USTL FLANDRES ARTOIS

LIFL U.A 369 du C.N.R.S

LABORATOIRE D'INFORMATIQUE FONDAMENTALE DE LILLE

N° d'ordre : 363

THÈSE

Nouveau Régime

présentée à

L'UNIVERSITE DES SCIENCES ET TECHNIQUES DE LILLE FLANDRES ARTOIS

pour obtenir le titre de

DOCTEUR en INFORMATIQUE

par

Yves ROOS

CONTRIBUTION A L'ETUDE DES FONCTIONS DE COMMUTATION PARTIELLE

Thèse soutenue le 8 juin 1989 devant la commission d'Examen

Membres du jury

Président	M. DAUCHET
Rapporteurs	R. CORI
	G. JACOB
Directeur de thèse	M. LATTEUX
Examinateurs	M. CLERBOUT
	D. PERRIN

UNIVERSITE DES SCIENCES
ET TECHNIQUES DE LILLE
FLANDRES ARTOIS

DOYENS HONORAIRES DE L'ANCIENNE FACULTE DES SCIENCES

M.H. LEFEBVRE, M. PARREAU.

PROFESSEURS HONORAIRES DES ANCIENNES FACULTES DE DROIT ET SCIENCES ECONOMIQUES, DES SCIENCES ET DES LETTRES

MM. ARNOULT, BONTE, BROCHARD, CHAPPELON, CHAUDRON, CORDONNIER, DECUYPER, DEHEUVELS, DEHORS, DION, FAUVEL, FLEURY, GERMAIN, GLACET, GONTIER, KOURGANOFF, LAMOTTE, LASSERRE, LELONG, LHOMME, LIEBAERT, MARTINOT-LAGARDE, MAZET, MICHEL, PEREZ, ROIG, ROSEAU, ROUELLE, SCHILTZ, SAVARD, ZAMANSKI, Mes BEAUJEU, LELONG.

PROFESSEUR EMERITE

M. A. LEBRUN

ANCIENS PRESIDENTS DE L'UNIVERSITE DES SCIENCES ET TECHNIQUES DE LILLE

MM. M. PAREAU, J. LOMBARD, M. MIGEON, J. CORTOIS.

PRESIDENT DE L'UNIVERSITE DES SCIENCES ET TECHNIQUES DE LILLE FLANDRES ARTOIS

M. A. DUBRULLE.

PROFESSEURS - CLASSE EXCEPTIONNELLE

M. CONSTANT Eugène	Electronique
M. FOURET René	Physique du solide
M. GABILLARD Robert	Electronique
M. MONTREUIL Jean	Biochimie
M. PARREAU Michel	Analyse
M. TRIDOT Gabriel	Chimie Appliquée

PROFESSEURS - 1ère CLASSE

M. BACCHUS Pierre	Astronomie
M. BIAYS Pierre	Géographie
M. BILLARD Jean	Physique du Solide
M. BOILLY Bénoni	Biologie
M. BONNELLE Jean-Pierre	Chimie-Physique
M. BOSCQ Denis	Probabilitéss
M. BOUGHON Pierre	Algèbre
M. BOURIQUET Robert	Biologie Végétale
M. BREZINSKI Claude	Analyse Numérique

M. BRIDOUX Michel	Chimie-Physique
M. CELET Paul	Géologie Générale
M. CHAMLEY Hervé	Géotechnique
M. COEURE Gérard	Analyse
M. CORDONNIER Vincent	Informatique
M. DAUCHET Max	Informatique
M. DEBOURSE Jean-Pierre	Gestion des Entreprises
M. DHAINAUT André	Biologie Animale
M. DOUKHAN Jean-Claude	Physique du Solide
M. DYMENT Arthur	Mécanique
M. ESCAIG Bertrand	Physique du Solide
M. FAURE Robert	Mécanique
M. FOCT Jacques	Métallurgie
M. FRONTIER Serge	Ecologie Numérique
M. GRANELLE Jean-Jacques	Sciences Economiques
M. GRUSON Laurent	Algèbre
M. GUILLAUME Jean	Microbiologie
M. HECTOR Joseph	Géométrie
M. LABLACHE-COMBIER Alain	Chimie Organique
M. LACOSTE Louis	Biologie Végétale
M. LAVEINE Jean-Pierre	Paléontologie
M. LEHMANN Daniel	Géométrie
Mme LENOBLE Jacqueline	Physique Atomique et Moléculaire
M. LEROY Jean-Marie	Spectrochimie
M. LHOMME Jean	Chimie Organique Biologique
M. LOMBARD Jacques	Sociologie
M. LOUCHEUX Claude	Chimie Physique
M. LUCQUIN Michel	Chimie Physique
M. MACKE Bruno	Physique Moléculaire et Rayonnements Atmosph.
M. MIGEON Michel	E.U.D.I.L.
M. PAQUET Jacques	Géologie Générale
M. PETIT Francis	Chimie Organique
M. POUZET Pierre	Modélisation - calcul Scientifique
M. PROUVOST Jean	Minéralogie
M. RACZY Ladislas	Electronique
M. SALMER Georges	Electronique
M. SCHAMPS Joel	Spectroscopie Moléculaire
M. SEGUIER Guy	Electrotechnique
M. SIMON Michel	Sociologie
Melle SPIK Geneviève	Biochimie
M. STANKIEWICZ François	Sciences Economiques
M. TILLIEU Jacques	Physique Théorique
M. TOULOTTE Jean-Marc	Automatique
M. VIDAL Pierre	Automatique
M. ZEYTOUNIAN Radyadour	Mécanique

PROFESSEURS - 2ème CLASSE

M. ALLAMANDO Etienne	Composants Electroniques
M. ANDRIES Jean-Claude	Biologie des organismes
M. ANTOINE Philippe	Analyse
M. BART André	Biologie animale
M. BASSERY Louis	Génie des Procédés et Réactions Chimiques

Mme BATTIAU Yvonne	Géographie
M. BEGUIN Paul	Mécanique
M. BELLET Jean	Physique Atomique et Moléculaire
M. BERTRAND Hugues	Sciences Economiques et Sociales
M. BERZIN Robert	Analyse
M.BKOUCHE Rudolphe	Algèbre
M. BODARD Marcel	Biologie Végétale
M. BOIS Pierre	Mécanique
M. BOISSIER Daniel	Génie Civil
M. BOIVIN Jean-Claude	Spectroscopie
M. BOUQUELET Stéphane	Biologie Appliquée aux enzymes
M. BOUQUIN Henri	Gestion
M. BRASSELET Jean-Paul	Géométrie et Topologie
M. BRUYELLE Pierre	Géographie
M. CAPURON Alfred	Biologie Animale
M. CATTEAU Jean-Pierre	Chimie Organique
M. CAYATTE Jean-Louis	Sciences Economiques
M. CHAPOTON Alain	Electronique
M. CHARET Pierre	Biochimie Structurale
M. CHIVE Maurice	Composants Electroniques Optiques
M. COMYN Gérard	Informatique Théorique
M. COQUERY Jean-Marie	Psychophysiologie
M. CORIAT Benjamin	Sciences Economiques et Sociales
Mme CORSIN Paule	Paléontologie
M. CORTOIS Jean	Physique Nucléaire et Corpusculaire
M. COUTURIER Daniel	Chimie Organique
M. CRAMPON Norbert	Tectolique Géodynamique
M. CROSNIER Yves	Electronique
M. CURGY Jean-Jacques	Biologie
Melle DACHARRY Monique	Géographie
M. DEBRABANT Pierre	Géologie Appliquée
M. DEGAUQUE Pierre	Electronique
M. DEJAEGER Roger	Electrochimie et Cinétique
M. DELAHAYE Jean-Paul	Informatique
M. DELORME Pierre	Physioliogie Animale
M. DELORME Robert	Sciences Economiques
M. DEMUNTER Paul	Sociologie
M. DENEL Jacques	Informatique
M. DE PARIS Jean Claude	Analyse
M. DEPREZ Gilbert	Physique du Solide - Cristallographie
M. DERIEUX Jean-Claude	Microbiologie
Melle DESSAUX Odile	Spectroscopie de la réactivité Chimique
M. DEVRAINNE Pierre	Chimie Minérale
Mme DHAINAUT Nicole	Biologie Animale
M. DHAMELINCOURT Paul	Chimie Physique
M. DORMARD Serge	Sciences Economiques
M. DUBOIS Henri	Spectroscopie Hertzienne
M. DUBRULLE Alain	Spectroscopie Hertzienne
M. DUBUS Jean-Paul	Spectrométrie des Solides
M. DUPONT Christophe	Vie de la firme (I.A.E.)
Mme EVRARD Micheline	Génie des procédés et réactions chimiques
M. FAKIR Sabah	Algèbre
M. FAUQUAMBERGUE Renaud	Composants électroniques

M. FONTAINE Hubert	Dynamique des cristaux
M. FOUQUART Yves	Optique atmosphérique
M. FOURNET Bernard	Biochimie Sturcturale
M. GAMBLIN André	Géographie urbaine, industrielle et démog.
M. GLORIEUX Pierre	Physique moléculaire et rayonnements Atmos.
M. GOBLOT Rémi	Algèbre
M. GOSSELIN Gabriel	Sociologie
M. GOUDMAND Pierre	Chimie Physique
M. GOURIEROUX Christian	Probabilités et Statistiques
M. GREGORY Pierre	I.A.E.
M. GREMY Jean-Paul	Sociologie
M. GREVET Patrice	Sciences Economiques
M. GRIMBLOT Jean	Chimie Organique
M. GUILBAULT Pierre	Physiologie animale
M. HENRY Jean-Pierre	Génie Mécanique
M. HERMAN Maurice	Physique spatiale
M. HOUDART René	Physique atomique
M. JACOB Gérard	Informatique
M. JACOB Pierre	Probabilités et Statistiques
M. Jean Raymond	Biologie des populations végétales
M. JOFFRE Patrick	Vie de la firme (I.A.E.)
M. JOURNEL Gérard	Spectroscopie hertzienne
M. KREMBEL Jean	Biochimie
M. LANGRAND Claude	Probabilités et statistiques
M. LATTEUX Michel	Informatique
Mme LECLERCQ Ginette	Catalyse
M. LEFEBVRE Jacques	Physique
M. LEFEBVRE Christian	Pétrologie
Melle LEGRAND Denise	Algèbre
Melle LEGRAND Solange	Algèbre
M. LEGRAND Pierre	Chimie
Mme LEHMANN Josiane	Analyse
M. LEMAIRE Jean	Spectroscopie hertzienne
M. LE MAROIS Henri	Vie de la firme (I.A.E.)
M. LEROY Yves	Composants électroniques
M. LESENNE Jacques	Systèmes électroniques
M. LHENAFF René	Géographie
M. LOCQUENEUX Robert	Physique théorique
M. LOSFELD Joseph	Informatique
M. LOUAGE Francis	Electronique
M. MAHIEU Jean-Marie	Optique-Physique atomique
M. MAIZIERES Christian	Automatique
M. MAURISSON Patrick	Sciences Economiques et Sociales
M. MESMACQUE Gérard	Génie Mécanique
M. MESSELYN Jean	Physique atomique et moléculaire
M. MONTEL Marc	Physique du solide
M. MORCELLET Michel	Chimie Organique
M. MORTREUX André	Chimie Organique
Mme MOUNIER Yvonne	Physiologie des structures contractiles
Mme MOUYART-TASSIN Annie Françoise	Informatique
M. NICOLE Jacques	Spectrochimie
M. NOTELET Françis	Systèmes électroniques
M. PARSY Fernand	Mécanique

M. PECQUE Marcel	Chimie organique
M. PERROT Pierre	Chimie appliquée
M. STEEN Jean-Pierre	Informatique

Je remercie Max Dauchet qui a accepté de présider le jury.

Je remercie Robert Cori et Gérard Jacob qui m'ont fait l'honneur de bien vouloir être rapporteurs pour cette thèse.

Je remercie également Dominique Perrin d'avoir accepté de participer au jury.

Je tiens particulièrement à exprimer toute ma reconnaissance à Michel Latteux qui a dirigé mes travaux. Ses encouragements, ses qualités humaines et sa passion contagieuse pour la recherche ont permis que cette thèse aboutisse.

Je tiens à associer à ces remerciements Mireille Clerbout qui, au delà des relations de travail, m'a permis de trouver au sein du LIFL une équipe d'amis appartenant aux différents domaines que cache l'unique terme d'*informatique*.

Je remercie Henri Glanc qui a réalisé le tirage de cette thèse .

Je remercie enfin mes proches, mes amis du bureau 336 et ceux de plus longue date pour leur soutien et leurs encouragements.

Table des matières

Introduction.

Pour les premières générations d'ordinateurs, la théorie des automates est un outil formel permettant de représenter le comportement des programmes séquentiels se déroulant sur ces machines mono-utilisateurs. Puis sont apparues les notions de parallélisme, de processus concurrents, de systèmes répartis, de réseaux, nécessitant souvent la mise en place de mécanismes complexes de synchronisation ou d'exclusion mutuelle pour l'accès à une ressource.

Un des modèles les plus utilisés pour représenter le comportement de processus parallèles est celui des réseaux de Pétri introduits par C.A. Pétri en 1960. Plus récemment, pour décrire le comportement d'un réseau de Pétri, Mazurkiewicz a utilisé dans [24] la notion de trace et de langage trace introduite, pour des problèmes de combinatoire, par Cartier et Foata dans [7] en 1969. Une trace étant un ensemble partiellement ordonné de lettres, représentant des noms d'actions, elle peut être considérée comme un modèle pour des processus concurrents.

Le fait que deux actions peuvent être effectuées simultanément peut être formalisé par une relation de commutation liant les lettres représentant ces actions. Cette relation exprime qu'il est indifférent d' effectuer ces actions dans un ordre précis. Plus précisément, si X est un alphabet, une relation de **commutation partielle** C définie sur X est une relation irréflexive et symétrique. Les traces sont les classes d'équivalence de la congruence engendrée par la relation C sur X^*.

Dans [8], M. Clerbout a introduit et étudié deux nouvelles familles de relations de commutation: la sous-famille des relations de **commutation partitionnée** dont la relation complémentaire est une relation d'équivalence, et la famille des relations de **semi-commutation**, versions non symétriques des relations de commutation partielle.

On peut identifier une relation de semi-commutation à un système de réécriture où toutes les règles sont de la forme $xy \longrightarrow yx$, x et y étant deux lettres distinctes. L'opération qui à un mot associe l'ensemble de tous les mots obtenus par réécritures dans le système est appelée fonction de semi-commutation. Ainsi, la règle $ba \longrightarrow ab$ définit sur l'alphabet $\{a,b\}$ une relation de semi-commutation. Alors que les relations de commutation partielle formalisent le parallélisme, les relations de semi-commutation expriment le fait que certaines actions doivent être effectuées avant d'autres. Supposons que la lettre a représente l'action *"produire"* un message dans un buffer, et que la lettre b représente l'action *"consommer"* un message dans ce même buffer. Si le buffer est de taille 1, le langage représentant le bon déroulement des opérations est $(ab)^*$.

Si on considère maintenant un buffer de taille infinie, on peut alors produire plus vite qu'on ne consomme mais on ne peut évidemment pas consommer un message non encore produit. Le langage représentant le bon déroulement des opérations est alors l'image du langage $(ab)^*$ par la fonction de semi-commutation associée, qui est l'union des images des mots du langage. On obtient dans ce cas le langage de semi-Dyck sur une paire de parenthèses.

Comme pour les langages commutatifs (voir les travaux de M. Latteux sur ce sujet, et en particulier, [2], [22], [23]), les relations de commutation partielle et de semi-commutation ont succité récemment de nombreux travaux, dont nous allons rappeler les principaux résultats dans les différents chapitres composant cet ouvrage.

Le premier chapitre est consacré aux notations, définitions et résultats de base qui seront utilisés par la suite. Les deux premières sections rappellent des notions de base de la théorie des langages formels et des graphes. Les sections suivantes sont consacrées à l'introduction des fonctions de commutation. En particulier, la dernière section donne les liens existant entre la famille des **semi-commutations** introduite par M. Clerbout [8], la famille des **commutations partielles**, introduites par P. Cartier et D. Foata [7], et celle des **commutations partitionnées**. Ces résultats, montrés par M. Clerbout [8], ont permis en particulier à D. Perrin [30], de simplifier un certain nombre de preuves concernant les relations de commutation partielle.

