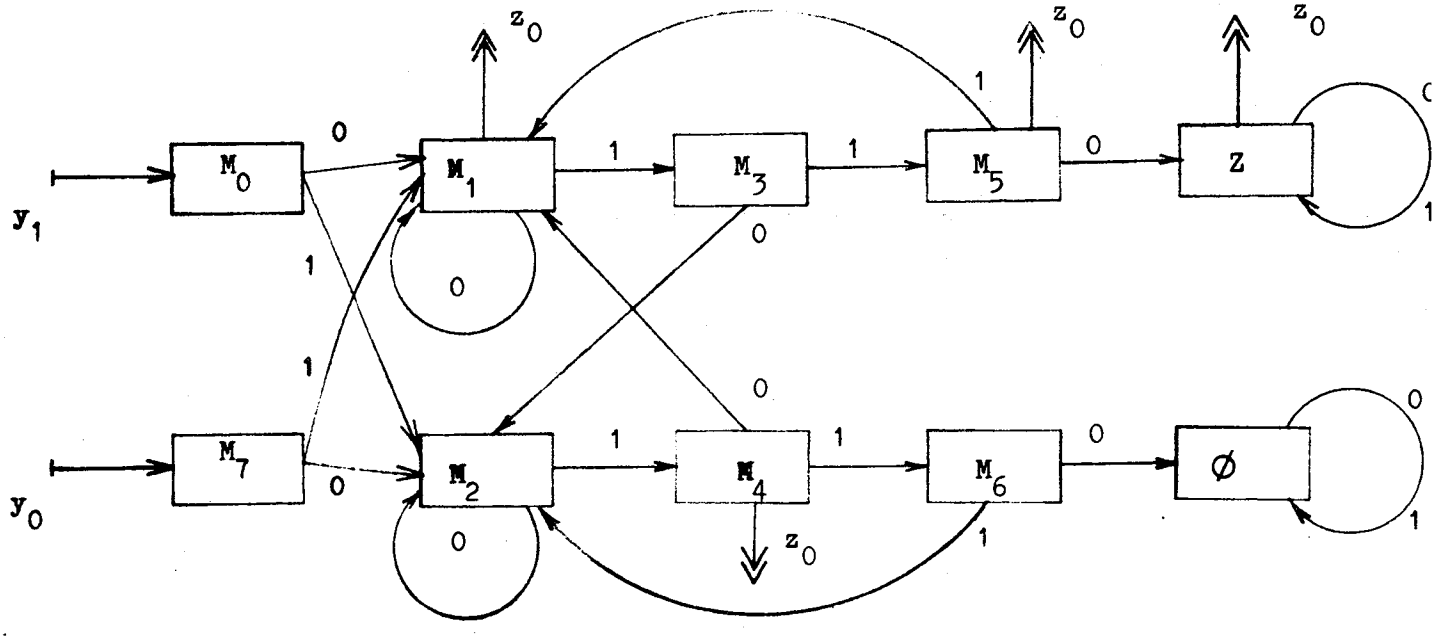


Figure V.1. Dualisation du graphe de précédence en diagramme de dépendance D(Σ).

L'assignation des états est définie par le codage des coétats w_j :

"K est dans l'état $z_i \Leftrightarrow$ les registres M_j de D(K) tels que $w_{ij} = 1$, sont actif

Si deux états z_i et z'_i ont la même assignation $D(z_i) = D(z'_i)$, c'est qu'ils so équivalents : $z_i \equiv z'_i$. Nous l'appelons <u>coassignation</u>.

Soit D_S l'assignation des états en forme état-sortie (§.II.2.2.) :

Coassignation D_S	w_0	w_1	w_2	w_3	w_4	w_5	w_6	w_7	$\emptyset$	Z	Sortie
z_0	0	1	0	0	1	1	0	1	0	1	y_0
z_1	1	1	0	0	1	1	0	0	0	1	y_1
z_2	0	0	1	1	0	1	0	1	0	1	y_0
z_3	1	1	0	1	0	0	1	0	0	1	y_1

$$n \leq d_S \leq |S| \leq n^n \quad \text{et} \quad d_S \leq 2^n \text{ où } n = |Z|. \tag{57}$$

Si l'on prend maintenant $\omega : \sigma^{-1}(y_1) \rightarrow w_0$ et $\alpha : \{M_1, M_4, M_5\} \rightarrow D(z_0)$,

on vérifie que la dualité dans la représentation matricielle du §.II.1 permet de retrouver la "décomposition" de [WEI.68]. D'une part, ceci établit la relatio entre deux définitions de la dualité [RAB.64 et DEP.68], d'autre part ceci amène à interpréter cette procédure non pas comme une décomposition mais comme un procédé d'implémentation, ainsi que nous l'avions exposé au paragraphe IV.4.3.

A ce titre-ci il est intéressant de comparer les figures I.16 et V.1.

THEOREME D'ASSIGNATION DUALE. V.1.

Soit un polymate $K = (U,V,Z,Y,\pi,\tau,\sigma)$ avec $|V| = q$ et $|Z| = n$; il existe une assignation des états D telle que chaque chiffre binaire w du codage d'un état soit fonction (par τ) de exactement q chiffres binaires du codage de l'état précédent.

1) Chaque code D(z) requiert au plus (n-1) chiffres binaires non nuls dans au moins n bascules et au plus d avec $d \leqslant 2^n - 2$.

2) Le polymate K peut être implémenté en une forme canonique R(K) isomorphe au polymate à transfert de registre D(K) à l'aide d'un seul type de module logique ACTIVATE à entrance fixe q et en nombre d compris entre n et 2^n.

3) L'état de repos du polymate est l'état stable (au sens des systèmes linéaires) ayant pour assignation le vecteur de d zéros.

Démonstration.

Il reste à démontrer que D(K) est isomorphe à une implémentation R(K) en forme canonique (figure II.1) en réalisant la fonction de liaison λ de D(K) à l'aide d'un réseau combinatoire $\bar{\tau}$ à entrance fixe. Pour cela nous définissons un boitier technologique recevant toutes les entrées multiplicatives et une entrée additive, toutes les entrées étant à codage primitif, comme sur la figure V.2. (Ce module est l'extension pour polymates des modules de [WEI.68]. La bascule M_i est choisie ici du type DATA mais pourrait être un JK dont u_i serait l'entrée additive modulo 2).

Il suffit de remarquer que l'assignation des états duale D_S se caractérise par le fait que la fonction combinatoire $\bar{\tau}$ est définie par

$$\lambda(M_j,v_h) = M_i \bullet \bar{\tau}(w_i,v_h) = w_j, \qquad (58)$$

de sorte que chaque bascule M_j est, pour une entrée donnée v_h, le successeur d'une seule bascule M_i.

Pour forcer l'état initial, $C_{z_\infty} = [0\ 1\ 0\ 0\ 1\ 1\ 0\ 1]$ permet à u_1, de mettre le coétat de z_0 dans le registre d'état. C.Q.F.D.