Nous introduisons, dans le second chapitre, une technique de **numérotation** des différentes occurrences d'une même lettre dans un mot. Celle - ci est basée sur la définition d'image par **substitution alphabétique** d'une fonction de semi-commutation. Les résultats élémentaires qui y sont présentés nous permettrons de retrouver certaines propriétés déjà connues, vérifiées par les fonctions de semi-commutation. Ils nous permettront également de clarifier un certains nombre de preuves du chapitre suivant.

Ce troisième chapitre est consacré à l'étude des **compositions de fonctions de commutation partitionnée** introduite par M. Clerbout dans [8] et [9]. Dans une première section, nous montrerons qu'un certain nombre de propriétés vérifiées par les fonctions de semi-commutation sont conservées par les compositions de fonctions de commutation partitionnée. Nous verrons qu'il n'en va pas de même pour le **lemme de projection** qui permettait, dans le cas d'une seule fonction de commutation partitionnée f, définie sur un alphabet X, et pour deux mots u et v de X^*, de décider simplement si v appartient à uf.

Le problème reste évidemment décidable: si $f_1 f_2,\ldots f_n$ sont n fonctions de commutation partitionnée définies sur un alphabet X, et u,v deux mots de X^*, il suffit de trouver une suite de mots $w_0,w_1,\ldots,w_n$ vérifiant: $w_0=u$, $w_n=v$, et $\forall\ i \in [1,n]$, $w_i \in w_{i-1} f_i$. Le nombre de mots à tester, fini, est cependant en général très élevé, et il semble souhaitable de se donner des outils permettant de réduire le coût de cette recherche.

Un autre objectif est de parvenir à **décider de l'égalité de plusieurs compositions de fonctions de commutation**. En particulier, peut-on décider simplement si une composition de fonctions de commutation partitionnée réalise la commutation totale?

Aussi, après avoir rappelé les résultats déjà obtenus par M. Clerbout dans ce domaine, nous étudierons en détail le cas d'un alphabet de cinq lettres où nous proposerons un nouvel algorithme qui, pour deux mots u et v et une composition de 3 fonctions de commutation 2-partitionnée f, permet de décider si $v \in u f$. Cet algorithme permettra également d'établir des **comparaisons** entre des compositions et des fonctions simples.

Nous donnerons ensuite un résultat plus général relatif à la décision: $f = com$. Enfin la dernière section de cette partie s'attachera à l'étude des **fonctions de rangement**, qui sont des compositions de fonctions de commutation partitionnée permettant de "*ranger*" un mot quelconque selon un ordre donné sur l'alphabet de travail.

Le quatrième chapitre est consacré à l'étude des fonctions de semi-commutation transformant des langages rationnels en langages algébriques. Dans une première section, nous rappelons les différents résultats concernant la **reconnaissabilité** de la fermeture de langages par fonctions de semi-commutation. Plusieurs résultats concernant à l'origine les commutations partielles restent vrais pour les semi-commutations ([18], [27], [29]).

La plupart des résultats s'exprime sous forme de propriétés du **graphe de commutation** ou du **graphe de non-commutation** associé à chaque fonction de semi-commutation. Si X est un alphabet, f une fonction de semi-commutation définie sur X^* et Y un sous-alphabet de X, le graphe de non-commutation de Y pour f est le graphe orienté ou les noeuds sont les lettres de Y et les arcs sont les couples de lettres distinctes (a,b) telles que $ba \notin ab\, f$. En particulier, Métivier a montré dans [28] que si le graphe de non-commutation de l'alphabet de tout facteur itérant d'un rationnel R est connexe, l'image de R par la fonction f est alors un langage rationnel.

Dans une seconde section, nous caractérisons les fonctions de semi-commutation **algébrico-rationnelles** qui sont des fonctions de semi-commutation f définies sur un alphabet X telles que pour tout langage rationnel $R \subset X^*$, $R f$ est un langage algébrique. La condition s' appuiera dans ce cas sur une propriété du graphe de commutation de X pour f. Cette condition peut s'exprimer de la façon suivante: Une fonction de semi-commutation f, associée à une relation C définie sur un alphabet X est algébrico-rationnelle si et seulement si pour tout couple (a,b) dans C, il n'existe pas c,d non nécessairement distincts dans $X-\{a,b\}$ tels que $(a,c) \in C$ et $(d,b) \in C$.

Dans la troisième et dernière section de ce chapitre, nous donnons une condition nécessaire et suffisante, portant sur une fonction de semi-commutation f définie sur un alphabet X et un mot u de X^*, pour que $u^* f$ soit un langage algébrique: $u^* f$ est algébrique si et seulement si le graphe de non-commutation de l'alphabet de u pour f comporte au plus deux composantes fortement connexes.

Le cinquième et dernier chapitre traite des **automates finis asynchrones** introduits par Zielonka [36] . Les automates finis usuels sont des systèmes centralisés puisque, à chaque instant, on a une information complète sur l'état de l'automate, cette information étant nécessaire pour déterminer le changement d'état provoqué par la lecture d'une lettre. Les automates finis asynchrones permettent, quant à eux, de rendre compte du comportement de systèmes répartis.

Dans une première section, nous rappelons les définitions et les propriétés de base. En particulier Zielonka a montré le remarquable résultat suivant: Tout langage régulier, fermé par une relation de commutation partielle C, peut être reconnu par un automate fini asynchrone pour la relation C.

Intuitivement, dans un automate asynchrone, chaque état est un vecteur et à chaque composante de ce vecteur est associé l'ensemble des lettres de l'alphabet qui peuvent "accéder" à cette composante. Ainsi chaque lettre ne connait de l'état que certaines composantes et ne peut modifier que ces composantes.

Dans la deuxième section, nous introduisons la définition d'**automates virtuellement asynchrones** sur une famille de sous-alphabets. Un automate "classique" est virtuellement asynchrone sur une famille de sous-alphabets F, si il existe un automate F-asynchrone qui lui est isomorphe. Nous donnerons alors une condition nécessaire et suffisante, décidable, pour qu'un automate soit virtuellement F-asynchrone.

La troisième et dernière section est consacrée à l'étude des **automates 2-asynchrones**, qui sont des automates asynchrones où chaque composante est associée à un sous-alphabet contenant au plus deux lettres. Après avoir rappelé un résultat de [16] qui montre que tout automate asynchrone peut être simulé par un automate 2-asynchrone, nous proposerons une preuve différente pour le résultat moins général suivant: Tout langage rationnel défini sur un alphabet X peut être reconnu par un automate 2-asynchrone où l'ensemble des sous-alphabets associés aux composantes d'état est l'ensemble: $\{ \{a,b\} / a \neq b, a,b \in X \}$.

Chapitre 1: Définitions, rappels.

1. Notations.

On notera $\mathbb{N}$, l'ensemble des entiers naturels, et $\mathbb{N}_+$, l'ensemble des entiers naturels non nuls.

Si p et q sont deux éléments de $\mathbb{N}$, $[p,q]$ désignera l'intervalle d'entiers défini par: $[p,q] = \emptyset$ si $q < p$, $[p,q] = \{p, p+1,...,q\}$ sinon.

Pour tout ensemble X, on notera $|X|$, le cardinal de l'ensemble X; $\Delta_X=\{(x,x) / x \in X\}$, la diagonale de l'ensemble X; et 2^X, l'ensemble des parties de X.

Si f est une application d'un ensemble X dans un ensemble Y, et x un élément de X, on notera $x f$ l'image de x par f.

Nous supposerons connus les résultats classiques de la théorie des langages formels [4].

Si X est un alphabet, X^* désignera le monoïde libre engendré par l'alphabet X et ε, le mot vide. Si w est un mot de X^*, x une lettre de X, et L un langage de 2^{X^*} , on notera:

$|w|$, la longueur du mot w;

$|w|_x$, le nombre d'occurrences de la lettre x dans le mot w;

$alph(w)=\{x \in X / |w|_x>0\}$, l'ensemble des lettres occurrant dans w;

$com(w)=\{u \in X^* / \forall x \in X, |u|_x=|w|_x\}$, la clôture commutative du mot w, et, par extension, $com(L)= \bigcup_{w\in L} com(w)$, la clôture commutative du langage L;

$F(w)=\{u \in X^* \ tq \ \exists v, v' \in X^*, \ w=vuv'\}$, l'ensemble des facteurs du mot w, et, par extension, $F(L)= \bigcup_{w\in L} F(w)$, l'ensemble des facteurs du langage L;

$FG(w)=\{u \in X^* \ tq \ \exists v \in X^*, \ w=uv\}$, l'ensemble des facteurs gauches du mot w, et, par extension, $FG(L)= \bigcup_{w\in L} FG(w)$, l'ensemble des facteurs gauches de L.

Pour deux mots w et w' de X^*, $w \sqcup\!\sqcup w'=\{ u_1v_1u_2v_2...u_nv_n / \forall i \in [1,n], u_i \in X^*, v_i \in X^*, w=u_1u_2....u_n, w'=v_1v_2....v_n\}$, désigne le **produit de mélange** (ou shuffle) des mots w et w'.

On dira que w' est un **sous-mot** de w, si il existe $w'' \in X^*$ tel que $w=w' \sqcup\!\sqcup w''$. C'est à dire que w' est obtenu à partir de w en effaçant certaines occurrences des lettres qui le composent.

Un **morphisme** f d'un monoïde X^* dans un monoïde Y^* est une application de X^* dans Y^* vérifiant: $\forall\ u,v \in X^*,\ (u\ v)f=(uf)(vf)$ et $\varepsilon f=\varepsilon$. Il est dit:

alphabétique si $Xf \subset Y \cup \{\varepsilon\}$.
strictement alphabétique si $Xf \subset Y$.
marqué si $X \subset Y$ et $\forall x \in X, \exists u \in (Y-X)^* / xf=xu$.

Si X est un alphabet, et w un mot de X^*, la **projection** du mot w sur un sous-alphabet Y est l'image du mot w par l'homomorphisme noté Π_Y, défini sur X par: $\forall x \in X, \Pi_Y(x)=x$ si $x \in Y$, $\Pi_Y(x)=\varepsilon$ sinon. Dans certains cas, si u est un mot de X^*, on poura noter $w\,\Pi_u$ la projection du mot w sur le sous-alphabet $alph(w)$.

Une **substitution** s de X^* dans 2^{Y^*} est une application vérifiant: $\forall\ u,v \in X^*,\ (u\ v)s=(us)(vs)$ et $\varepsilon s=\varepsilon$. Cette substitution est **alphabétique** si $\forall x \in X, xs \subset Y \cup \{\varepsilon\}$.

On notera $D_1^*(x,y)=\{w \in \{x,y\}^* / |w|_x=|w|_y\}$, le langage de Dyck sur l'alphabet $\{x,y\}$ et $D_1'^*(x,y)=\{w \in \{x,y\}^* / |w|_x=|w|_y$ *et* $\forall w' \in FG(w), |w'|_x \geq |w'|_y\}$, le langage de semi-Dyck sur l'alphabet $\{x,y\}$. Quand il ne sera pas nécessaire de préciser l'alphabet, nous noterons D_1^* (respectivement $D_1'^*$) le langage de Dyck (respectivement semi-Dyck) sur deux lettres.

Enfin, on notera $\boldsymbol{RAT}$, la famille des langages rationnels, $\boldsymbol{ALG}$, celle des langages algébriques et $\boldsymbol{Ocl}$, la plus petite famille de langages contenant $D_1'^*$ et fermée par transduction rationnelle, produit, union et étoile, c'est à dire la famille des langages à un compteur.

2. Graphes.

Nous présentons dans cette section quelques définitions de base de la théorie des graphes. La terminologie utilisée est celle de [3].

Un **graphe** G est un couple (S,A) où S est un ensemble fini de **sommets** et A est une partie de $S \times S$ dont les éléments sont appelés **arcs**. Une **arête** est une paire de sommets réunis par au moins un arc. Cette notion est particulièrement utile quand A est une partie symétrique de $S \times S$, G est alors appelé **graphe non orienté**.

Un graphe G est dit **étiqueté** sur un ensemble fini E, si chaque arc (ou arête) de G est étiqueté par un élément de E.

On appelle **graphe partiel** d'un graphe $G=(S,A)$ tout graphe $G'=(S,A')$ où $A' \subset A$. Un **sous-graphe** d'un graphe $G=(S,A)$ est un graphe $G''=(S',A'')$ vérifiant $S' \subset S$ et $A''=A \cap S'xS'$.

L'**union** de d'un graphe $G=(S,A)$ et d'un graphe $G'=(S',A')$ est le graphe $G \cup G'=(S \cup S',A \cup A')$. Dans le cas particulier où les graphes G et G' ont le même ensemble de sommets, nous parlerons alors de la **superposition** des graphes G et G', qui est donc le graphe $(S,A \cup A')$.

Un **chemin** dans un graphe est une suite d'arcs telle que l'extrémité terminale de chaque arc coïncide avec l'extrémité initiale du suivant. Une **chaîne** est une suite d'arêtes telle que chaque arête ait une extrémité commune avec la suivante.

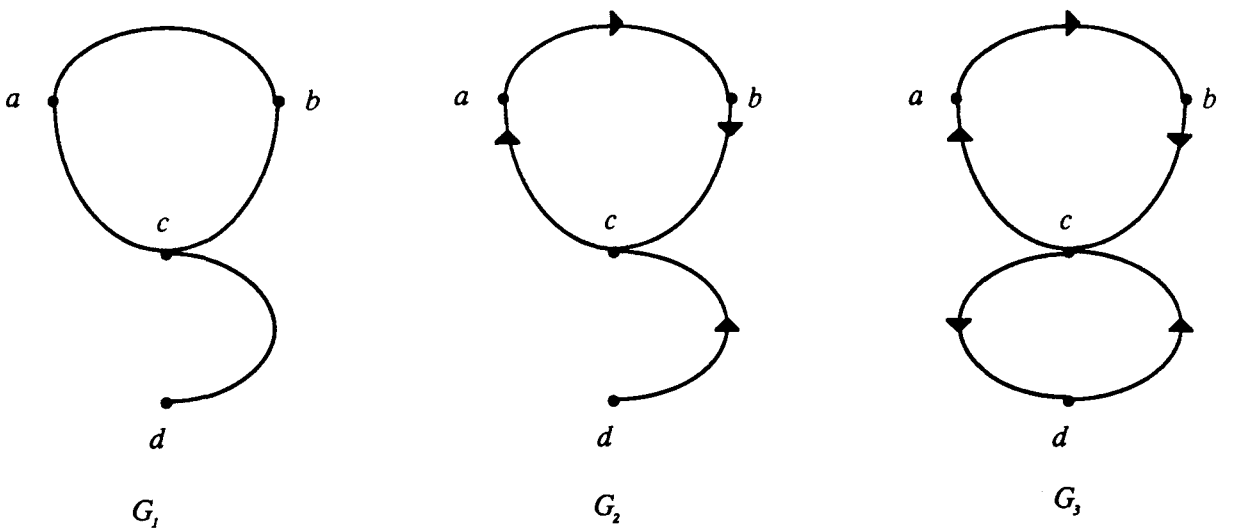
Un chemin est **simple** si il ne contient pas deux fois le même arc; il est **élémentaire** si il ne passe pas deux fois par le même sommet. Un **circuit** est un chemin fini dont le sommet initial et le sommet terminal coïncident. La **longueur** d'un chemin est le nombre d'arcs qui le composent.

Dans le cas où on considère les arêtes d'un graphe, les termes de chemin et de circuit prennent respectivement les noms de **chaîne** et de **cycle**.

Un graphe $G=(S,A)$ où pour toute paire de sommets (x,y) il existe au moins une chaîne réunissant x et y, est dit **connexe**. Si il existe au moins un chemin de x vers y et au moins un chemin de y vers x, G est dit **fortement connexe**. Le graphe G est **complet** si $A=SxS$. L'ensemble des sommets de tout sous-graphe complet de G est appelé **clique** du graphe G. Une famille $\{S_1,S_2,\ldots,S_n\}$ de parties de l'ensemble des sommets S est un **recouvrement** du graphe G si: $\forall (x,y) \in A, \exists i \in [1,n] / \{x,y\} \subset S_i$.

Soit G un graphe, et Γ la relation d'équivalence définie sur l'ensemble S des sommets du graphe par: $\forall (x,y) \in SxS$, $(x,y) \in \Gamma$ si il existe dans G un chemin de x vers y, et un chemin de y vers x. Une **composante fortement connexe** du graphe G est un sous-graphe dont les nœuds sont tous les sommets appartenant à une même classe d'équivalence pour la relation Γ. Dans la suite, on confondra, sous une seule dénomination, une composante fortement connexe et l'ensemble de ses sommets. Dans le cas où on considère non plus des chemins mais des chaînes, on parle alors de **composantes connexes**.

Exemple 2.1: Considérons les graphes G_1,G_2, et G_3 ci-dessous. Le graphe G_1 est connexe, le graphe G_3 est fortement connexe, mais le graphe G_2 n'est pas fortement connexe. Le sous-graphe du graphe G_3 engendré par le sous-ensemble de sommets $T=\{a,b,c\}$ est une composante fortement connexe de G_3, et T est une clique (maximale) du graphe G_1.



G_1 G_2 G_3

3. Semi-commutations.

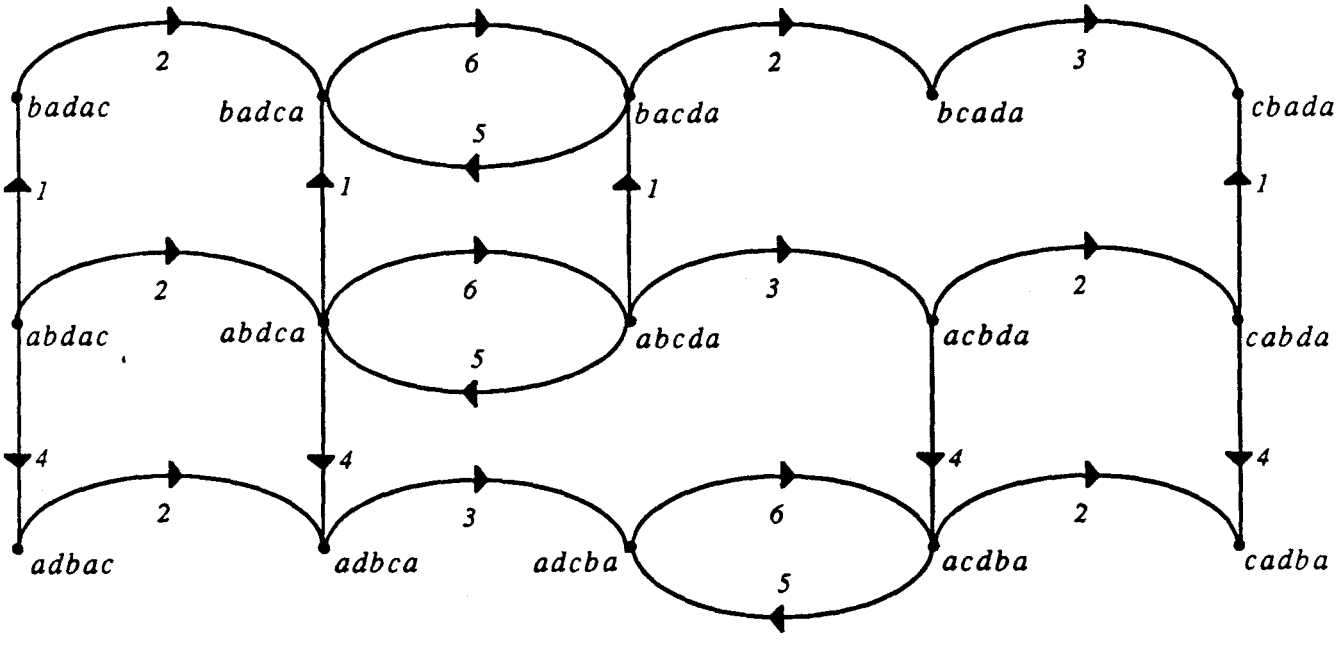
Définition 3.1: *Une **relation de semi-commutation** C sur un alphabet X, est un sous-ensemble de $X\times X-\Delta_X$. La relation complémentaire de C: $\bar{C}=X\times X-C$, est appelée **relation de non-commutation** (associée à C).*

Définition 3.2: *Soit C une relation de semi-commutation définie sur un alphabet X. Le **système de semi-commutation** $S_C=\langle X,R_C\rangle$ associé à la relation C est le système de réécriture où R_C est l'ensemble des règles défini par: $R_C=\{xy\rightarrow yx, (x,y)\in C\}$.*

Exemple 3.3: Soit $C=\{(a,b),(a,c),(b,c),(b,d),(c,d),(d,c)\}$ définie sur $X=\{a,b,c,d\}$. Le système de semi-commutation associé est le système $S_C=\langle X,R_C\rangle$, où R_C comporte les 6 règles:

$$ab \xrightarrow[1]{} ba,$$
$$ac \xrightarrow[2]{} ca,$$
$$bc \xrightarrow[3]{} cb,$$
$$bd \xrightarrow[4]{} db,$$
$$cd \xrightarrow[5]{} dc,$$
$$dc \xrightarrow[6]{} cd.$$

Considérons le mot $u=abdac$. L'ensemble des réécritures successives obtenues à partir de u dans le système S_C est donné par le graphe suivant:



Etant donné un système de semi-commutation $S_C=\langle X,R_C\rangle$, et deux mots w et w' de X^*, nous dirons que w se dérive directement en w' dans le système S_C, et on notera $w \xrightarrow[C]{} w'$, si il existe des mots v_1, v_2 dans X^*, et deux lettres x,y dans X vérifiant: $w=v_1xyv_2$, $w'=v_1yxv_2$ et $xy \longrightarrow yx \in R_C$. Nous dirons que w se dérive en w' dans le système S_C, et on notera $w \xrightarrow[C]{*} w'$, si il existe des mots $w_1, w_2, \ldots, w_p$ dans X^* vérifiant: $w_1=w$, $w_p=w'$, $\forall\, i \in [1,p\text{-}1]$, $w_i \xrightarrow[C]{} w_{i+1}$.

Remarque: Si il n'y a aucune ambiguïté sur la relation de semi-commutation utilisée, nous noterons plus simplement $w \xrightarrow{*} w'$ (respectivement $w \longrightarrow w'$) si w se dérive (respectivement se dérive directement) en w' dans le système de semi-commutation.

Nous pouvons maintenant définir la fonction de semi-commutation associée à la relation C.

Définition 3.4: *Soient X un alphabet et C une relation de semi-commutation définie sur X. La fonction de semi-commutation associée à la relation C est la fonction f_C: $X^* \rightarrow 2^{X^*}$, définie par: $\forall w \in X^*$, $w f_C = \{w' \in X^* / w \stackrel{*}{\underset{C}{\rightarrow}} w'\}$.*

Par extension, nous pouvons définir l'image d'un langage par une fonction de semi-commutation: Si X est un alphabet, et f_C une fonction de semi-commutation définie sur X^*, on notera pour tout langage L inclus dans X^*, $L f_C = \bigcup_{w \in L} w f_C$.

La caractérisation suivante, due à M. Clerbout [8], (voir également [18] pour le cas des commutations partielles), permet de décider si un mot appartient à l'image par une fonction de semi-commutation d'un autre mot en limitant la vérification à leurs projections sur les couples de lettres de l'alphabet.

Lemme 3.5 (lemme de projection): *Soient f une fonction de semi-commutation définie sur un alphabet X et w ,w', deux mots de X^*,alors:$w' \in wf$ si et seulement si $\forall x,y \in X$, $w' \Pi_{xy} \in w \Pi_{xy} f$.*

4. Produit de mixage.

L'image d'un mot par une fonction de semi-commutation peut également être décrite grâce au produit de mixage, introduit par De Simone dans [19]. (voir également [35] et [20]).

Définition 4.1: *On appelle produit de mixage de deux langages L et L', l'ensemble $L \amalg L'$ défini par $L \amalg L' = \{ w \in (X \cup X')^* / w \Pi_X \in L$ et $w \Pi_{X'} \in L' \}$ où $X=alph(L)$ et $X'=alph(L')$.*

Pour des langages L et L', on vérifie facilement les propriétés suivantes: $(alph(L) \cap alph(L')=\varnothing) \Rightarrow (L \amalg L' = L \sqcup L')$ et $(alph(L)=alph(L')) \Rightarrow (L \amalg L' = L \cap L')$. En revanche, l'exemple suivant montre qu'on n'a pas, en général

$$L \amalg L' = \bigcup_{(w,w') \in L \times L'} (\{w\} \amalg \{w'\})$$

Exemple 4.2: Soient $L=\{ab,abc\}$ et $L'=\{cab\}$. Comme $alph(L)=alph(L')=\{a,b,c\}$, on obtient $L \sqcap L' = L \cap L' = \varnothing$, alors que $\{ab\} \sqcap \{cab\} = \{cab\}$.

Cependant, on montre facilement que l'égalité est vérifiée si pour tout mot w de L, $alph(w)=alph(L)$ et pour tout mot w' de L', $alph(w') =alph(L')$. En notant plus simplement, pour deux mots w et w', $w \sqcap w'$ le produit de mixage des langages $\{w\}$ et $\{w'\}$, on peut montrer, par induction sur $|alph(w)|$ et à partir de la remarque précédente et du *lemme 3.5*:

Lemme 4.3: *Soient f une fonction de semi-commutation définie sur un alphabet X et w un mot de X^*, alors on a l'égalité* $wf = \underset{xy \in X}{\sqcap} (w \, \Pi_{xy}) \, f$

5. Commutations partielles.

Une relation de semi-commutation ayant la propriété d'être symétrique est appelée **relation de commutation partielle** ([7], [18]). Le système et la fonction associés sont alors appelés système de commutation partielle et fonction de commutation partielle, et nous noterons CP, l'ensemble des fonctions de commutation partielle. Dans ce cas, le *lemme 3.5* peut s'énoncer de manière plus précise: (voir [18])

Lemme 5.1: *Soient X un alphabet, C une relation de commutation partielle sur X, et w,w' deux mots de X^*, alors $w' \in wf_C$ si et seulement si $\forall (x,y) \in \bar{C}$, $w' \Pi_{xy}=w \Pi_{xy}$ où f_C est la fonction de commutation partielle associée à la relation C.*

6. Commutations partitionnées.

Les **relations de commutation partitionnée** sont un cas particulier de relations de commutation partielle, donc de semi-commutation. Il s'agit de relations dont le complémentaire est une relation d'équivalence.

Si C est une relation de commutation partitionnée sur un alphabet X, les classes d'équivalence de la relation $\bar{C}$ définissent alors une partition de l'alphabet X: $\{X_1, X_2, ..., X_k\}$. Nous pouvons alors donner la définition équivalente suivante:

Une relation $C \subset X \times X$ est une relation de commutation partitionnée si il existe une partition de X, $\{X_1, X_2, ..., X_k\}$ telle que: $C=\{ (x,y) \in X \times X / \exists i,j \in [1,k], i \neq j, x \in X_i, y \in X_j \}$.

Nous dirons alors que la relation C est une relation de commutation k-partitionnée, associée à la partition $\{X_1, X_2, ..., X_k\}$. Le système et la fonction associés sont appelés système de commutation k-partitionnée et fonction de commutation k-partitionnée. Pour tout entier strictement positif k, nous noterons P_k l'ensemble des fonctions de commutation k-partitionnée et $P= \bigcup_{k \in N_*} P_k$, l'ensemble des fonctions de commutation partitionnée.

Toute relation de commutation partitionnée étant une relation de commutation partielle dont le complémentaire est une relation transitive, le *lemme 3.5* prend alors la forme suivante:

Lemme 6.1: *Soient X un alphabet, f une fonction de commutation partitionnée définie sur X^*, associée à la partition $\{X_1, X_2, ..., X_k\}$ et w ,w' deux mots de X^*, alors: $w' \in wf$ si et seulement si $\forall i \in [1,k]$, $w' \Pi_{X_i} = w \Pi_{X_i}$.*

Remarques:

- Le résultat précédent peut s'énoncer de manière équivalente de la façon suivante: Si f est une fonction de commutation partitionnée associée à une partition $\{X_1, X_2, ..., X_k\}$ d'un alphabet X et w un mot de X^*, $wf = w_1 \sqcup w_2 \sqcup ... \sqcup w_k$ où $\forall i \in [1,k]$, $w_i = w \Pi_{X_i}$.

- La fonction de commutation totale sur X^* peut être considérée comme la fonction de commutation partitionnée associée à la partition de X: $\{ \{x\}, x \in X \}$.

7. Liens entre les différentes familles de commutations.

Dans [8], M. Clerbout a étudié les liens existant entre les familles SC, CP, P. Ainsi, la proposition suivante montre qu'il est possible de simuler toute fonction de commutation partielle à partir d'une fonction de commutation partitionnée.

Proposition 7.1 [8]: *Pour toute fonction de commutation partielle f, il existe un morphisme marqué h et une fonction de commutation partitionnée g tels que $f = h\ g\ h^{-1}$.*

Cette proposition a permis de simplifier les preuves de plusieurs résultats portant sur les commutations partielles.

La proposition suivante exprime un résultat étonnant: les relations de semi-commutation, qui ne sont pas symétriques, peuvent également être obtenues à partir de relations de commutation partitionnée, donc de relations symétriques. Le résultat est beaucoup plus difficile à établir que dans le cas précédent. En outre la décomposition d'une fonction de semi-commutation ne s'exprime pas de façon aussi simple que celle d'une fonction de commutation partielle.

Proposition 7.2 [8]: *Toute fonction de semi-commutation peut être obtenue par compositions de morphismes, morphismes inverses et de fonctions de commutation partitionnée.*

Chapitre 2: Fonctions de semi-commutation et numérotation.

1. Image d'une fonction de semi-commutation par substitution alphabétique.

2. Intersection avec un rationnel.

3. Numérotation des différentes occurrences d'une même lettre.

1. Image d'une fonction de semi-commutation par substitution alphabétique.

Pour être en mesure de montrer certains résultats relatifs aux fonctions de semi-commutation et au produit de langages, M Clerbout et M. Latteux ont introduit la définition suivante.

Définition 1.1 [12]: *Soit* $S=\langle X,R\rangle$, *un système de semi-commutation et* $\tau: X^* \longrightarrow 2^{Y^*}$, *une substitution alphabétique. L'image de S par* τ *est le système* $S_\tau=\langle Y,R_\tau\rangle$ *où* $R_\tau=\{y_1y_2 \longrightarrow y_2y_1 \mid y_1,y_2 \in Y, y_1\neq y_2, \exists\, u \longrightarrow v \in R, y_1y_2 \in u\,\tau\}$.

Si f et f_τ sont les fonctions associées respectivement aux systèmes de semi-commutation S et S_τ, f_τ est alors l'image par τ de la fonction de semi-commutation f, et on peut établir les relations suivantes.

Lemme 1.2 [12]: *Soient f une fonction de semi-commutation définie sur un alphabet X et* $\tau: X^* \longrightarrow 2^{Y^*}$, *une substitution alphabétique, alors pour tout mot w de* X^*, $w f \tau \subset w \tau f_\tau$, *et pour tout mot w' de* Y^*, $w' \tau^{-1} f \subset w' f_\tau \tau^{-1}$.

Dans le cas particulier où τ est une projection, on retrouve ainsi un résultat montré par M.Clerbout.

Corollaire 1.3 [8]: *Soient f une fonction de semi-commutation définie sur un alphabet X, Y un sous-alphabet de X et w un mot de* X^*, *alors* $\forall\, w' \in w f$, $w' \Pi_Y \in (w \Pi_Y) f$.

Un cas particulier intéressant est celui où la substitution alphabétique est un morphisme inverse strictement alphabétique. On obtient alors le résultat suivant:

Lemme 1.4: [12] *Soient f une fonction de semi-commutation définie sur un alphabet X et* $g: Y^* \longrightarrow X^*$, *un morphisme strictement alphabétique. Soient* $\tau=g^{-1}$ *et* f_τ, *l'image par* τ *de f, alors,* $f\tau=\tau f_\tau$ *et* $f_\tau g=g f$.

En revanche, ces égalités ne sont pas toujours vérifiées si τ est un morphisme alphabétique, ou même un morphisme strictement alphabétique, comme le montre l'exemple suivant:

Exemple 1.5: Soient $X=\{a,b,c\}$ et $Y=\{a,b\}$. Soient f la fonction de semi-commutation associée au système $S=\langle X,\{ab \rightarrow ba\}\rangle$, $g:X^* \rightarrow Y^*$ défini par: $a\ g=a$ et $b\ g=b$, $c\ g=\varepsilon$ et $g':X^* \rightarrow Y^*$ défini par: $a\ g'=a$ et $b\ g'=c\ g'=b$. La fonction f_g et la fonction $f_{g'}$, images respectives de la fonction f par g et g', sont alors toutes deux égales à la fonction f' associée au système $S'=\langle Y,\{ab \rightarrow ba\}\rangle$. Considérons le mot $w=acb$. On obtient $w\ f\ g=\{ab\}$ alors que $w\ g\ f'=\{ab,ba\}$. De même, $w\ f\ g'=\{abb\}$ alors que $w\ g'\ f'=a \sqcup\!\sqcup bb$. Considérons maintenant le mot $w'=ab$. On a $w'\ f'\ g^{-1}=\{ab,ba\} \sqcup\!\sqcup c^*$ alors que $w'\ g^{-1}\ f = a\,b \sqcup\!\sqcup c^* + c^*bac^*$. Enfin, $w'\ f'\ g'^{-1}=\{ab,ba,ac,ca\}$ mais $w'\ g'^{-1}\ f=\{ab,ba,ac\}$.