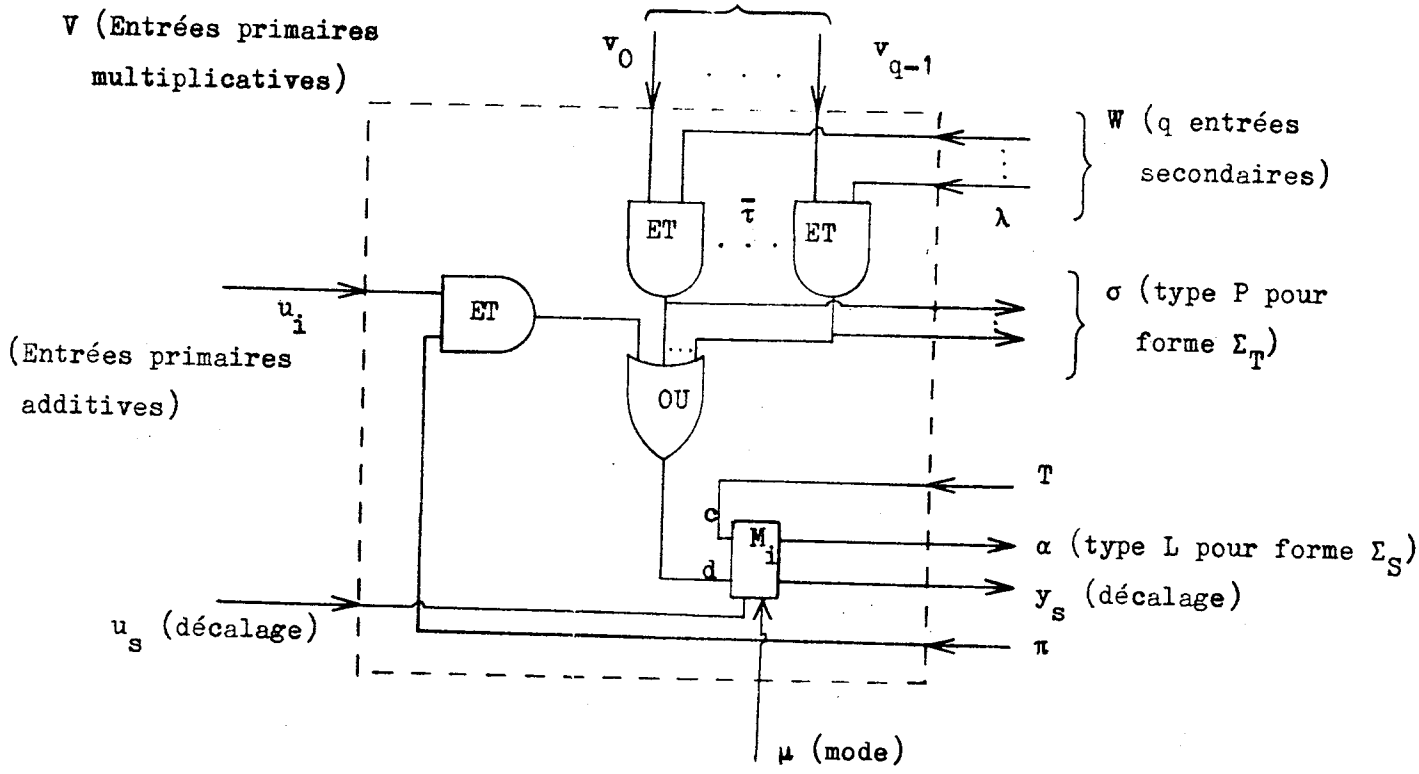


Figure V.2. Module ACTIVATE.

Les bascules peuvent être montées en registre à décalage en ajoutant une fonction de mode μ qui inhibe π, τ et σ (fonctionnement parallèle) de sorte que chaque période d'horloge T entraîne : $M_i \leftarrow u_s$, $y_s \leftarrow M_i$.

V.2. MISE EN FORMES CANONIQUES DE COASSIGNATION.

V.2.1. Forme Etat-Sortie KDS.

La fonction de liaison λ peut-être déduite de la table de coassignation et mise en forme de table de liaison. Elle permet de mettre en évidence une forme canonique d'interconnexion appelée KD avec table des coétats $\bar{\tau}$.

τ	v_0	v_1
z_0	z_1	z_2
z_1	z_1	z_2
z_2	z_2	z_3
z_3	z_1	z_0

Table des états de Σ

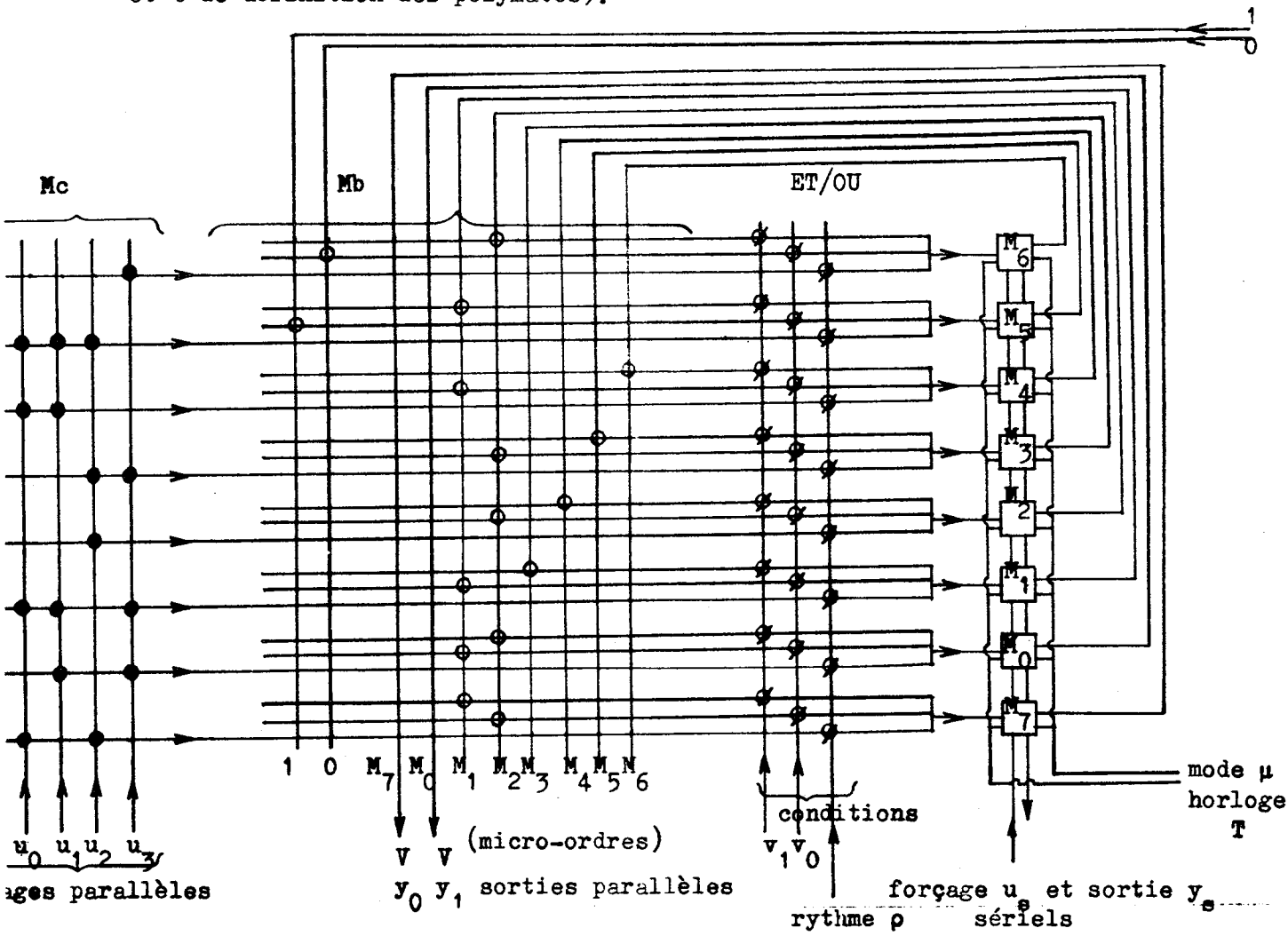
λ	v_0	v_1
M_0	M_1	M_2
M_1	M_1	M_3
M_2	M_2	M_4
M_3	M_2	M_5
M_4	M_1	M_6
M_5	1	M_1
M_6	0	M_2
M_7	M_2	M_1