Nous sommes donc amenés à introduire la définition suivante:

Définition 1.6: *Soit f une fonction de semi-commutation définie sur un alphabet X, une substitution alphabétique $\tau: X^* \rightarrow 2^{Y^*}$ sera dite compatible avec la fonction f si $f\ \tau=\tau\ f_\tau$, et $\tau^{-1}\ f=f_\tau\ \tau^{-1}$.*

Dans le chapitre 3, où nous étudierons les compositions de fonctions de commutation partitionnée, nous utiliserons le résultat suivant:

Lemme 1.7: *Toute projection définie sur un alphabet X est compatible avec toute fonction de commutation partitionnée définie sur X.*

Preuve: Soient f une fonction de commutation partitionnée associée à la partition $\{X_1,X_2,\ldots,X_k\}$ de l'alphabet X et $Y \subset X$. L'image de f par la projection Π_Y est alors la fonction de commutation associée à la partition $\{X_1 \cap Y, X_2 \cap Y,\ldots,X_k \cap Y\}$, c'est à dire la restriction de f à Y. Nous devons donc montrer $\Pi_Y\ f=f\ \Pi_Y$ et $f\ (\Pi_Y)^{-1}=(\Pi_Y)^{-1}\ f$. Soit $w \in X^*$, alors

$$
\begin{aligned}
(w\ f)\ \Pi_Y &= ((w\ \Pi_{X_1}) \sqcup\!\sqcup (w\ \Pi_{X_2}) \sqcup\!\sqcup \ldots \sqcup\!\sqcup (w\ \Pi_{X_k}))\ \Pi_Y \\
&= (((w\ \Pi_{X_1})\ \Pi_Y) \sqcup\!\sqcup ((w\ \Pi_{X_2})\ \Pi_Y) \sqcup\!\sqcup \ldots \sqcup\!\sqcup ((w\ \Pi_{X_k})\ \Pi_Y)) \\
&= (((w\ \Pi_Y)\ \Pi_{X_1}) \sqcup\!\sqcup ((w\ \Pi_{X_2})\ \Pi_Y) \sqcup\!\sqcup \ldots \sqcup\!\sqcup ((w\ \Pi_Y)\ \Pi_{X_k})) \\
&= w\ \Pi_Y\ f. \text{ De même, soit } w' \in (X-Y)^*, \text{ alors} \\
(w'\ f)\ (\Pi_Y)^{-1} &= (w'\ \Pi_{X_1}) \sqcup\!\sqcup (w'\ \Pi_{X_2}) \sqcup\!\sqcup \ldots \sqcup\!\sqcup (w'\ \Pi_{X_k}) \sqcup\!\sqcup (X-Y)^* \\
&= (w'\ \Pi_{X_1}) \sqcup\!\sqcup (w'\ \Pi_{X_2}) \sqcup\!\sqcup \ldots \sqcup\!\sqcup (w'\ \Pi_{X_k}) \sqcup\!\sqcup (X_1-Y)^* \sqcup\!\sqcup (X_2-Y)^* \sqcup\!\sqcup \ldots \sqcup\!\sqcup (X_k-Y)^* \\
&= (w'\ \Pi_{X_1} \sqcup\!\sqcup (X_1-Y)^*) \sqcup\!\sqcup (w'\ \Pi_{X_2} \sqcup\!\sqcup (X_2-Y)^*) \sqcup\!\sqcup \ldots \sqcup\!\sqcup (w'\ \Pi_{X_k} \sqcup\!\sqcup (X_k-Y)^*) \\
&= ((w \sqcup\!\sqcup (X-Y)^*)\ \Pi_{X_1}) \sqcup\!\sqcup ((w \sqcup\!\sqcup (X-Y)^*)\ \Pi_{X_2}) \sqcup\!\sqcup \ldots \sqcup\!\sqcup ((w \sqcup\!\sqcup (X-Y)^*)\ \Pi_{X_k}) \\
&= w'\ (\Pi_Y)^{-1}\ f.
\end{aligned}
$$

□

Remarque: Ce résultat devient faux dans le cas où f est une fonction de commutation partielle car la relation de non commutation associée n'est pas toujours transitive. Nous pouvons le constater dans l'exemple suivant: Soient $X=\{a,b,c\}$, et f la fonction de commutation partielle associée à la relation $C=\{(a,b)\}$. Considérons le mot $w=acb$, comme c ne commute ni avec a, ni avec b, l'image de w par f ne contient que w. On a alors $w f \Pi_{\{a,b\}}=\{ab\}$, alors que $w \Pi_{\{a,b\}} f=\{ab,ba\}$.

2. Intersection avec un rationnel.

Un résultat analogue concernant une intersection avec un rationnel ne peut être obtenu que pour des rationnels vérifiant certaines propriétés:

Lemme 2.1: *Soient f une fonction de semi-commutation définie sur un alphabet X, R un rationnel inclus dans X^*, clos par f, et K un rationnel inclus dans X^*, clos par f et par f^{-1}, alors les deux relations suivantes sont vérifiées:*

$-\forall u \in X^*, u(\cap R)f \subset uf(\cap R)$.

$-(\cap K)f=f(\cap K)$.

En particulier, si f est une fonction de commutation partielle, $f=f^{-1}$, et $(\cap R)f=f(\cap R)$.

Preuve: Si $u \in R$, alors $u(\cap R)f=uf=uf(\cap R)$ puisque R est fermé par f, sinon $u(\cap R)f=\varnothing$. Soit, maintenant $w \in uf(\cap K)$, puisque K est fermé par f^{-1}, $u \in K$ et $w \in u(\cap K)f$. On en déduit $(\cap K)f=f(\cap K)$.

□

3. Numérotation des différentes occurrences d'une même lettre.

Il est souvent utile, pour étudier certains phénomènes liés aux fonctions de semi-commutation, de considérer des mots ne contenant qu'une occurrence des lettres qui le composent. Afin de différencier plusieurs occurrences d'une même lettre dans un mot, nous introduisons la définition suivante:

Soient C une relation de semi-commutation définie sur un alphabet X, et f_C la fonction de semi-commutation associée. A tout entier strictement positif k, on associe:

l'alphabet $X_k = X \times [1,k]$,

la relation $C_k = \{ ((a,i),(b,j)) \in X_k \times X_k / (a,b) \in C \}$,

l'application: $num_k : X^* \longrightarrow X_k^*$ défini par: $\varepsilon\, num_k = \varepsilon$, et $\forall u \in X^*$, $\forall x \in X$, $(u\,x)\, num_k = (u\ num_k)\,(x,p)$, où $p = inf(k, |u x|_x)$. Il s'agit d'une application séquentielle ou gsm (voir [4]).

Nous pouvons alors énoncer:

Proposition 3.1: *Soient f_C une fonction de semi-commutation définie sur un alphabet X et u,v deux mots de X^*, alors $v \in u f_C$ si et seulement si $v\ num_k \in (u\ num_k)\, f_{C_k}$.*

Preuve: Il suffit de montrer $f_C\ num_k = num_k\, f_{C_k}$. Soient $h_k : X_k^* \longrightarrow X^*$, le morphisme strictement alphabétique défini par $\forall (x,i) \in X_k$, $(x,i)\, h_k = x$, et le rationnel $R_k = FG\left(\coprod_{x \in X} (x,1)\,(x,2) \ldots (x,k-1)(x,k)^+ \right)$.

Clairement, $num_k = h_k^{-1}(\cap R_k)$. De plus, par définition de l'image d'une fonction de semi-commutation par substitution alphabétique, f_{C_k} est l'image par h_k^{-1} de la fonction f_C. Du *lemme 1.3* on déduit donc: $f_C\ num_k = f_C\ h_k^{-1}(\cap R_k) = h_k^{-1} f_{C_k}(\cap R_k)$. Maintenant, comme R_k est fermé par f_{C_k} et $f_{C_k}^{-1}$, on déduit du *lemme 2.1*:

$f_C\ num_k = h_k^{-1}(\cap R_k)\, f_{C_k} = num_k\, f_{C_k}$

□

Remarque: Pour un mot u donné, on prendra en général $k \geq |u|$, ce qui assurera que $u\ num_k$ ne contient pas deux occurrences de la même lettre. Aussi, pour alléger certaines démonstrations, on pourra supposer qu'on fait agir des fonctions de semi-commutation sur des mots ne contenant qu'une seule occurrence de chaque lettre, sans utiliser formellement le mécanisme de numérotation.

Définition 3.2: *Soient f une fonction de semi-commutation définie sur un alphabet X, $k \in \mathbb{N}_+$ et $X_k = X \times [1,k]$. Pour tout Y inclus dans X_k, nous appellerons* ***projection sélective*** *d'un mot u de X^* selon Y, le mot $u\, PS_Y = (u\ num_k)\ \Pi_Y\ h_k$.*

De même que pour une projection classique, nous noterons, pour un mot w de X_k^*, $PS_w = PS_{alph(w)}$.

En appliquant le *lemme 1.2* aux projection sélectives, on obtient le résultat suivant, qui correspont en quelque sorte à une extension du *corollaire 1.3*:

Lemme 3.3: *Soient f une fonction de semi-commutation définie sur un alphabet X, $k \in \mathbb{N}_+$, $X_k = X \times [1,k]$ et $Y \subset X_k$. Alors, pour tous mots u,v de X^*, on a $(v \in u f) \Rightarrow (v\, PS_Y \in (u\, PS_Y) f)$.*

Dans le cas où Y ne contient qu'une seule lettre de X_k, on retrouve ainsi un résultat montré par Y. Métivier.

Lemme 3.4 [26]: *Soit f une fonction de semi-commutation définie sur un alphabet X. Pour tous mots u_1,u_2,v_1,v_2 de X^* et toute lettre x de X, on a: $(v_1 x v_2 \in u_1 x u_2 f$ et $|v_1|_x = |u_1|_x) \Rightarrow (v_1 v_2 \in u_1 u_2 f)$.*

Si f est une fonction de commutation partitionnée, on obtient, en utilisant la *proposition 3.1* et le *lemme 1.7*, le résultat suivant:

Lemme 3.5: *Soient f une fonction de commutation partitionnée définie sur un alphabet X, $k \in \mathbb{N}_+$, et $Y \subset X \times [1,k]$, alors $f PS_Y = PS_Y f$.*

De même, la numérotation des différentes occurrences d'une lettre et les projections sélectives permettent de montrer le lemme suivant:

Lemme 3.6 [26]: *Soient f une fonction de semi-commutation définie sur un alphabet X, et C la relation de semi-commutation associée à la fonction f. Pour tout u,u',v,v' dans X^*, $u'v' \in uvf$ si et seulement si il existe des mots u_1,u_2,v_1,v_2 vérifiant: $u_1 u_2 \in uf$, $v_1 v_2 \in vf$, $u' \in u_1 v_1 f$, $v' \in u_2 v_2 f$, et $\forall (x,y) \in alph(u_2) \times alph(v_1)$, $(x,y) \in C$.*

Preuve: La condition est clairement suffisante. Réciproquement, on sait d'après le *lemme 1.7* qu'on peut considérer que uv est formé de lettres toutes différentes, si tel n'était pas le cas, il suffirait de numéroter les différentess occurrences des mêmes lettres apparaissant dans uv. Posons $u_1 = u' \Pi_{alph(u)}$, $u_2 = v' \Pi_{alph(u)}$, $v_1 = u' \Pi_{alph(v)}$, et $v_2 = v' \Pi_{alph(v)}$. Comme $u'v' \in uvf$, on déduit du *lemme 1.7* que $u_1 u_2 = (u'v' \Pi_{alph(u)}) \in uf$. De même, $v_1 v_2 = (u'v' \Pi_{alph(v)}) \in vf$. Considérons maintenant a,b deux lettres différentes de $alph(u_1 v_1) = alph(u')$, alors si $\{a,b\} \subset alph(u_1)$ ou $\{a,b\} \subset alph(v_1)$, $u_1 v_1 \Pi_{ab} = u'v' \Pi_{ab}$, sinon $u_1 v_1 \Pi_{ab} = uv \Pi_{ab}$ par construction de u_1 et de v_1. De plus $u_1 v_1\, com = u'\, com$. On

déduit alors: $\forall\ a,b \in alph(u'),\ u'\Pi_{ab} \in (u_1v_1\Pi_{ab})f$, d'où $u' \in u_1v_1 f$, d'après le *lemme 3.5* du chapitre 1. De la même façon, on démontre $v' \in u_2v_2 f$. Enfin, considérons $a \in alph(u_2)$ et $b \in alph(v_1)$, alors $uv\,\Pi_{ab}=ab$ et $u'v'\,\Pi_{ab}=ba$, on déduit donc, d'après le *lemme 1.7*, que $ba \in abf$, et, comme a est différent de b, $(a,b) \in C$, où C est la relation de commutation associée à la fonction f.

□

Nous utiliserons, dans le chapitre 4, la conséquence suivante de ce lemme, qui correspont au cas où $v'=\varepsilon$.

Corollaire 3.7: *Soit f une fonction de semi-commutation définie sur un alphabet X. Pour tous mots w,u_1,u_2,v_1,v_2 de X^* vérifiant $u_1u_2 \in wf$, $v_1v_2 \in wf$ et $u_1\ com=v_1\ com$, il existe des mots $w_1,w_2 \in X^*$ tels que $u_1 \in w_1 f$, $v_1 \in w_1 f$, $u_2 \in w_2 f$, $v_2 \in w_2 f$ et $w_1w_2 \in wf$.*

Preuve: On utilise la preuve précédente, appliquée à la fonction de semi-commutation f^{-1}, et de la même façon on considère que les lettres apparaissant dans w sont toutes différentes. On déduit donc que si $w \in u_1v_1 f^{-1}$, il existe des mots $w_1=w\,\Pi_{alph(u_1)}$ et $w_2=w\,\Pi_{alph(u_2)}$, vérifiant $w_1 \in u_1 f^{-1}$, $w_2 \in u_2 f^{-1}$, et $w \in w_1w_2 f^{-1}$. De même, il existe des mots $w_3=w\,\Pi_{alph(v_1)}$ et $w_4=w\,\Pi_{alph(v_2)}$, vérifiant $w_3 \in v_1 f^{-1}$, $w_4 \in v_2 f^{-1}$, et $w \in w_3w_4 f^{-1}$. Or $alph(u_1)=alph(v_1)$ et $alph(u_2)=alph(v_2)$, puisque $u_1\ com=v_1\ com$. Donc $w_1=w_3$ et $w_2=w_4$, ce qui termine la preuve du *corollaire 3.7*.

□

Chapitre 3: Compositions de fonctions de commutation partitionnée.

1. Définitions, notations, rappels.

Dans cette partie, nous utiliserons les notations définies dans [8] et [9]. En particulier, nous noterons:

P^k: l'ensemble des compositions de k fonctions de commutation partitionnée, où k est un élément de $\mathbb{N}$. En particulier, P^0 dénotera l'ensemble réduit à la fonction "identité" qui à tout mot u associe le singleton $\{u\}$.

$P^* = \bigcup_{k \in \mathbb{N}} P^k$: l'ensemble des compositions de fonctions de commutation partitionnée.

P_X: l'ensemble des fonctions de commutation partitionnée définies sur l'alphabet X.

P_X^*: l'ensemble des compositions de fonctions de commutation partitionnée définies sur l'alphabet X.

Ψ_X: l'ensemble des partitions de l'alphabet X.

Ψ_X^*: le monoïde libre engendré par l'alphabet Ψ_X: à $u=\psi_1\psi_2...\psi_k$ de Ψ_X^* correspond l'élément de P_X^*: $f_u = f_{\psi_1} f_{\psi_2} \cdots f_{\psi_k}$.

Pour alléger les notations, si $\psi=\{X_1,X_2,...,X_k\}$ est un élément de Ψ_X, nous noterons indifféremment $\psi=(X_1,X_2,...,X_k)$ ou $\psi=(w_1,w_2,...,w_k)$ où $\forall\ i \in [1,k]$, $alph(w_i)=X_i$ et $\forall\ x \in X_i$, $|w_i|_x=1$. De même, $f_{(w_1,w_2,...,w_k)}$ dénotera l'élément de P_X, associé à la partition ψ.