Table de liaison de Σ

τ	v_0	v_1
w_0	$\emptyset$	$\emptyset$
w_1	w_0, w_1, w_4	w_5, w_7
w_2	w_2, w_3, w_7	w_0, w_6
w_3	$\emptyset$	w_1
w_4	$\emptyset$	w_2
w_5	$\emptyset$	w_3
w_6	$\emptyset$	w_4
w_7	$\emptyset$	$\emptyset$
0	w_5	$\emptyset$
1	w_6	$\emptyset$

Table des coétats de Σ

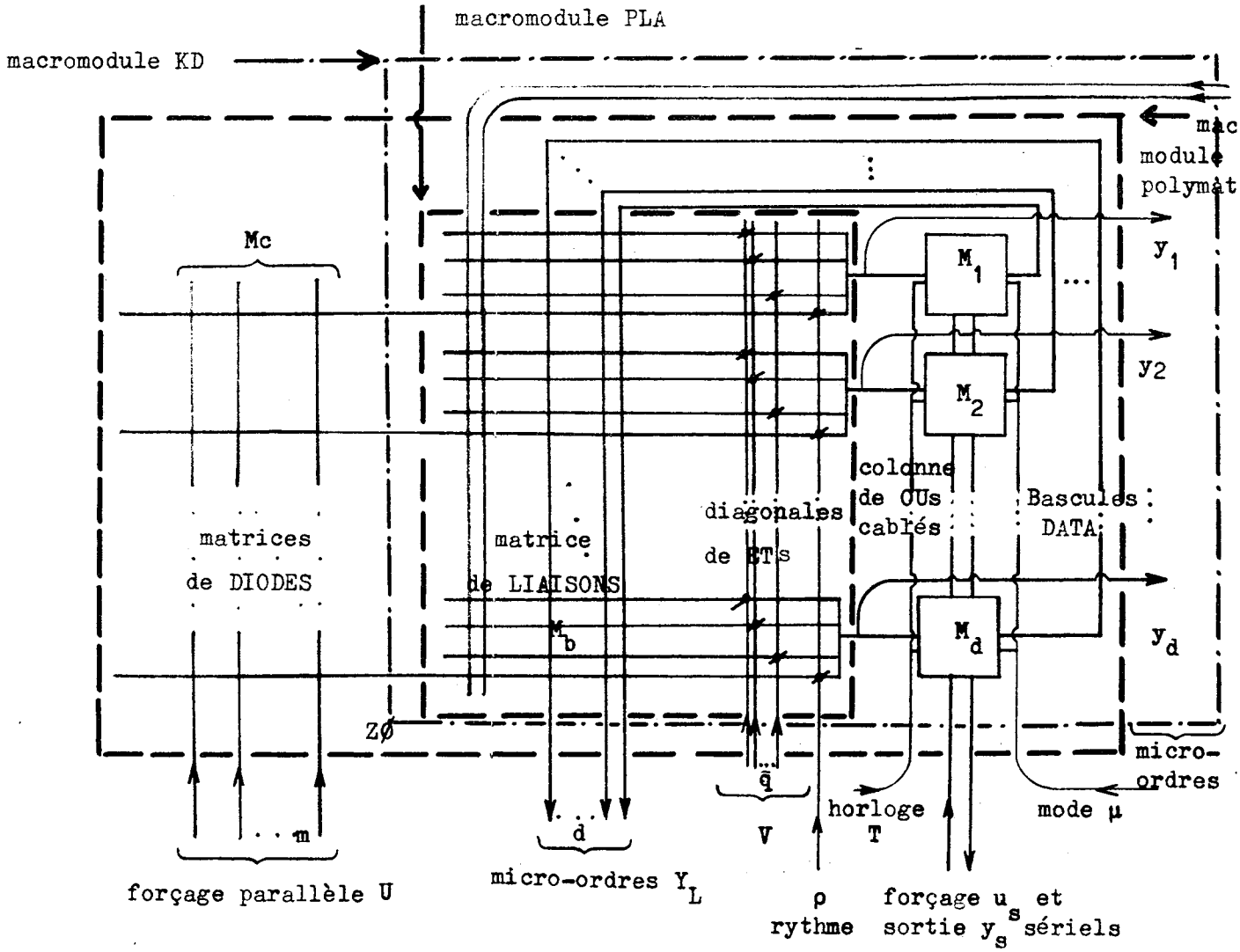
Figure V.3. Mise en forme canonique de coassignation R(Σ) état-sortie. (Les mémoires mortes de connexion Ma,Mb et Mc correspondent aux matrices A,B et C de définition des polymates).



Mc est polynomiale en ligne et requiert donc une diode à chaque point programmable (⊕). Mb est monomiale en ligne (table de liaison λ) et ne nécessite pas de diode aux points programmables. (⊕ signifie la fonction ET).

Remarquer la simplicité de ce schéma par rapport à une réalisation microprogrammée. Aucune sélection d'adresse n'est effectuée en entrée de la mémoire de dynamique Mb et plusieurs lignes sont simultanément actives. Les connexions dans Mb correspondent exactement aux éléments de la table de liaison λ et ceux de Mc à la table d'assignation D_S.

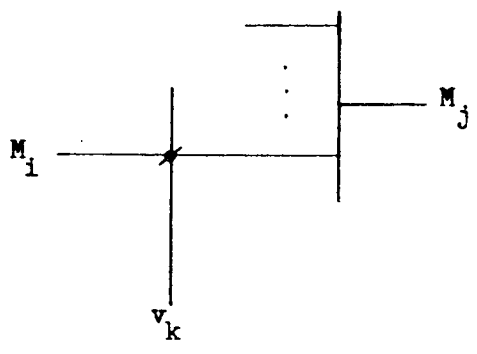
Figure V.4. Coassignation en forme canonique état-sortie KDS.

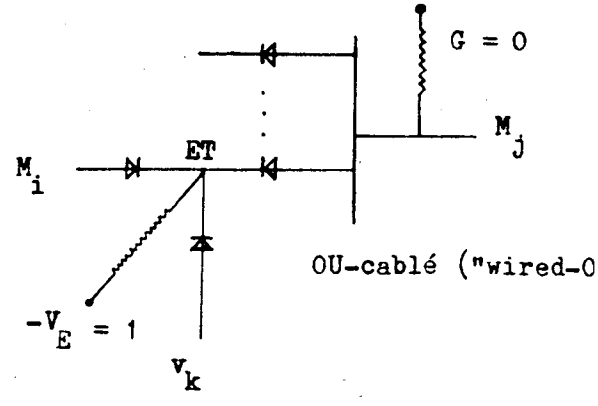


Les sorties Y peuvent être connectées, soit sur les colonnes de Mb ("LEVEL" pris en sortie Y_L des bascules) pour une réalisation de forme S, soit sur les lignes de Mb ("PULSE" pris en entrée Y_p des bascules) pour une réalisation de forme T;

Définitions : 1) Un module KD avant programmation de Mb et Mc est appelé macro-module. [THUR.71], un PLA est un macromodule spécial("Programmable Logic Array").

2) Symbolisme dans le cas d'une alimentation négative :





V.2.2. Forme Transition-Sortie KDT.

Le théorème V.1. (d'assignation duale) peut être converti de la forme état-sortie (Σ_S en II.2.2.) à la forme transition-sortie (Σ_T) au moyen de la représentation en opérande croisé (cf.§.II.1.4.iv et théorème II.8.) où les matrices de dynamique représentent les probabilités conditionnelles $p(z_j, y_k | z_i, v_k)$. (59

Les matrices de transition doivent être décomposées selon que la sortie est l'un ou l'autre des y_i pendant le branchement sur condition v_j, de sorte que $v_j = \Sigma_{i=1}^{p} (y_i | v_j)$. Il faut alors considérer les coétats $e_k = \tilde{v}e_{ij}$ (60)

générés par multiplication matricielle à partir de $e_{ij} = (y_i | v_j)\, e_0$ où (61)

$e_0^t = (1\ 1 \ldots 1)$.

Pour l'exemple de [WEI.68] il vient :

$$(y_0 \mid v_0) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad (y_1 | v_0) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \Rightarrow v_0 = (y_0|v_0)+(y_1|v_0$$

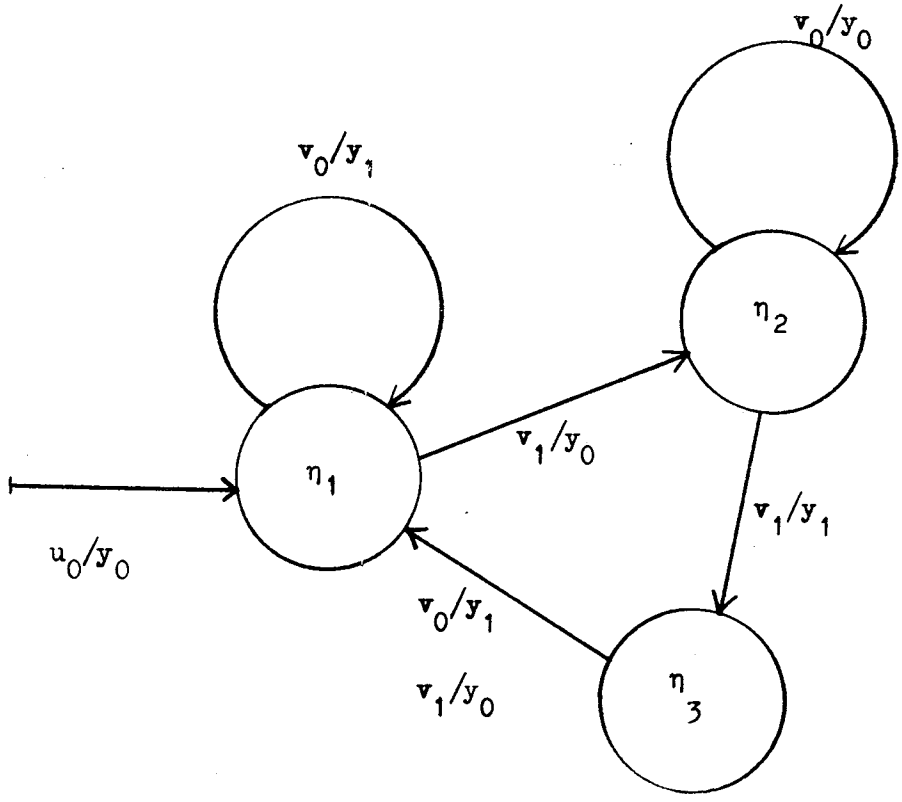
$$(y_0|v_1) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \quad (y_1|v_1) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \Rightarrow v_1 = (y_0|v_1)+(y_1|v_1$$

Les classes d'équivalence sortie-entrée sont :

$$\mathbb{M}_T \triangleq V^*(Y \mid V) \ /\equiv_D \qquad (62)$$

en forme transition-sortie, ce qui revient à remplacer dans le théorème précédent les coétats $w_i \triangleq A\, y_i^t$ par les coétats $e_{jk} \triangleq p(y_k | z_i, v_j)$.

Figure V.5. Graphe de $G(\Sigma)$ en forme Transition-sortie.



Il est remarquable que z_0 et z_1 reçoivent la même coassignation dans D_T puisqu'ils sont équivalents (à η_1) en forme T.

Ces classes d'équivalence M_T — générées par $e_0 = (1\ 1\ 1\ 1)^t$ ($\triangleq$ e défini en II.1.2.b), forment algébriquement un module linéaire dont $|M_T| \triangleq d_T$ vecteurs polynomiaux sur $\{0,1\}$ sont atteignables :

Assignation D_T	e_0	e_1	e_2	e_3	e_4	e_5
z_0	1	0	1	1	0	0
z_1	1	0	1	1	0	0
z_2	1	1	0	0	0	0
z_3	1	0	1	0	1	0

d'où $R(\Sigma)$ en forme T avec $|M_T| = 4$, alors que $|M_S| = 8$.

Par exemple:

$v_1 e_2 = (y_0|v_1)e_2 + (y_1|v_1)e_2$

$= e_4 + e_1$.

Définition.

Deux polymates sont similaires s'ils admettent la même fonction de comportement (§.II.1.4.).

Soient n_S et n_T les nombres d'états en formes minimales similaires Σ_S et Σ_T respectivement: $n_T \leq n_S \leq n_T \times p$ où $p = |Y|$. (63)

Il est inutile de procéder à la procédure de conversion minimisant Σ_T avant d'effectuer l'assignation duale D_T, car deux états équivalents en forme transition-sortie (comme z_0 et z_1 ici) reçoivent automatiquement la même assignation.

La forme transition-sortie KDT donnée par l'assignation D_T offre le double avantage d'être plus compacte ($n_T \leq n_S$ et $d_T \leq d_S < d_T \times p$) et plus rapide (logique de sortie P et non L : cf.§.I.3.4.). D'ailleurs la structure très régulière de la forme canonique de coassignation, avantageuse pour une technologie LSI, réduit aux aléas des composants la variance sur le temps de cycle des polymates R(K), ce qui permet de les employer avec une période réduite à son minimum.

Elle permet aussi de ne pas implémenter une bascule sans successeur, c'est-à-dire pour une sortie y_i dont le coétat e_{ij} est tel qu'il n'existe pas de condition v telle que $e_{ij} = ve_k$ pour un autre coétat e_k.