Exemple 1.1: Soit $X=\{a,b,c,d\}$: à $(ab,cd)(ac,bd)$ de Ψ_X^* correspond $g=f_{(ab,cd)}f_{(ac,bd)}$. Soient $u=abdc$, $v=cdba$ et $v'=cabd$, alors $v \in u\,g$: en effet, il existe un mot $w=dcab$ vérifiant $w \in u\,f_{(ab,cd)}$ et $v \in w\,f_{(ac,bd)}$. En revanche, $v' \notin u\,g$. Si tel était le cas, il existerait un mot w' vérifiant $w' \in u\,f_{(ab,cd)}$ et $v' \in w'\,f_{(ac,bd)}$. D'après le *lemme 6.1* du chapitre 1, les égalités suivantes seraient vérifiées:

$$u\ \Pi_{ab} = w'\ \Pi_{ab} = ab,$$
$$u\ \Pi_{cd} = w'\ \Pi_{cd} = dc,$$
$$w'\ \Pi_{ac} = v'\ \Pi_{ac} = ca,$$
$$w'\ \Pi_{bd} = v'\ \Pi_{bd} = bd.$$

Et ces relations sont clairement incompatibles.

On définit sur Ψ_X^* la relation d'ordre partiel suivante:

Définition 1.2: *Pour tous mots u et v de Ψ_X^*, on a $u \leq v$ si et seulement si $\forall w \in X^*$, $w f_u \subset w f_v$.*

La relation $\leq$ confère à Ψ_X une structure de treillis . Si $\psi=\{X_1,X_2,\ldots,X_p\}$ et $\varphi=\{Y_1,Y_2,\ldots,Y_q\}$ sont deux éléments de Ψ_X, nous noterons $\psi \vee \varphi$ le plus petit majorant de $\{\psi,\varphi\}$, correspondant à la partition définie par l'ensemble $\{X_i \cap Y_j / i \in [1,p], j \in [1,q], X_i \cap Y_j \neq \varnothing\}$, et $\psi \wedge \varphi$ le plus grand minorant de $\{\psi,\varphi\}$, correspondant à la partition définie par l'ensemble $\{X_i \cup Y_j / i \in [1,p], j \in [1,q], X_i \cap Y_j \neq \varnothing\}$. Le plus petit élément de Ψ_X , noté 0, est associé à la relation où rien ne commute: c'est la partition $\{X\}$. Le plus grand élément, noté 1, est associé à la commutation totale: c'est la partition $\{\{x\}, x \in X\}$. D'autre part, la relation $\leq$ est compatible avec le produit dans Ψ_X^*, on en déduit une congruence sur Ψ_X^*:

Définition 1.3: *Pour tous mots u et v de Ψ_X^*, $u \equiv v$ si et seulement si $u \leq v$ et $v \leq u$ c'est à dire, $f_u = f_v$.*

Exemple 1.4: Soit $X=\{a,b,c,d\}$. Dans ce cas, $1=(a,b,c,d)$ et $0=(abcd)$. Clairement, $(ab,cd) \leq (a,b,cd)$, donc $(ab,cd) \vee (a,b,cd) = (a,b,cd)$ et $(ab,cd)(a,b,cd) \equiv (a,b,cd)$. En revanche, $(ab,cd)(ac,bd)$ n'est pas congru à un élément de Ψ_X. En effet, comme chaque lettre peut commuter au moins une fois avec toute autre lettre, on aurait $(ab,cd)(ac,bd) \equiv 1$, ce qui n'est pas le cas, comme le montre l'exemple 1.1.

M. Clerbout a donné dans [9] un certain nombre de règles de simplification et de réduction des mots de Ψ_X^*, en particulier:

Règle 1.5: $\forall \psi,\varphi \in \Psi_X,\ (\psi \leq \varphi) \Rightarrow (\psi\varphi \equiv \varphi\psi \equiv \varphi)$

Règle 1.6: $\forall \psi \in \Psi_X, \forall u,v \in \Psi_X^*,\ \psi u \psi v \psi \equiv \psi u v \psi$

Cette dernière règle d'effacement montre que tout mot de Ψ_X^* est toujours équivalent à un mot qui ne contient pas plus de deux occurrences de la même lettre, d'où le résultat suivant:

Proposition 1.7: $\Psi_X^*/\equiv$ *est fini.*

Les deux règles suivantes sont également très utiles:

Règle 1.8: *Soient* $\psi,\psi',\varphi_1,\varphi_2,\ldots,\varphi_k$ *des lettres de* Ψ_X. *Soit* $\psi''=\psi \wedge \psi'$, *alors* $\psi\varphi_1\varphi_2\ldots\varphi_k\psi' \equiv \psi(\varphi_1 \vee \psi'')(\varphi_2 \vee \psi'')\ldots(\varphi_k \vee \psi'')\psi'$.

En particulier, si $\psi=\psi'$, on a $\psi''=\psi$, $\psi \leq (\varphi_1 \vee \psi)$ et $\psi \leq (\varphi_k \vee \psi)$, donc: $\psi\varphi_1\varphi_2\ldots\varphi_k\psi \equiv (\varphi_1 \vee \psi)(\varphi_2 \vee \psi)\ldots(\varphi_k \vee \psi)$.

Règle 1.9: *Soient* $\psi=(Z,Z_1,Z_2,\ldots,Z_p)$ *et* $\varphi=(Z \cup Y,Y_1,Y_2,\ldots,Y_q)$ *deux lettres de* Ψ_X, *alors:* $\psi\varphi \equiv \psi\varphi'$ *et* $\varphi\psi \equiv \varphi'\psi$, *où* $\varphi'=(Z,Y,Y_1,Y_2,\ldots,Y_q)$.

Grâce à ces règles de simplifications, M. Clerbout a étudié complètement le cas d'un alphabet de quatre lettres, et la proposition suivante a permis d'établir que tout mot de Ψ_X^*, où $X=\{a,b,c,d\}$, se réduit toujours, et de manière unique, à un mot de longueur inférieure ou égale à 2.

Proposition 1.10 [9]: *La fonction de commutation* $f(ab,cd)\,f(ac,bd)\,f(ad,bc)$ *réalise la commutation totale sur l'alphabet* $X=\{a,b,c,d\}$, *ce qui peut encore s'exprimer par:*$(ab,cd)(ac,bd)(ad,bc) \equiv (a,b,c,d)=1$.

Pour terminer, rappelons que, si on ne fixe pas la taille de l'alphabet, il n'est pas toujours possible de réduire la composée de k fonctions de commutation partitionnée. Ceci se déduit de résultat plus général suivant:

Proposition 1.11 [8]: *Si* $RAT\,P^k=\{ R\,g \mid R \in RAT,\ g \in P^k\}$, *alors* $\forall\ k \geq 0$, $RAT\,P^k \subsetneq RAT\,P^{k+1} \subsetneq (RAT\,P)\,H^{-1}\,H_{s\alpha}$, *où* $(RAT\,P)\,H^{-1}\,H_{s\alpha}=\{ L\,g^{-1}\,h \mid L \in RAT\,P,\ g$ *est un morphisme,* h *est un morphisme strictement alphabétique}*.

2. Propriétés conservées et problèmes de caractérisation.

Un certain nombre de propriétés vérifiées par les fonctions de semi-commutation le sont également par les compositions de fonctions de commutation partitionnée. Si X est un alphabet et $w=\psi_1\psi_2...\psi_n$ est un mot de Ψ_X^*, on peut montrer facilement les résultats suivants, par simple induction sur n et en utilisant les lemmes de la première partie. Ainsi, le lemme suivant est-il une conséquence directe de la *proposition 3.1* du chapitre 2.

Lemme 2.1: *Soient $k \in \mathbb{N}_+$ et u,v deux mots de X^*, alors $v \in u f_w$ si et seulement si, $v\ num_k \in (u\ num_k)\ g$, où $g=g_1\ g_2...g_n$ avec $\forall\ i \in [1,n]$, g_i est l'image par num_k de f_{ψ_i}.*

Du *lemme 3.5* du chapitre 2, on déduit de la même façon:

Lemme 2.2: *Soient $k \in \mathbb{N}_+$ et $Y \subset Xx[1,k]$, alors $f_w\ PS_Y = PS_Y\ f_w$.*

En revanche, l'exemple 1.1 nous montre que le lemme de projection ne s'applique plus au cas des compositions de fonctions de commutation. En effet, on peut vérifier dans l'exemple 1.1 que: $\forall\ (x,y) \in XxX$, $v' \Pi_{xy} \in (u \Pi_{xy})\ g$ alors que $v' \notin u\ g$. La caractérisation des couples de mots (u,v) vérifiant pour une composition de fonctions de commutation partitionnée donnée f: $v \in u f$ sera donc plus compliquée que dans le cas d'une seule fonction de commutation. Si $f=f_1\ f_2...f_k$, il s'agit en effet de trouver une suite de mots $w_0,w_1,...,w_k$ vérifiant: $w_0=u$, $w_k=v$ et $\forall\ i \in [1,k]$, $w_i \in w_{i-1} f_i$. Il n'y a, bien sûr, qu'un nombre fini de mots à tester, mais la recherche est, en général, longue et coûteuse. Aussi se pose la question de savoir si il est possible d'améliorer cet algorithme de décision.

Une réponse à cette question est donnée dans les sections suivantes pour les compositions de deux fonctions de commutation partielle et, dans le cas d'un alphabet de cinq lettres, pour la composée de trois fonctions de commutation partitionnée.

3. Composition de deux fonctions de commutation partielle.

L'étude complète de la composition de deux fonctions de commutation partitionnée a été faite dans [9], mais ce qui nous préoccupe dans cette section est la possibilité de décider simplement, étant données f_1 et f_2 deux fonctions de commutation partielle et u,v deux mots commutativement équivalents de X^*, si v est dans l'image de u par la composée $f_1 f_2$.

Une première méthode consiste évidemment à calculer effectivement l'image de u par f_1, puis l'image de chacun des mots obtenus par f_2 et de vérifier si le mot v est atteint. Cette méthode est très couteuse quant au nombre de mots à construire. Une deuxième méthode, plus avantageuse serait de calculer les ensembles $u f_1$ d'une part, $v f_2$ d'autre part et de vérifier ensuite si leur intersection est non vide. Nous proposons ici une troisième méthode, basée sur une propriété montrée dans [20], concernant les familles reconstructibles. Avant d'énoncer cette propriété, nous rappelons le définition du graphe commutation ou de non-commutation d'une fonction de semi-commutation.

Définition 3.1: *Soit f une fonction de semi-commutation définie sur un alphabet X, associée à la relation C. Le graphe de commutation(resp. de non-commutation) de la fonction f est le graphe noté (X,f) (resp. $(X,\overline{f})$), où les sommets sont les lettres de X et les arcs sont les couples de lettres distinctes (x,y) qui ne sont pas dans C (resp. les couples de lettres qui sont dans C).*

Dans le cas des fonctions de commutation partielle, le graphe de non-commutation est alors un graphe non orienté. C. Duboc a montré la caractérisation suivante:

Proposition 3.2: [20] *Soit f une fonction de commutation partielle définie sur un alphabet X et $\{X_1,X_2,\ldots,X_n\}$ une famille de cliques recouvrant le graphe de non-commutation de la fonction f. Alors, pour tous mots u,v de X^*, on a: $v \in uf$ si et seulement si $\forall i \in [1,n]$, $u \Pi_{X_i}=v \Pi_{X_i}$.*

Pour un recouvrement réalisé avec des cliques de cardinalité au plus égale à deux, on retrouve ainsi le *lemme 5.1* du premier chapitre.

En relation avec la *proposition 3.2*, la notion de famille reconstructible a été introduite par R. Cori et Y. Métivier.

Définition 3.3: [17] *Soient* $F=\{X_1,X_2,..,X_n\}$ *une famille de sous-alphabets d'un alphabet* X *et* $W=(w_1,w_2,...,w_n)$ *une famille de mots où* $\forall\, i \in [1,n]$, $w_i \in X_i^*$.

- *W est une famille* ***reconstructible*** *si il existe un mot w de X^* tel que:* $\forall\, i \in [1,n]$, $w \Pi_{X_i}=w_i$. *Le mot w est alors appelé un* ***reconstruit de la famille*** *W.*
- *W est une famille* ***quasi-reconstructible*** *si:* $\forall\, i,j$ $w_i \Pi_{X_j}=w_j \Pi_{X_i}$.
- *W est une famille* ***faiblement quasi-reconstructible*** *si:* $\forall\ i,j$, $\forall\, x \in X_i \cap X_j$, $/w_i/_x=/w_j/_x$.

Les fonctions de commutation partielle et les familles reconstructibles sont liées par l'intermédiaire de la proposition suivante, montrée dans [20], mais que nous énonçons sous une forme différente:

Lemme 3.4: [20] *Soient f une fonction de commutation partielle définie sur un alphabet X, $F=\{X_1,X_2,...,X_n\}$ un recouvrement par cliques du graphe de non-commutation de f et u un mot de X^*, alors l'ensemble des images par f du mot u est l'ensemble des reconstruits de la famille reconstructible:* $(u \Pi_{X_1},u \Pi_{X_2},...,u \Pi_{X_n})$.

Pour permettre une représentation commode de l'ensemble des images d'un mot par une fonction de commutation partielle, C. Duboc a introduit la notion de **graphe des occurrences** d'une famille (faiblement) quasi-reconstructible. Cette définition peut être donnée également en utilisant la numérotation des différentes occurrences d'une même lettre dans un mot.

Définition 3.5: *Soit* $W=(w_1,...,w_n)$ *une famille (faiblement) quasi-reconstructible. Le graphe des occurrences de la famille W est le graphe (S,A) où* $S=\bigcup_{i\in[1,n]} alph(num_k(w_i))$ *avec* $k=sup\{/w_i/,\ i \in [1,n]\}$ *et il existe un arc de (x,p) vers (y,q) si il existe un mot w_r dans W tel que (x,p) précède (y,q) dans* $num_k(w_r)$.

La caractérisation suivante est alors fort utile:

Lemme 3.6: [20] *Une famille (faiblement) quasi-reconstructible est reconstructible si et seulement si son graphe des occurrences est sans cycle.*

Dans la suite, nous utiliserons le graphe des occurrences de l'image d'un mot u par une fonction de commutation partielle f , noté $O(u\,f)$, défini comme étant le graphe des occurrences de la famille reconstructible engendré par u et le recouvrement par les cliques maximales du graphe de non-commutation de f. Nous utiliserons également le graphe des occurences d'un mot u défini comme étant le graphe des occurrences de la famille reconstructible contenant u comme seul élément. Le *lemme 3.4* nous permet alors d'énoncer:

Lemme 3.7: *Soient f une fonction de commutation partielle définie sur un alphabet X et u,v deux mots de X^*, alors v est dans uf si et seulement si $O(u\,f)=O(v\,f)$.*

Reprenons maintenant notre problème initial: étant données deux fonctions de commutation partielle f_1 et f_2 définies sur un alphabet X et deux mots u,v de X^*commutativement équivalents, il s'agit de vérifier si $v \in uf_1f_2$. Avec ce qui précède, nous pouvons énoncer:

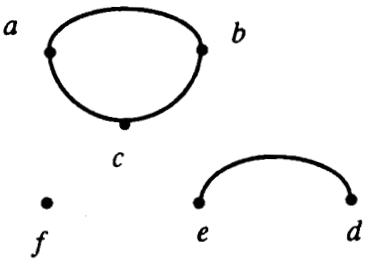
lemme 3.8: *Soient f_1 et f_2 deux fonctions de commutation partielle définies sur un alphabet X et u,v deux mots commutativement équivalents de X^*, alors: $v \in uf_1f_2$ si et seulement si la superposition des graphes des occurrences $O(uf_1)$ et $O(vf_2)$ est sans cycle.*

Preuve: $v \in uf_1f_2$ si et seulement si il existe un mot w vérifiant $w \in uf_1$ et $w \in vf_2$. Soient $\{X_1,X_2,\ldots,X_p\}$ et $\{X'_1,X'_2,\ldots,X'_q\}$ les recouvrements par cliques maximales des graphes de non-commutation respectifs des fonctions f_1 et f_2, D'après la *proposition 3.2* on déduit $v \in uf_1f_2$ si et seulement si il existe un mot w de X^* vérifiant $\forall\, i \in [1,p]$, $w\,\Pi_{X_i}=u\,\Pi_{X_i}$ et $\forall\, i \in [1,q]$, $w\,\Pi_{X'_i}=v\,\Pi_{X'_i}$. Ce qui peut encore s'exprimer: $v \in uf_1f_2$ si et seulement si la famille faiblement quasi-reconstructible $(u\,\Pi_{X_1},u\,\Pi_{X_2},\ldots,u\,\Pi_{X_p},v\,\Pi_{X'_1},v\,\Pi_{X'_2},\ldots,v\,\Pi_{X'_q})$ est reconstructible. On obtient alors le résultat voulu grâce au *lemme 3.6*. Dans le cas où la superposition des graphes des occurrences respectifs de uf_1 et vf_2 est sans cycle, celle-ci détermine alors un ordre strict partiel sur l'ensemble de ses sommets et tout ordre strict total compatible avec cet ordre partiel permet d'obtenir un mot w vérifiant $w \in uf_1$ et $w \in vf_2$.