V.3. RELATIONS ENTRE DECOMPOSITION, ASSIGNATION ET PARALLELISME.

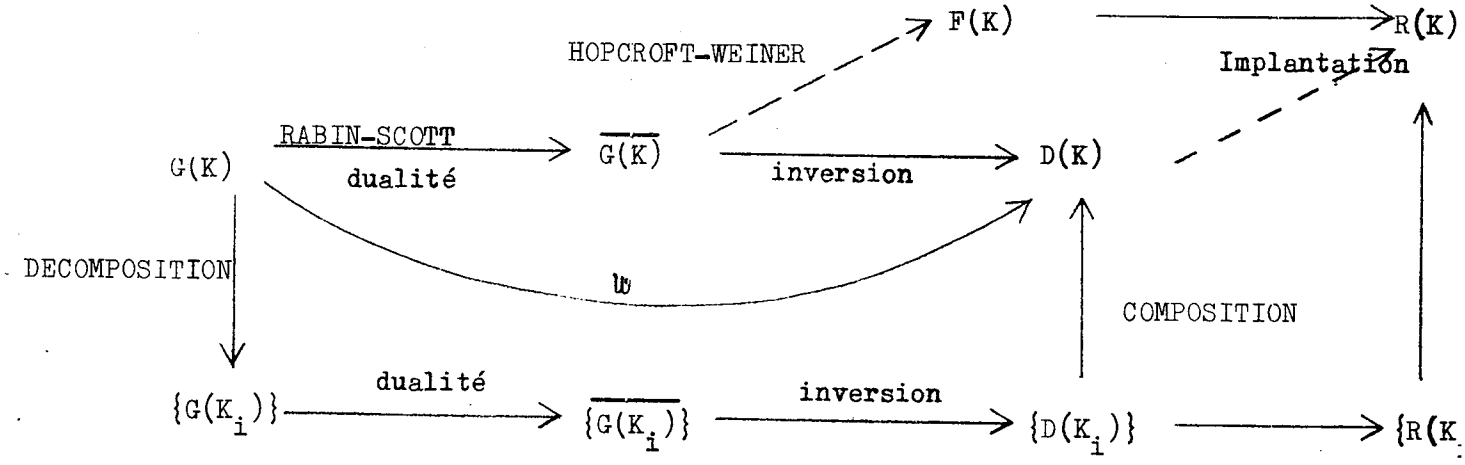


Figure V.6. Diagramme commutatif des transformations de dualité et de composit

Le théorème V.2. et le théorème IV.3 (§. IV.4.2) fournissent deux techniques pour résoudre les difficultés inhérentes au séquencement collatéral (c'est-à-dire parallèle) comme dans l'exemple de la figure IV.3 : d'abord le cas des automates possibilistes est ramené à celui des polymates K à transitions d'états complexes, puis le cas de ceux-ci est ramené par décomposition au cas des automates K_i déterministes.

Ainsi, une partie du graphe de dépendance D(Σ) étendant au cas complexe, le graphe de G(Σ) et correspondant au graphe de précédence de la figure IV. (§.IV.4.3) se présente comme suit :

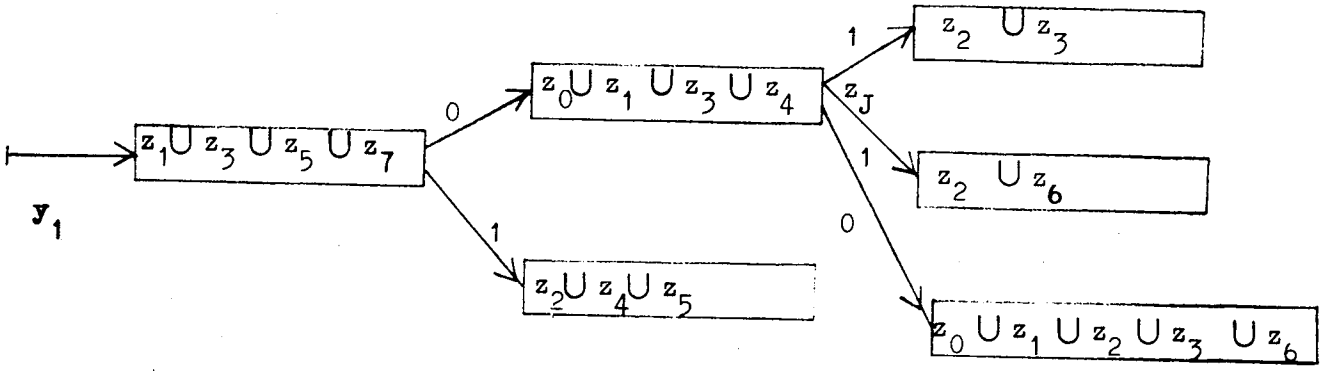


Figure V.7. Graphe de dépendance partiel D(Σ) de l'extension complexe de G(Σ).

PROPRIETES. V.2.

1) Un diagramme de dépendance comme D(Σ) est déterministe (λ est une fonction monovalente) parce que Σ est séquentiel. L'adjonction de z_J pour transformer (Figure IV.3) Σ en K, rend D(K) possibiliste (§.III.4.2) comme indiqué sur le diagramme partiel de la figure V.7.

2) Si l'état initial choisi sur G(K) était z_2, les registres contenant $(z_2 \cup z_3)$ et $(z_2 \cup z_6)$ devraient être l'un et l'autre actifs. Cependant D(K) n'est possibiliste que si les transferts fusionnant en z_J sont étiquetés par le même signal de commande v_h.

3) L'assignation duale se compare au codage de GRAY (b bascules) et au codage primitif (n bascules) des états comme suit : $\log_2 n \leqslant b \leqslant n \leqslant d_S \leqslant 2^n - 2$. (64

COROLLAIRE DE LA COASSIGNATION. V.3.

La forme canonique de coassignation R(K) permet le déroulement de microalgorithmes simultanés synchrones et le séquencement collatéral dans un même microalgorithme.

Démonstration.

Contrairement au cas de la microprogrammation [WIL.51], aucune sélecti d'adresse n'est effectuée en entrée de la mémoire de dynamique Mb, de sorte que plusieurs fonctions de transition τ_i peuvent être simultanément exécutées sur une même forme canonique R(K).

Soient K_1 et K_2 avec d_1 et d_2 coétats respectivement ; alors si un macromodule KD satisfait la relation $d \geqslant d_1 + d_2$, il peut supporter la programmation simultanée de K_1 et K_2.

Ceci revient à une décomposition en somme directe (Théorème II.3) d'un polymate, que le théorème des appels imbriqués (§.IV.4.2) permet d'exploiter pour le séquencement collatéral.