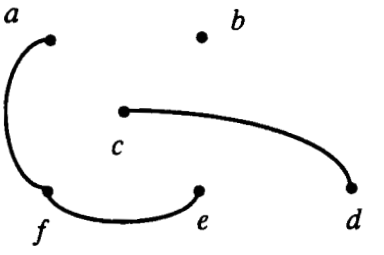
□

Nous allons appliquer ce résultat sur un exemple.

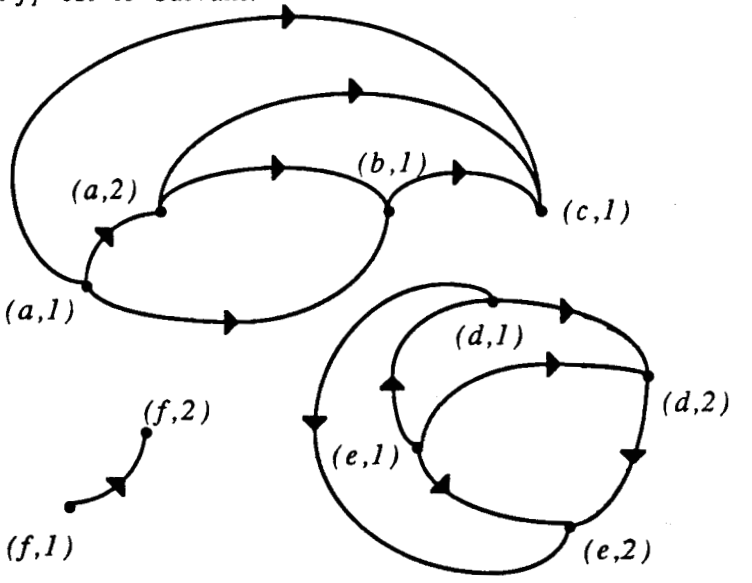
Exemple 3.9: Soit l'alphabet $X=\{a,b,c,d,e,f\}$. Soient f_1 la fonction de commutation partielle définie sur X, donnée par le graphe de non-commutation:



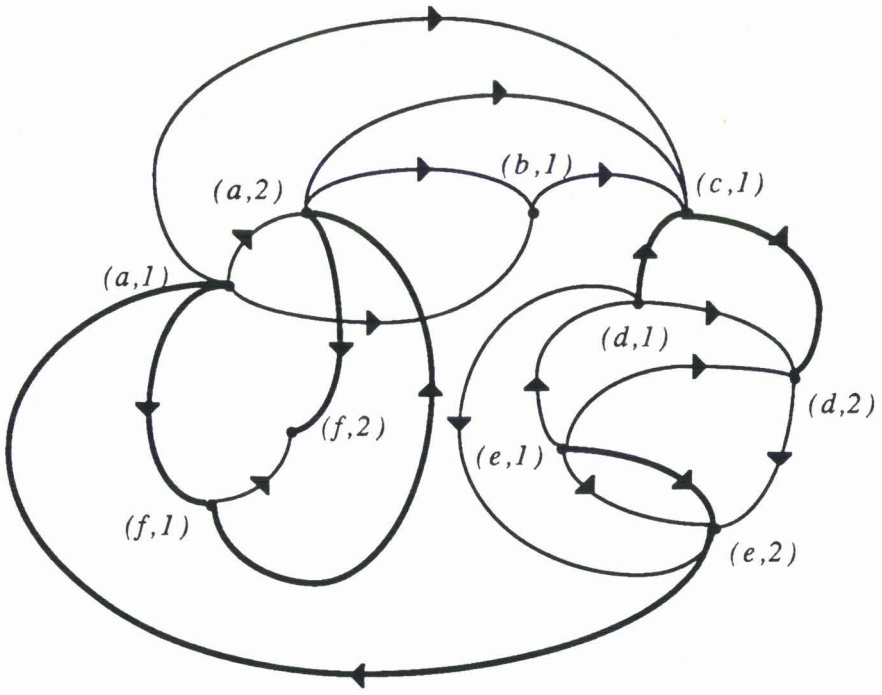
et f_2 la fonction de commutation partielle donnée par le graphe de non-commutation:



Considérons les mots $u=aefabdcdfe$ et $v=eedafcbadf$. Le graphe des occurrences de $u\,f_1$ est le suivant:



La superposition du graphe des occurrences de $u\,f_1$ avec le graphe de couvereture de celui de $v\,f_2$ nous donne alors le graphe suivant:



Ce graphe contient le cycle $(a,2)(c,1)(d,2)(e,2)(f,1)(a,2)$ donc v n'est pas dans l'image de u par $f_1 f_2$.

4. Compositions de trois fonctions de commutation partielle.

Comme nous venons de le voir, il est facile et rapide de décider si un mot v est dans l'image d'un mot u par la composée de deux fonctions de commutation partielle. Le problème se complique singulièrement dès que l'on passe à la composée de trois fonctions de commutation partielle. Nous pouvons cependant énoncer:

Lemme 4.1: *Soient f_1, f_2, et f_3 trois fonctions de commutation partielle définies sur un alphabet X et u,v deux mots de X^*. Soient $G_u=(S,A_u)$ le graphe des occurrences de u f_1 et $G_v=(S,A_v)$ le graphe des occurrences de v f_3, alors v est dans $u f_1 f_2 f_3$ si et seulement si il existe un graphe $G=(S,A)$ vérifiant les deux propriétés suivantes:*

(1) Pour tous éléments de S $(x,i),(y,j)$, si x et y ne commutent pas par f_2, alors il existe dans G soit un arc de (x,i) vers (y,j), soit un arc de (y,j) vers (x,i).

(2) Ni la superposition des graphes G et G_u, ni la superposition des graphes G et G_v ne contient de cycle.

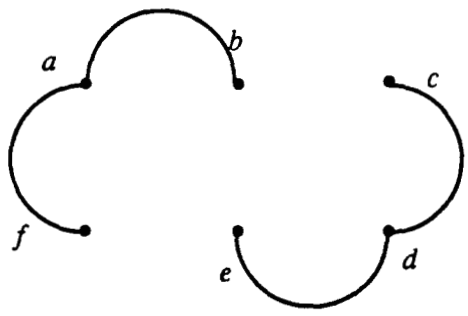
Preuve: Le mot v est dans $u f_1 f_2 f_3$ si et seulement si il existe deux mots de X^*: w et w' tels que: $w \in u f_1$, $w' \in v f_3$ et $w' \in w f_2$. Supposons qu'il existe un graphe G vérifiant les propriétés *(1)* et *(2)*. On peut alors trouver deux graphes complets sans cycle G_w et $G_{w'}$ ayant S comme ensemble des sommets et admettant respectivement comme sous-graphe la superposition des graphes G et G_u d'une part, la superposition des graphes G et G_v d'autre part. Les graphes G_w et $G_{w'}$ déterminent alors deux mots w et w' qui vérifient clairement: $w \in u f_1$, $w' \in v f_3$ et $w' \in w f_2$.

Réciproquement, si il existe deux mots de X^* w et w' tels que $w \in u f_1$, $w' \in v f_3$ et $w' \in w f_2$, alors le graphe $G = O(w f_2) = O(w' f_2)$ vérifie la propriété *(1)* par définition des graphes des occurrences. De plus $O(w f_1) = O(u f_1)$ et $O(w' f_3) = O(v f_3)$, le graphe G vérifie donc la propriété *(2)*.

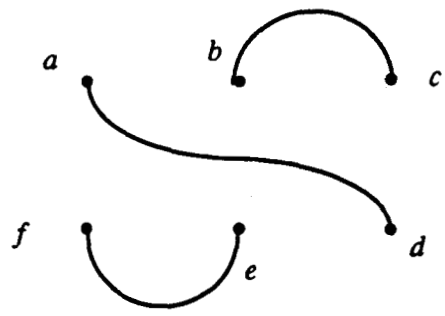
□

Les propriétés *(1)* et *(2)* sont cependant difficiles à établir comme le montrent les exemples suivants.

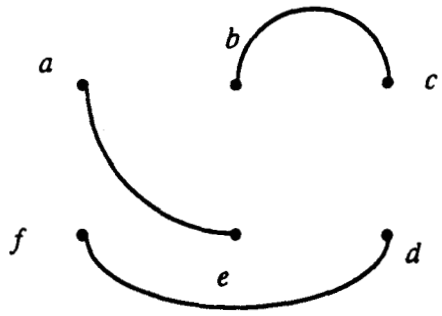
Exemple 4.2: Soient l'alphabet $x = \{a,b,c,d,e,f\}$ et $f_1 f_2 f_3 f_4$ les quatre fonctions de commutation partielle définies sur X par leur graphe de non-commutation donnés ci-dessous.



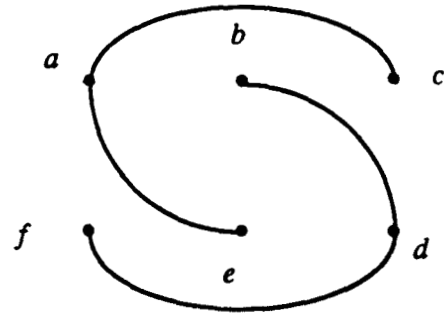
f_1



f_2

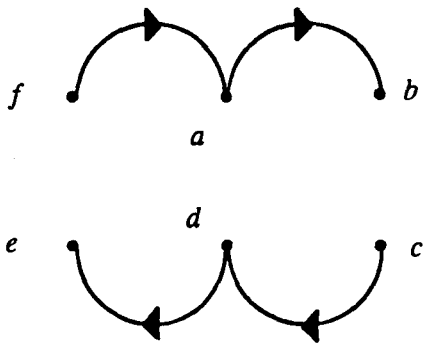


f_3

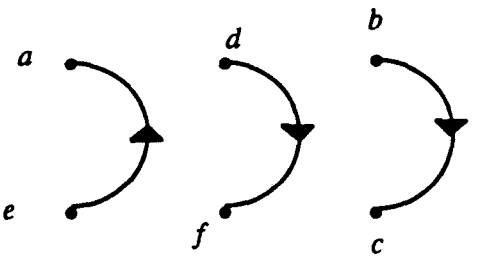


f_4

Soient $u=fabcde, v=eabdcf$ et $v'=eadbcf$. Nous allons tout d'abord vérifier si v est dans l'image de u par $f_1 f_2 f_3$. La construction des graphes des occurrences $O(u\ f_1)$ et $O(v\ f_3)$ nous donnent les graphes suivant, où par souci de clarté et comme aucune lettre de X n'apparait plusieurs fois dans u,v et v', nous n'avons pas fait figurer le numéro des lettres dans les noms des sommets.

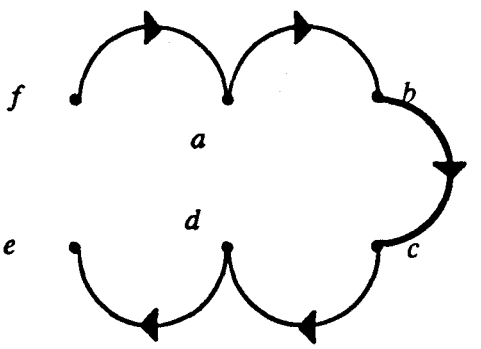


$O(u\ f_1)$



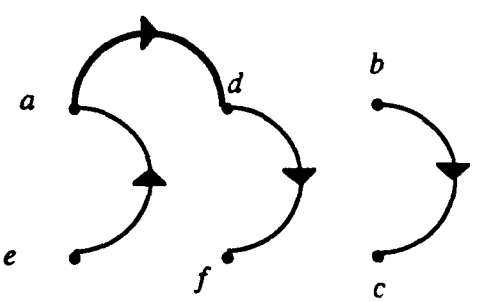
$O(v\ f_3)$

Comme les lettres b et c ne commutent pas par f_2, et qu'il existe un arc de b vers c dans $O(v\ f_3)$, on déduit que si il existe un graphe vérifiant les propriétés du *lemme 4.1*, ce graphe contient nécessairement un arc de b vers c. Par superposition avec $O(u\ f_1)$, on obtient alors le graphe suivant:



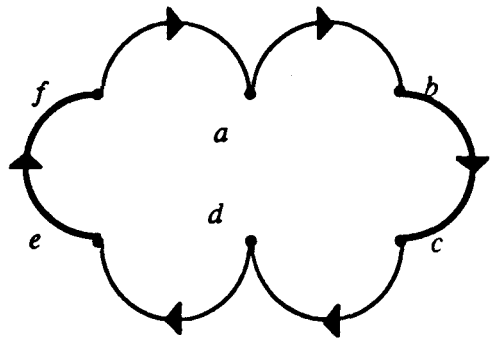
G_1

Les lettres a et d ne commutent pas par f_2, comme il existe un chemin de a vers d dans le graphe G_1 ci-dessus, un arc de a vers d doit exister dans un graphe qui vérifierait les propriétés du *lemme 4.1*. Par superposition avec $O(v\ f_3)$ on obtient:

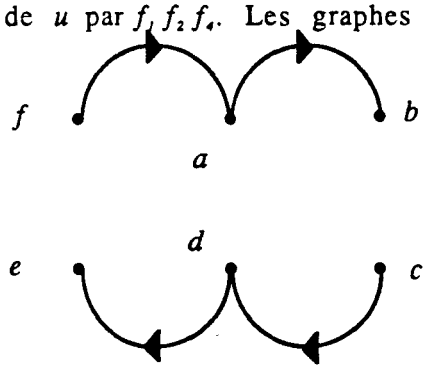


G_2

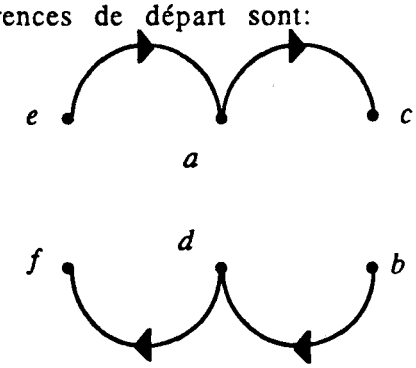
Pour les mêmes raisons, un arc de e vers f doit exister. Par superposition avec le graphe G_1, on obtient le graphe ci-dessous qui contient le cycle f,a,b,c,d,e,f.



Il n'existe donc pas de graphe vérifiant les propriétés du *lemme 4.1*, donc v n'est pas dans l'image de u par $f_1 f_2 f_3$. Vérifions maintenant si v est dans l'image de u par $f_1 f_2 f_4$. Les graphes des occurrences de départ sont:

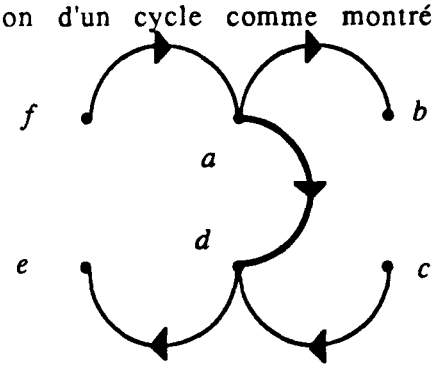


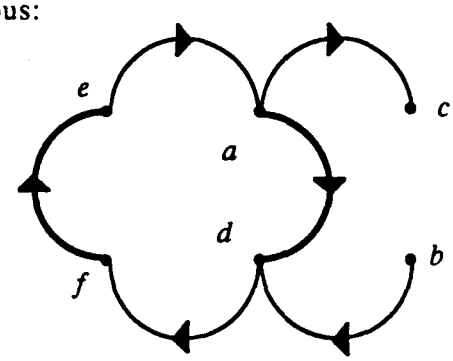
$O(u\ f_1)$



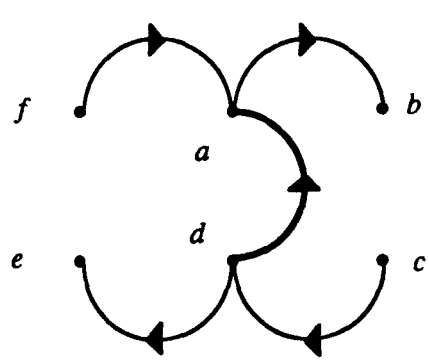
$O(v\ f_4)$

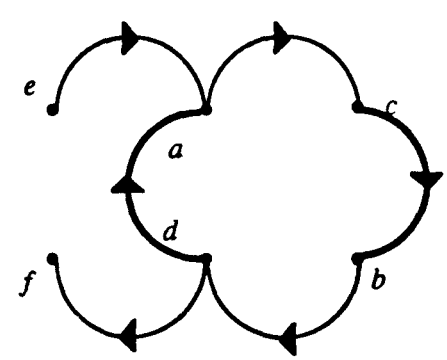
On s'aperçoit que, contrairement au premier cas, aucun ordre supplémentaire n'est à reporter. Cependant il est encore impossible de trouver un graphe vérifiant les propriétés du *lemme 4.1*. En effet le choix d'un arc de a vers d entraine alors la nécessité de l'existence d'un arc de f vers e et le choix d'un arc de d vers a entraine celle de l'existence d'un arc de c vers b. Dans les deux cas il y a formation d'un cycle comme montré ci-dessous:





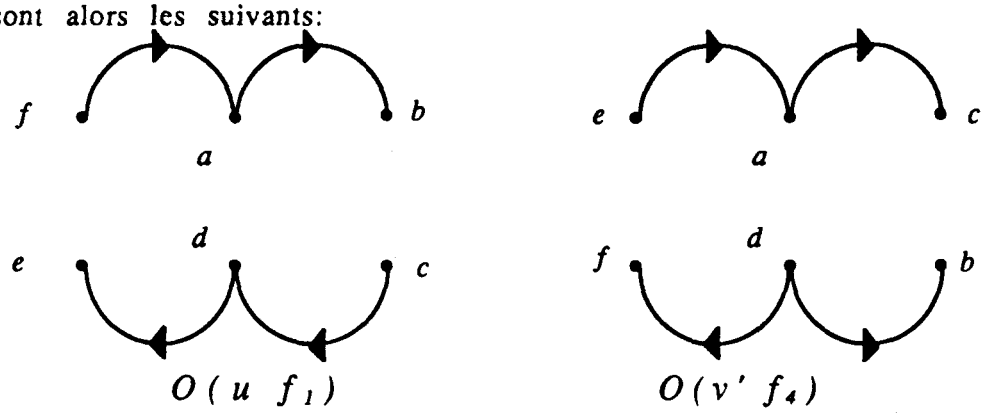
Choix d'un arc de a vers d



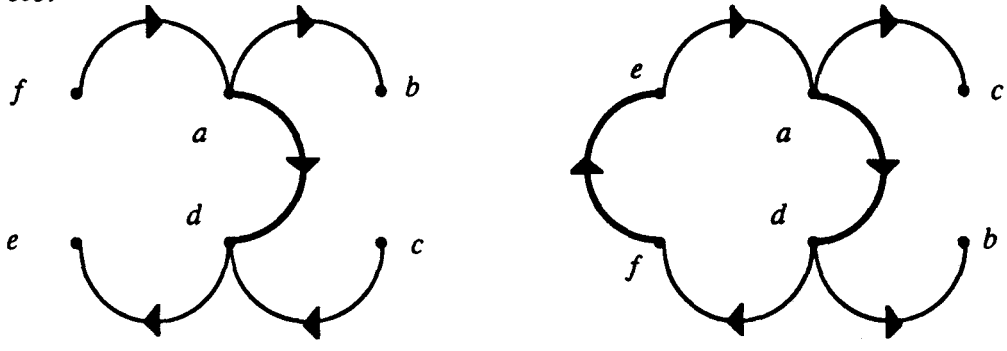


Choix d'un arc de d vers a

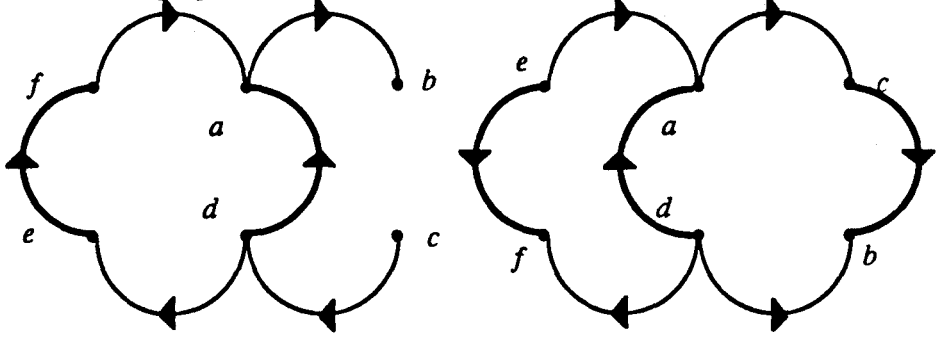
Vérifions enfin si $v'=eadbcf$ est dans l'image par $f_1 f_2 f_4$. Les graphes de départ sont alors les suivants:



Comme précédemment, aucun ordre supplémentaire n'est directement imposé. Cependant le choix d'un arc orienté de a vers d entraine alors la création d'un cycle:



Cela n'implique pas qu'il est impossible de trouver un graphe vérifiant les propriétés du *lemme 4.1*, mais simplement que le choix de l'existence d'un arc de a vers d était mauvais. En revanche le choix d'un arc de d vers a permet d'aboutir aux deux graphes ci-dessous:



Ces deux graphes permettent alors de construire respectivement les mots $cdefab$ et $defacb$ qui vérifient bien: $cdefab \in uf_1$, $defacb \in vf_4$ et $cdefab \in defacb f_2$.

Les trois parties de cet exemple nous montre la difficulté de mettre en pratique le *lemme 4.1*. Cependant, l'utilisation de ce lemme est beaucoup plus simple pour la composée de trois fonctions de commutation 2-partitionnée, ce qui sera particulièrement utile dans la section 6 où l'on traite du cas d'un alphabet de cinq lettres.

5. Mots irréductibles du monoïde Ψ_X^*.

Les règles de simplification définies dans la section 1 sont des outils permettant dans certains cas de savoir si une composition de fonctions de commutation partitionnée peut s'écrire plus simplement sous la forme d'une autre composition qui serait "plus courte". La notion d'irréductibilité est liée à ces problèmes de simplification.

Définition 5.1:[9] *Un mot u de* Ψ_X^* *est irréductible si il n'existe aucun mot v de* Ψ_X^* *vérifiant* $|v|<|u|$ *et* $v \equiv u$.

Bien que les compositions irréductibles soient en nombre fini comme l'indique la *proposition 1.7*, il n'est pas toujours facile d'assurer qu'une composition donnée est irréductible. La proposition suivante qui caractérise une famille particulière de compositions irréductibles de fonctions de commutation partitionnée sera fort utile dans la section suivante:

Proposition 5.2: *Soit un alphabet* $X=\{y,x_1,x_2,\ldots,x_{n-1},x_n\}$, *où* $n\geq 2$. *Alors* $\phi=(yx_1,X-\{y,x_1\})\ (yx_2,X-\{y,x_2\})\ldots(yx_{n-1},X-\{y,x_{n-1}\})$ *est irréductible.*

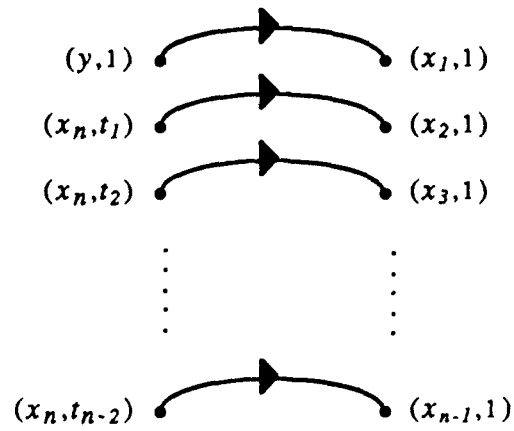
Preuve: Clairement, il suffit de montrer les deux propriétés suivantes:

1) ϕ n'est équivalent à aucun mot plus court formé uniquement avec des lettres apparaissant dans ϕ.

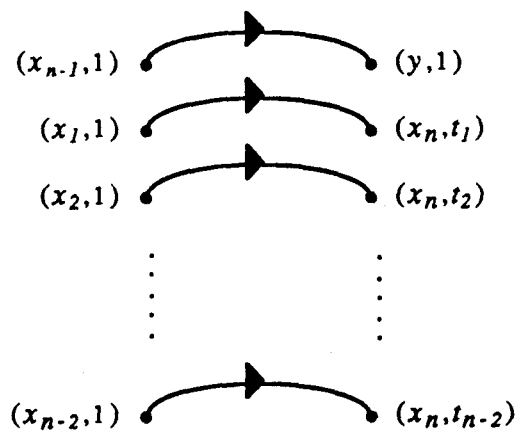
2) ϕ n'est pas plus grand que toute lettre de Ψ_X autre que celles qui le composent.

La première propriété est évidente: en effet si φ est un mot plus court de Ψ_X^*, formé à partir de certaines lettres composant ϕ, alors il existe un i dans $[1,n-1]$ tel que x_i et x_n ne peuvent jamais commuter, contrairement à ce que permet f_ϕ.

Pour montrer la deuxième propriété, nous allons tout d'abord faire la remarque suivante: Soit $u \in X^*$ tel que son graphe des occurrences contienne le sous-graphe G_u:



et soit $v \in X^*$ tel que son graphe des occurrences contienne le sous-graphe G_v:



Les t_i n'étant pas nécessairement tous différents.

Posons pour tout i de $[1,n-1]$, $\phi_i=(yx_i,X-\{y,x_i\})$. Nous allons vérifier que $v \notin uf_\phi$. En effet, dans le cas contraire, il existerait des mots $w_1, w_2, ..., w_{n-2}$ qui vérifieraient: $w_1 \in uf_{\phi_1}$, $w_{n-2} \in vf_{\phi_{n-1}}$, et $\forall\, i \in [3,n-2]$, $w_i \in w_{i-1}f_{\phi_i}$. Nous allons montrer, par induction sur i, que la propriété suivante doit alors être vérifiée: $\forall\, i \in [1,n-1]$, $w_i\, PS_{(y,1)(x_i,1)}=yx_i$. Comme $w_1 \in uf_{\phi_1}$, la propriété est vérifiée pour $i=1$. Supposons la propriété vérifiée pour $k \geq 1$, comme x_k et x_n restent toujours groupées dans $\phi_{k+1}\phi_{k+2}...\phi_{n-1}$ et qu'il existe un arc de $(x_k,1)$ vers (x_n,t_k) dans le graphe G_v, on déduit qu'il existe également un arc de $(x_k,1)$ vers (x_n,t_k) dans le graphes des occurrences de w_k.

De plus, par hypothèse de récurrence, il existe également un arc de $(y,1)$ vers $(x_k,1)$. Enfin comme x_n et x_{k+1} ont toujours été groupés dans $\phi_1\phi_2...\phi_k$ et qu'il existe un arc de (x_n,t_k) vers $(x_{k+1},1)$ dans le graphe G_u, cet arc existe également dans le graphe des occurrences de w_k. On déduit alors que $w_k\, PS_{(y,1)(x_{k+1},1)}=yx_{k+1}$. Comme $w_{k+1} \in w_k f_{\phi_{k+1}}$, on obtient $w_{k+1}\, PS_{(y,1)(x_{k+1},1)}=yx_{k+1}$. Pour $n-1$, la propriété s'écrit $w_{n-1}\, PS_{(y,1)(x_{n-1},1)}=yx_{n-1}$ ce qui est incompatible avec l'hypothèse $w_{n-1} \in v f_{\phi_{n-1}}$. Donc $v \notin u f_\phi$.

Pour chaque lettre φ de Ψ_X n'apparaissant pas dans ϕ, nous allons donc exhiber deux mots u et v tels que $v \in u f_\varphi$ et dont les graphes des occurrences contiennent respectivement les sous-graphes G_u et G_v, et nous aurons donc ainsi montré la propriété 2. Nous pouvons séparer l'ensemble des lettres à étudier en trois sous-ensembles:

a) L'ensemble des lettres supérieures ou égales à $(y,X-\{y\})$ pour lequel les mots $u=yx_1x_nx_2x_n...x_nx_{n-1}$ et $v=x_1x_nx_2x_n...x_nx_{n-1}y$ conviennent.

b) L'ensemble des lettres supérieures ou égales à $(yx_n,X-\{y,x_n\})$. On peut prendre dans ce cas les mots $u=yx_nx_1x_2...x_{n-1}$ et $v=x_1x_2...x_{n-1}yx_n$ où on a confondu toutes les occurrences de la lettre x_n: tous les t_i sont égaux.

c) L'ensemble des lettres supérieures ou égales à une lettre de la forme $(yx_{i_1}x_{i_2}...x_{i_p},X-\{y,x_{i_1},x_{i_2}...x_{i_p}\})$ avec $n-1\geq p>1$. Nous supposerons que $i_1<i_2<...<i_p$. Posons de plus $X-\{y,x_{i_1},x_{i_2}...x_{i_p}\}=\{x_{j_1},x_{j_2}...x_{j_q}\}$ avec $j_1<j_2<...<j_q$. Envisageons deux sous-cas:

c.1) x_n est dans le même groupe que y. Trois sous-cas sont à envisager:

c.1.1) $j_1>1$ et $j_q<n-1$. Alors il existe un entiers r tel que $i_r=j_1-1$. On peut prendre $u=x_nx_{i_{r+1}}x_n....x_nx_{i_p}yx_nx_{i_1}x_n...x_nx_{i_r}x_nx_{j_1}x_{j_2}...x_{j_q}$ et $v=x_{j_1}x_{j_2}...x_{j_q}x_nx_{i_{r+1}}x_n....x_nx_{i_p}yx_nx_{i_1}x_n...x_nx_{i_r}x_n$. En effet, on a bien y qui précède x_1 dans u puisque $x_1=x_{i_1}$. De même on a x_{n-1} qui précède y dans v puisque $x_{n-1}=x_{i_p}$. Enfin, on a confondu certaines occurrences de x_n (on considère que la dernière occurrence de x_n joue le rôle de toute les autres).

c.1.2) $j_1\geq 1$ et $j_q=n-1$. On peut dans ce cas prendre $u=yx_nx_{i_1}x_n...x_nx_{i_p}x_nx_{j_1}x_{j_2}...x_{j_q}$ et
$v=x_{j_1}x_{j_2}...x_{j_q}yx_nx_{i_1}x_n...x_nx_{i_p}x_n$.

c.1.3) $j_1=1$ et $j_q\leq n-1$. On peut dans ce cas prendre $u=x_nx_{i_1}x_n...x_nx_{i_p}x_nyx_{j_1}x_{j_2}...x_{j_q}$ et
$v=x_{j_1}x_{j_2}...x_{j_q}x_nx_{i_1}x_n...x_nx_{i_p}x_ny$.

c.2) x_n n'est pas dans le même groupe que y. Nous pouvons de nouveau séparer ce cas en quatre sous-cas:

c.2.1) $q=1$. On peut prendre $u=x_nx_{n-1}yx_1x_2\ldots x_{n-2}$ et $v=x_{n-1}yx_1x_2\ldots x_{n-2}x_n$ où on a confondu toutes les occurrences de la lettre x_n.