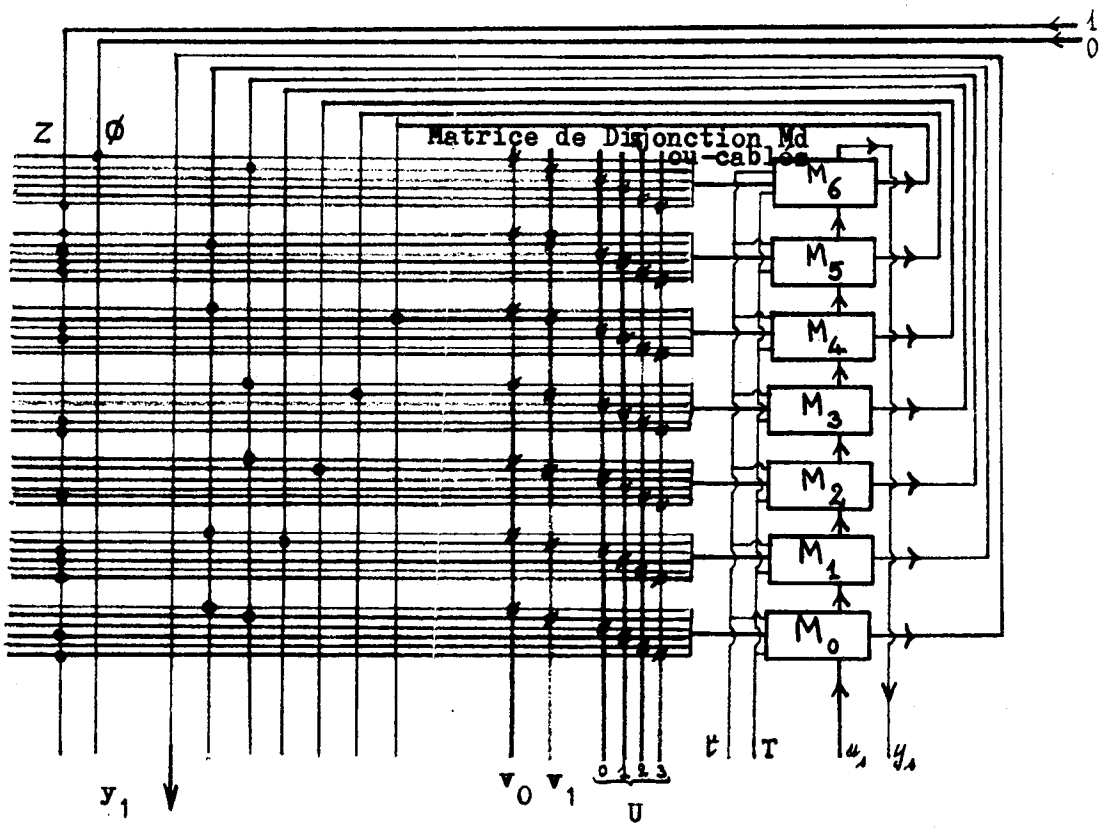
C.Q.F.D.

THEOREME DE LA PROGRAMMATION SANS DIODES. V.4.

La forme canonique de coassignation peut être implémentée sur un module universel KD à logique séquentielle programmable sans diodes.

Démonstration.

Les circuits actuels à haut degré d'intégration requièrent des points programmables de connexion de diodes pour permettre un logique universelle. C'est le cas des PROM ("Programmable Read-Only Memory"), des AROM ("Associative Read-only Memory") et des PLA ("Programmable Logic Array"). Mais la coassignation entraîne d'une part que Mb est monomiale en ligne, d'autre part que les entrées U de forçage parallèle peuvent être traitées comme des entrées V agissant exclusivement sur le coétat Z ainsi que le montre le schéma suivant. Sur le plan technologique le lecteur peut consulter [THU.71].



Matrice programmable sans diodes Mp.

(M_7 a pu être supprimé ici car y_0 est l'inverse logique de y_1).

Figure V.8. Forme canonique KD de coassignation programmable sans diodes.

La fonction de rythme est rendue inutile par le fait-même que le codage de l'alphabet d'entrée {U,V} est primitif (1 parmi m + q). Soit Mp la nouvelle matrice de dynamique sans diodes. Elle comporte d+2 colonnes mais d × (m+q) lignes alors que Mb en avait d × (q+1) ; par contre Mc avait d lignes et m colonnes.

C.Q.F.D.

Le lecteur intéressé à des détails concrets d'application technologiqu de ces résultats peut consulter [DEP.74.b,c et d].

LEMME DE MINIMISATION.V.5.

La construction de la table de coassignation D fournit une réalisation à nombre minimal de coétats, que le microalgorithme donné initialement pour K soit en forme minimale ou non.

Démonstration.

La relation sortie-entrée $\equiv_D$ telle que $w \equiv_D w' \Leftrightarrow \sigma(\tau(z_i,w)) = \sigma(\tau(z_i,w'))$, $\forall z_i \in Z$ induit l'espace de coétat $\mathbb{W} \triangleq V^*/\equiv_D$ où V^* est l'itéré de V et $\mathbb{W}$ est indépendant de la représentation ou de la minimalité de Z.

C.Q.F.D.

CHAPITRE VI

PERSPECTIVES ET DEVELOPPEMENTS

CHAPITRE VI

PERSPECTIVES ET DEVELOPPEMENTS

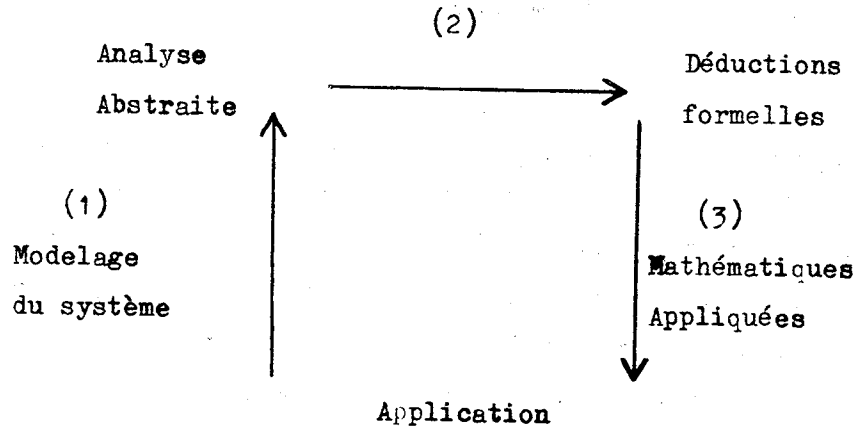
VI.1. L'INGENIERIE DES SYSTEMES FACE AUX MATHEMATIQUES APPLIQUEES.

Nous avons démontré l'adéquation du modèle des polymates au problème de la formalisation de l'architecture des calculateurs dans le cas le plus génér où intervient le traitement parallèle. D'une manière naturelle, nous avons exhib le rôle structurel d'une dualité entre séquencement (temps) et transferts (espac permettant d'exhiber un algorithme de conversion de polymate à transitions d'éta (représentation fonctionnelle d'une unité de commande) en polymate à transferts de registres (représentation topologique de sa réalisation) dans le cas plus général avec séquencement collatéral. Cette conversion établit le lien entre la Dualité en Informatique étudiée dans cette thèse-ci et la Dualité dans les Automates présentée dans "Operand Investigation of Stochastic Systems" [DEP.68] d'un part, et la dualité en analyse spectrale présentée dans "Fondements algébriques de la Transformation de Fourier Rapide" [DEP.70.b - 71.a,b] d'autre part.

La comparaison du théorème V.1. (d'assignation duale) et des travaux initiaux de [WEI.68] illustre comment une nouvelle démonstration d'un théorème existant peut valoir un nouveau théorème, dans la mesure où elle apporte une nouvelle sémantique à des résultats incomplètement exploitables sans celle-ci. En particulier la technique d'assignation duale permet de traiter les transducteurs ($p \geqslant 2$) et non seulement les accepteurs ($p = 2$) ; la coassignation est un substitut potentiel de la microprogrammation pour technologie LSI.

Il est bon à ce point d'insister sur la nature pragmatique des trava présentés.

Figure VI.1. La démarche de l'Ingénierie des Systèmes.



C'est seulement parce que nous avons élaboré un modèle - ou plutôt un gamme de modèles - adéquat pour un problème concret, puis développé un formalis une méthodologie et des résultats sur modèle, et que nous avons finalement appliqué cette méthodologie et ces résultats sur le problème initial, que nous pouvons dire que ce travail relève de l'Ingénierie des Systèmes. Ainsi, il est intéressant de confronter les résultats de [WEI.68] et ceux du chapitre V : la phase (1) permet en effet l'attribution d'une sémantique nouvelle à une structu mathématique qui peut alors être exploitée de manière originale.

Nous venons d'opposer la démarche de l'Ingénierie des Systèmes à la démarche des Mathématiques Appliquées qui est la recherche de domaines d'application pour des résultats abstraits ou des modèles connus initialement.

Il est bon de noter cependant que cette opposition ne doit pas devenir un cloisonnement d'écoles de pensée pour deux raisons : d'abord la démarche des Mathématiques Appliquées est occasionnellement fertile en soi, comme dans l'expérience en [DEP.70.b,c et 71.a,b] ; ensuite, cette démarche est une partie intégrante de l'Ingénierie des Systèmes : par exemple, lorsque la phase (1) de "modelage du système informatique" nous a conduit à spécifier un processeur comm une composition en tourbillon $P = \langle K,D \rangle$, la phase (2) "d'analyse abstraite" a consisté à ramener ce modèle à une généralisation de la théorie mathématique existante des machines séquentielles.

VI.2. ASPECTS LOGISTIQUES DE LA DECOMPOSITION MODULAIRE.

L'intérêt de la décomposition modulaire d'un système n'est pas seulement de permettre la parallélisation et donc l'accélération du traitement, ni de permettre la simplification de la conception et des tests, il est aussi et peut-être surtout de permettre une planification fine rigoureuse. C'est un principe élémentaire de la Théorie de la Décision en présence d'incertitude que de ramener les évaluations au niveau de sous-tâches aussi élémentaires que possible pour deux raisons :

- l'évaluation de la charge et de la durée prévisionnelle de réalisation est plus facile pour une tâche petite,
- la composition des évaluations élémentaires entraîne une compensation entre erreurs d'évaluation et accroît la véracité de l'estimation globale.

Autant les recherches dans le logiciel que dans le matériel ont pendant la dernière décennie placé une emphase excessive sur les aspects syntaxiques du traitement et de la traduction de l'information par opposition à leurs aspects sémantiques. Quant aux problèmes pragmatiques, détachés des précédents, ils ne pouvaient avoir grand sens. C'est pourquoi les techniques algébriques de décomposition [HAR.66] n'ont jamais convaincu les praticiens qui exigent prudemment une démonstration au préalable, que l'apprentissage d'une nouvelle technique est un investissement profitable. En l'occurrence les bases de cette technique (les théories des Automates) ne font généralement pas partie du bagage des concepteurs de machines, quoique elles tendent à être considérées comme des outils fondamentaux pour l'Architecture des Systèmes logiciels [BAR.72].

Surtout nous avons montré qu'une approche sémantique (c'est-à-dire inspirée par la structure duale en D) de la décomposition de K en polymates paramétrés programmables est concevable ; il est alors intéressant de reconsidérer l'un des aspects les plus attirants qui était promis par les techniques algébriques de décomposition et qui demeure applicable :

soit K doté de k états et décomposable en K1 et K2 tels que $k \leqslant k1 . k2$; le nombre d'états de la réalisation décomposée est alors $\tilde{K} = k1 + k2$.

Si la réalisation est faite avec une assignation des états aussi compacte que possible (ce qui n'est faisable qu'en implémentation cablée), alors K et $\tilde{K}$ requièrent pratiquement le même nombre de bascules et les avantages de la décomposition sont de rendre la réalisation plus simple (moins de boucles de réinformation), plus adaptable et diagnosticable (par association du niveau de détection d'erreur à un module de génération, par exemple, sur le paradigme L présenté en [DEP.72] sur la conception structurée).

Mais si l'assignation des états (sur implémentation cablée, microprogrammée,...) est primitive dans chaque sous-module K_i (une bascule DATA par phase), alors le nombre de bascules est réduit considérablement par la décomposition grâce à la relation suivante :

$$k1 + k2 < k1 . k2 \qquad \forall\, k1, k2 > 1 \text{ et } k \geqslant 5.$$

Tout ceci demeure a fortiori dans le cas de la coassignation.

VI.3. EXTENSIONS DES RECHERCHES PRESENTEES.

Il n'est pas essentiel que la coassignation soit d'intérêt industriel immédiat, mais il était important que le concept de polymate fasse la preuve de son efficacité en engendrant des résultats non évidents qui ne pouvaient être atteints clairement sans lui.

L'augmentation de coût résultant de l'emploi de la coassignation avec un nombre de bascules égal à d (à comparer à un minimum de $\log_2 n$ en codage compact de GRAY ou à un nombre n en assignation primitive) est compensée par une réduction de coût associé à l'élimination des diodes aux points-mémoires de la matrice de commande, ainsi que par l'accélération du fonctionnement résultant de la non-sélection d'adresse en entrée de la mémoire de commande et de la régularité fonctionnelle et géométrique de la logique qui permet un ajustement précis du temps de cycle des macromodules KD à leur minimum. En outre toute logique de décodage est éliminée.

De telles considérations d'ordre économique peuvent réserver des surprises dans la pratique de sorte que Vérité en 1974 peut être Mensonge en 1984 et la Recherche ne peut être jugée avec la même mesure que l'Industrie. Ce qui est fondamental alors dans le Concept de polymate c'est donc le rapprochement mathématique qu'il permet d'accomplir entre des Ecoles de Pensée divergentes de l'Automatique, de l'Informatique et de l'Algèbre sur le plan des modèles d'une part, entre les théoriciens [TURING, WANG,...] et les praticiens [THURBER, BERG, BELL,...] d'autre part.

REFERENCES BIBLIOGRAPHIQUES

Seuls, les documents en rapport effectif avec cette thèse sont mentionnés et aucun effort d'exhaustivité n'a été entrepris. Le classement est par ordre alphabétique du nom des auteurs puis par date de publication.

Les références sont données par les 3 premières lettres du nom du premier auteur, suivies des deux derniers chiffres de l'année de publication (et d'une lettre minuscule pour différencier éventuellement).

ARBIB,M.A. "Theories of Abstract Automata".
Prentice-Hall Series in Automatic Computation. (1969).

BANCILHON,F.,DEPEYROT,M. "Dispersion matrices and stochastic Automata morphisms" in "Theory of Machines and Computations". Academic Press. pp.153-166. (August 1971.c).

BANCILHON,F.,DEPEYROT,M. "Réalisation d'un modèle stochastique pour un automate non fiable à partir d'expériences d'entrée-sortie". Revue Bleue de l'AFCET. R.A.I.R.O. 8ème Année, B-2, pp. 127-153. (Juin 1974.e).

BARNES,H.B. "A programmer's view of Automata".
Computing Surveys, Vol. 4, nr.1, pp. 221-240. (December 1972).

BELL,C.G.,NEWELL,A. "Computer structures : Readings and examples".
Mc Graw Hill. (1971).

BELL,C.G.,GRASON,J.,NEWELL,A. "Designing computers and digital Systems".
Digital Press. (1972).

BOOLE,G. "An investigation of the laws of thought". London. (1854). [réédition Dover].

CADDEN,W.J. "Equivalent Sequential Circuits". IRE Transactions on Circuit Theory, pp. 30-34. (March 1959).

DEPEYROT,M. "Operand investigation of Stochastic Systems". Stanford Doctoral Dissertation. (May 1968).

DEPEYROT,M. "Un algorithme rapide de décomposition". Symposium International "Systèmes Logiques-Conceptions et Applications" [BRUXELLES]. (Septembre 1969).

DEPEYROT,M. "Un modèle algébrique réaliste de calculateur". Symposium AFCET. [PARIS].S.I-4, pp. 59-73. (Septembre 1970.a).

DEPEYROT,M. "Fondements Algébriques de la Transformation de Fourier Rapide". Cahier n° 3 de l'I.R.I.A. pp. 25-169. (Novembre 1970.b).

DEPEYROT,M. "Linear system identification using real-time deconvolution". IEEE-C-19, nr.12. pp. 1139-1145. (December 1970.c).

DEPEYROT,M.,MONDELLI,J. "Organisation hiérarchique de la Transformation de Fourier Rapide". Automatique. 16e Année, Tome XVI,n° 4, pp. 232-241. (Avril 1971.a).

DEPEYROT,M.,MARMORAT,J.P.,MONDELLI,J. "An Automaton-Theoretic approach to the Fast Fourier Transform". Proceedings of the Symposium on Computers and Automata. Polytechnic Institute of Brooklyn. (13-15 Avril 1971.b).

DEPEYROT,M. "Sur la synchronisation et la fiabilité des automates". R.A.I.R.O. Revue Bleue de l'AFCET. 5ème Année, B-2, pp. 87-110. (Décembre 1971. e).

DEPEYROT,M. "Link Dynamics and Interface Decomposition in PMS". (Notes on Structured Design). Proceedings of the "Advanced EEC Course in Computing System Architecture". [Alpe d'Huez]. (December 1972).

DEPEYROT,M. "Design of Interfaces as stochastic Channels with memory". GRENOBL International Workshop on Computer Architecture. (June 1973.a).

DEPEYROT,M. "Duality and Hierarchy in Data Processors". (to be published). Article IRIA. IA/111.(February 1974.a). Rapport LABORIA.n° 90.

DEPEYROT,M. "Modules universels à logique séquentielle programmable sans diode: pour circuits intégrés à grande échelle". [Brevet d'invention]. (21/3/1974.b). Rapport LABORIA.n° 86.

DEPEYROT,M. "Prolégomènes à une Théorie des Interfaces". Article IRIA. IA/114. (Mars 1974.c).

DEPEYROT,M. "Les Principes de Synchronisation, Dualité et Hiérarchie dans les Systèmes Informatiques". Etude IRIA. IE/52. (Mars 1974.d).

DJATSCHENKO,W.F.,LAZAREW,W.G. "Umformung logischer Algorithmenschemata und Vereinfachung der Struktur von Mikroprogramm-Automaten". EIK, Vol. 4, pp. 173-186. (1968).

FAURRE,P.,DEPEYROT,M. "Note sur les inverses généralisées de matrices". Article de l'I.R.I.A. (Juin 1970).IA/14. (A paraître : 1974.f.).

GERACE,G.B. "Digital System Design Automation-A method for designing a digital system as a sequential network system". IEEE-C. 17, nr. 11, pp. 1044-1061. (November 1968).

HARTMANIS,J.,STEARNS,R.E. "Algebraic Structure Theory of Sequential machines". Prentice Hall. (1966).

HUFFMAN,D.A. "The Synthesis of sequential switching circuits. Journal of the Franklin Institute, Vol. 257, pp. 161-190, 275-303. (1954).

IANOV,I.I. "On the Equivalence and Transformation of program schemes". Com. ACM, Vol. 1, nr. 10, pp. 8-12. (October 1958).

KALMAN,R.E. "Lectures on Controllability and Observability". C.I.M.E. Bologna. (July 1968).

KARP,R.M.,MILLER,R.E.,WINOGRAD,S. "The organization of computations for uniform recurrence equations". Journal ACM. Vol. 14, nr.3. pp. 563-590. (July 1967).

KARP,R.M.,MILLER,R.E. "Parallel Program Schemata". Journal of Comp. and System Sc., Vol. 3, pp. 147-195. (1969).

MARTIN,D.F.,ESTRIN,G. "Models of computations and systems-Evaluation of vertex probabilities in graph models of computations". J. ACM. Vol. 14, nr. 2, pp. 281-299. (April 1967).

MEALY,G.H. "A method for synthesizing sequential circuits". The Bell System Technical Journal, pp. 1045-1079. (Septembre 1955).

MOORE,E.F. (Editor). "Sequential Machines : Selected Papers". Addison-Wesley , (1964).

NERODE,A. "Linear automaton Transformations". Proceedings of the American Mathematical Society, Vol. IX, pp. 541-544. (August 1958).

PAZ,A. "Formal Series, Finiteness Properties and Decision Problems". Technical report nr. 4, Technion-Haifa. (May 1970).

PAZ,A. "Introduction to probabilistic Automata". Academic Press. (1971).

RABIN,M.O.,SCOTT,D. "Finite Automata and their decision problems". IBM Jour.of R and D, Vol. 3, pp. 114-125. (April 1959).

RANEY,G.N. "Sequential Functions". Journal ACM, Vol. 5, pp. 177-180. (1958).

SHANNON,C.E.,WEAVER,W. "The Mathematical Theory of Communication. The University of Illinois Press. URBANA. Illinois. (1949).

SHANNON,C.E.,Mc CARTHY,J. (Editors). "Automata Studies". Annals of Mathematics Studies. Princeton University Press, nr. 34. (1956).

THURBER,K.J.,BERG,R.O. "Universal Logic modules implemented using LSI memory techniques". AFIPS.FJCC. pp. 177-194. (1971).

TURING,A.M. "On Computable numbers, with an application to the Entscheidunsgprol Proc. London Math. Soc., Vol. 42, Series 2, pp. 230-265, (1936) and 544-546. (1937).

WANG,H."A variant to Turing's Theory of computing machines". J. ACM., Vol. 4, nr. 1, pp. 63-92. (1957).

WEINER,P.,HOPCROFT,J.E. "Bounded fan-in, bounded fan-out, uniform decompositions of sychronous sequential machines". IEEE. Proc. Letters, Vol. 56 , pp. 1219-1220. (July 1968).

WILKES,M.V. "The Best way to design an automatic Calculating machine". Manchester University computer inaugural Conference. pp. 16-18. (July 1951).

WIRTH.N. "A note on Program Structures for Parallel Processing". Com. ACM, Vol. 9, nr. 5, pp. 320-321. (May 1966).