c.2.2) $j_1=1$ et $j_{q-1}=n-1$. On peut alors prendre u donné par

$u\ num_{/u/}=(x_n,t_{i_p})(x_{j_r},1)(x_n,t_{j_r})\ldots.$
$..(x_n,t_{j_{q-1}})(x_{j_q},1)(x_n,t_{i_1})\ldots.$
$..(x_n,t_{i_{p-1}})(x_{i_p},1)(y,1)(x_{j_1},1)(x_n,t_{j_1})\ldots.$
$..(x_{j_{r-1}},1)(x_n,t_{j_{r-1}})(x_{i_1},1)\ldots(x_{i_{p-1}},1),$

et v donné par $v\ num_{/u/}=(x_{i_p},1)(x_n,t_{i_p})(x_{j_r},1)(x_n,t_{j_r})\ldots$
$\ldots(x_n,t_{j_{q-1}})(x_{j_q},1)(y,1)(x_{i_1},1)\ldots$
$\ldots(x_{i_{p-1}},1)(x_n,t_{i_1})\ldots$
$\ldots(x_n,t_{i_{p-1}})(x_{j_1},1)(x_n,t_{j_1})\ldots.$
$..(x_{j_{r-1}},1)(x_n,t_{j_{r-1}}),$

où $j_r=i_p+1$. En effet i_p est différent de i_1 par hypothèse. De plus, on a bien dans $u\ num_{/u/}$, $(y,1)$ avant $(x_1,1)$ et dans $v\ num_{/u/}$, $(y,1)$ après $(x_{n-1},1)$ puisque $x_1=x_{j_1}$ et $x_{n-1}=x_{j_q}$. En outre, il faut vérifier que pour tout k de $[1,n-2]$, (x_n,t_k) est avant $(x_{k+1},1)$ dans $u\ num_{/u/}$ et $(x_k,1)$ est avant (x_n,t_k) dans $v\ num_{/u/}$. Cela est bien vérifié pour $k=i_p$ puisque $j_r=i_p+1$. De même pour $k\geq j_r$, pour la même raison. Comme $j_{r-1}<i_p$, cela est également vérifié pour $k\in\{i_1,\ldots,i_{p-1}\}$ et pour $k\in\{j_1,\ldots,j_{r-1}\}$.

c.2.3) $j_1>1$. On peut alors vérifier que les mots $u=x_nx_{j_1}x_nx_{j_2}\ldots.$
$\ldots.x_{j_q}x_nx_{i_r}\ldots x_{i_p}yx_{i_1}\ldots x_{i_{r-1}}$
et $v=x_{i_r}\ldots x_{i_p}x_nx_{j_1}x_nx_{j_2}\ldots x_{j_q}yx_{i_1}\ldots x_{i_{r-1}}x_n$ avec $i_r=j_1-1$ conviennent.

c.2.4) $j_{q-1}<n-1$. On vérifie, toujours de façon similaire, que les mots $u=x_nx_{i_r}\ldots x_{i_p}yx_{j_1}x_n\ldots x_{j_q}x_nx_{i_1}\ldots x_{i_{r-1}}$
et $v=x_{i_r}\ldots x_{i_p}yx_{i_1}\ldots x_{i_{r-1}}x_nx_{j_1}x_n\ldots x_{j_q}x_n$ avec $i_r=j_q+1$ conviennent. Ce dernier cas termine la preuve de la *proposition* 5.2 □

6. Le cas d'un alphabet de cinq lettres.

Dans cette section, nous allons étudier l'ensemble de toutes les compositions de fonctions de commutation partitionnée définies sur un alphabet de cinq lettres. Nous apportons en particulier une réponse négative à la question formulée dans [9] qui est de savoir si, sur un alphabet quelconque, toute composition de fonction de commutation partitionnée s'écrit de manière unique sous la forme d'une composition de fonctions de commutation irréductible qui serait alors la "forme normale" de la composition initiale.

Dans ce qui suit, $X=\{a,b,c,d,e\}$ désigne l'alphabet de cinq lettres considéré, et $\alpha,\beta,\gamma,\delta,\xi$ désigne une permutation quelconque des lettres a,b,c,d,e. Avec ces notations, nous pouvons séparer l'ensemble Ψ_X des partitions de l'alphabet X en 7 types:

Type 1: $(abcde)$ correspondant à la fonction où rien ne commute.
Type 2: $(\alpha,\beta\gamma\delta\xi)$
Type 3: $(\alpha\beta,\gamma\delta\xi)$
Type 4: $(\alpha,\beta,\gamma\delta\xi)$
Type 5:$(\alpha,\beta\gamma,\delta\xi)$
Type 6: $(\alpha,\beta,\gamma,\delta\xi)$
Type 7:(a,b,c,d,e) correspondant à la commutation totale.

Nous allons maintenant étudier les différents mots de longueur deux du monoïde Ψ_X^* engendré par Ψ_X, correspondant aux composées de deux fonctions de commutation partitionnée définies sur X. Plus précisemment, nous isolerons les mots de longueur deux de Ψ_X^* qui sont irréductibles. Nous pouvons tout d'abord remarquer que si une des lettres composant un mot de Ψ_X^* de longueur strictement plus grande que 1 est la lettre de type 1 ou la lettre de type 7, ce mot n'est alors pas irréductible. Dans le premier cas, on peut effacer la lettre de type 1, et dans le second on obtient la commutation totale: (a,b,c,d,e). Nous ne considérerons donc que des mots de Ψ_X^* contenant des lettres de type 2,3,4,5 ou 6. Et nous pouvons énoncer:

Lemme 6.1: *Il existe deux types de compositions de fonctions de commutation partitionnée sur l'alphabet $X=\{a,b,c,d\}$, associées aux deux types de mots de de Ψ_X^* suivants:$(\alpha\beta,\gamma\delta\xi)(\alpha\gamma,\beta\delta\xi)$ et $(\alpha,\beta\gamma,\delta\xi)(\alpha,\beta\delta,\gamma\xi)$.*

Preuve: Soit $u=\varphi\psi$ un mot de longueur 2 de de Ψ_X^* . Supposons tout d'abord que φ est une lettre de type 2 (on obtiendrait le même résultat dans le cas où ψ est une lettre de type 2). Alors, soit α est isolée dans ψ auquel cas $\varphi \leq \psi$ (cf *définition 1.2*). On obtient alors $\varphi\psi \equiv \psi$ d'après la règle 1.5. Sinon la règle 1.9 peut s'appliquer une ou plusieurs fois et il existe ψ' dans Ψ_X tel que $\varphi\psi \equiv \psi'\psi' \equiv \psi'$. Examinons sur un exemple ce type de réduction, les autres cas possibles se réduisant de manière similaire: Supposons $\varphi=(a,bcde)$ et $y=(bc,ade)$ alors $(a,bcde)(bc,ade) \equiv (a,bcde)(bc,a,de)$ (règle 1.9) et $(a,bcde)(bc,a,de) \equiv (a,bc,de)(a,bc,de)$ donc $\varphi\psi=(a,bc,de)$. Un raisonnement analogue pour le cas où φ est une lettre de type 4 ou 6 permet de conclure qu'un mot irréductible de longueur 2 de de Ψ_X^* ne peut contenir que des lettres de type 3 et des lettres de type 5. Supposons maintenant que φ est une lettre de type 3 et ψ une lettre de type 5. Posons $\varphi=(\alpha\beta,\gamma\delta\xi)$. Deux cas se présentent:

1) α ou β est isolée dans ψ. Supposons que $\psi=(\alpha,\beta\gamma,\delta\xi)$, le résultat serait identique dans les autres cas. Alors $\varphi\psi \equiv (\alpha,\beta,\gamma,\delta\xi)$ par applications successives de la règle 1.9.

2) Ni α ni β ne sont isolées dans ψ. Envisageons deux sous-cas:

2.1) $\psi=(\gamma,\alpha\beta,\delta\xi)$. Alors $\psi > \varphi$ et $\varphi\psi \equiv \psi$. Le résultat serait le même si δ ou ξ avaient été isolées dans ψ.

2.2) $\psi=(\gamma,\alpha\delta,\beta\xi)$. Alors $\varphi\psi \equiv (\gamma,\alpha\beta,\delta\xi)(\gamma,\alpha\delta,\beta\xi)$. On obtient donc la composée de deux lettres de type 5.

On peut donc considérer qu'un mot irréductible de longueur 2 de de Ψ_X^* ne peut être formé que de deux lettres de type 3 ou de deux lettres de type 5. Dans ce dernier cas, seul le type $(\alpha,\beta\gamma,\delta\xi)(\alpha,\beta\delta,\gamma\xi)$ ne se réduit pas grâce aux règles de la première section. On peut remarquer que dans ce cas, tout ce passe comme ci on travaillait sur l'alphabet de quatre lettres $\{\beta,\gamma,\delta,\xi\}$ puisque α est toujours isolé. On retrouve alors dans $(\beta\gamma,\delta\xi)(\beta\delta,\gamma\xi)$ la forme irréductible de la *proposition 5.2*, ce qui nous assure que $(\alpha,\beta\gamma,\delta\xi)(\alpha,\beta\delta,\gamma\xi)$ est bien un type de mot de deux lettres irréductible de Ψ_X^*. Les mots de Ψ_X^* formés de deux lettres de type 3 et qui ne se réduisent pas avec les règles de la première section sont également d'un seul type: $(\alpha\beta,\gamma\delta\xi)(\alpha\gamma,\beta\delta\xi)$. On remarque que deux lettres de X restent toujours groupées dans φ et dans ψ. Tout se passe comme si on travaillait sur un alphabet de 4 lettres, les différentes occurrences de ces lettres de X qui sont toujours groupées pouvant être considérées comme les différentes occurrences d'une même lettre. On reconnait alors, comme précédemment, la forme de mots irréductibles de la *proposition 5.2*, ce qui nous assure que ces deux derniers types de mots de longueur 2 de Ψ_X^* sont bien des mots irréductibles.

□

Nous allons maintenant isoler les mots irréductibles de trois lettres de Ψ_X^*:

Lemme 6.2: *Il existe 2 types de compositions irréductibles de longueur 3 de fonctions de commutation partitionnée sur l'alphabet $X=\{a,b,c,d,e\}$ associées aux 2 types de mots de Ψ_X^* suivants: $(\alpha\beta,\gamma\delta\xi)(\alpha\gamma,\beta\delta\xi)(\alpha\delta,\beta\gamma\xi)$ et $(\alpha\beta,\gamma\delta\xi)(\alpha\gamma,\beta\delta\xi)(\gamma\delta,\alpha\beta\xi)$.*

Preuve: Des résultats obtenus dans la preuve du *lemme 6.1*, on déduit au plus deux cas possibles de mots irréductibles de longueur 3 de Ψ_X^*. Il s'agit de la concaténation d'un mot du type $(\alpha\beta,\gamma\delta\xi)(\alpha\gamma,\beta\delta\xi)$ avec une lettre de Ψ_X de type 3, ou de la concaténation d'un mot du type$(\alpha,\beta\gamma,\delta\xi)(\alpha,\beta\delta,\gamma\xi)$ avec une lettre de type 5. Pour ce dernier cas, ce qui précède nous conduit à préciser que dans cette lettre de type 5, α doit être isolée. Comme nous l'avons déjà vu dans la preuve du *lemme 6.1*, tout se passe comme si on travaillait "séparément" avec l'alphabet de 4 lettres $\{\beta,\gamma,\delta,\xi\}$ puisque α a la possibilité de commuter à tout moment avec toutes les autres lettres de l'alphabet X par les fonctions de commutation associées. Les résultats déjà obtenus dans [9] concernant les compositions de fonctions de commutation partitionnée sur un alphabet de 4 lettres nous permettent donc de déduire qu'il n'existe pas de mot irréductible de longueur 3 dans Ψ_X^* contenant des lettres de type 5. En particulier, la *proposition 1.10* nous permet d'écrire: $(\alpha,\beta\gamma,\delta\xi)(\alpha,\beta\delta,\gamma\xi)(\alpha,\beta\varepsilon,\gamma\delta)\equiv(\alpha,\beta,\gamma,\delta,\xi)$ est la commutation totale. Il nous reste donc à étudier le premier cas: La concaténation d'un mot du type $(\alpha\beta,\gamma\delta\xi)(\alpha\gamma,\beta\delta\xi)$ avec une lettre de type 3. Soit φ cette lettre. Clairement φ doit être différent de $(\alpha\beta,\gamma\delta\xi)$, sinon, en application du *lemme 1.8*, le mot obtenu ne serait pas irréductible. On obtient alors, aux permutations près, deux formes possibles pour φ: $(\alpha\delta,\beta\gamma\xi)$ ou $(\alpha\beta\xi,\gamma\delta)$ d'où deux types possibles de mots:$(\alpha\beta,\gamma\delta\xi)(\alpha\gamma,\beta\delta\xi)(\alpha\delta,\beta\gamma\xi)$ et $(\alpha\beta,\gamma\delta\xi)(\alpha\gamma,\beta\delta\xi)(\alpha\beta\xi,\gamma\delta)$.

On peut reconnaitre dans le premier type la forme des mots irréductibles de la *proposition 5.2*, nous sommes donc sûr que ce premier type correspond bien à un type de mots irréductibles. Pour le deuxième type, si on suppose qu'il ne s'agit pas d'un type de mots irréductibles, alors, comme toutes les lettres de l'alphabet X commutent au moins une fois deux à deux par l'intermédiaire des fonctions de commutation associées, on déduit que ce type de mot est équivalent à un type de mot de Ψ_X^* de 2 lettres dont une au moins est de type 2,4,5,6 ou 7. Les preuves précédentes montrent que ce ne peut être qu'un type de mot de deux lettres de type 5 irréductible ou la commutation totale. Ce dernier cas est à écarter: par exemple, *beacd* n'est pas dans l'image de *abdec* par $f_{(ab,cde)(ac,bde)(cd,abe)}$. Pour ce qui concerne le premier cas, nous avons vu qu'une même lettre de X est toujours isolée dans les deux lettres de Ψ_X de type 5 qui forment des mots de Ψ_X^* de longueur 2 et irréductibles. D'autre part, $f_{(\alpha\beta,\gamma\delta\xi)(\alpha\gamma,\beta\delta\xi)(\gamma\delta,\alpha\beta\xi)}$ réalise clairement la commutation totale sur tout sous-alphabet de quatre lettres de $\{\alpha,\beta,\gamma,\delta,\xi\}$. On déduit donc que si $(\alpha\beta,\gamma\delta\xi)(\alpha\gamma,\beta\delta\xi)(\gamma\delta,\alpha\beta\xi)$ est équivalent à un mot de Ψ_X^* formé de deux lettres de type 5, la fonction associée doit réaliser la commutation totale sur le sous-alphabet de quatre lettres ne contenant pas celle qui est isolée dans ces deux éléments de type 5. Cela entraine alors qu'il ne peut s'agir que de la commutation totale sur l'alphabet complet, ce qui est en contradiction avec le résultat précédent et termine la preuve du *lemme 6.2*.

□

Nous allons maintenant étudier plus précisément les deux formes de composition associées aux deux types de mots de longueur trois irréductibles. Le lemme suivant montre que l'utilisation du *lemme 4.1* est plus simple dans le cas de ces compositions. Nous prendrons un représentant particulier pour chacun des deux types de mots irréductibles de longueur trois, ces résultats s'étendent ensuite en utilisant les permutations appropriées. Pour le premier type, nous prendrons comme représentant la fonction $f_{(ab,cde)(ac,bde)(ad,bce)}$ et pour le second, nous prendrons $f_{(ab,cde)(ac,bde)(cd,abe)}$. Soient u et v deux mots commutativement équivalents de $\{a,b,c,d,e\}^*$. Les graphes $O(u\, f_{(ab,cde)})$ et $O(v\, f_{(ad,bce)})$ déterminent sur $Z = alph(u\ num_{/u/})$ deux ordres stricts partiels que nous noterons respectivement R_0 et S_0. Posons $Z_{ac} = alph((u\ \Pi_{ac})\ num_{/u/})$ et $Z_{bde} = alph((u\ \Pi_{bde})\ num_{/u/})$. Soit la suite de relations définies sur Z par: $\forall\, i \in \mathbb{N}_+$, $R_{2i-1} = (R_{2i-2} \cup S_{2i-2}(bde))^+$, $S_{2i-1} = (S_{2i-2} \cup R_{2i-2}(bde))^+$, $R_{2i} = (R_{2i-1} \cup S_{2i-1}(ac))^+$, $S_{2i} = (S_{2i-1} \cup R_{2i-1}(ac))^+$, où pour toute relation T définie sur Z, $T(ac)$ et $T(bde)$ désignent respectivement les restrictions de T à Z_{ac} et à Z_{bde}. Comme u et v sont de longueur finie, il existe un entier m tel que $R_m(ac) = S_m(ac)$ et $R_m(bde) = S_m(bde)$. Avec ces notations, nous pouvons énoncer:

Lemme 6.3: *Le mot v est dans l'image de u par $f_{(ab,cde)(ac,bde)(ad,bce)}$ si et seulement si R_m et S_m sont deux ordres stricts (partiels) sur Z.*

Si on considère maintenant la fonction $f_{(ab,cde)(ac,bde)(cd,abe)}$, on obtient le même énoncé, R_0 et S_0 étant déterminés par les graphes $O(u\, f_{(ab,cde)})$ et $O(v\, f_{(cd,abe)})$. Nous allons prouver ce lemme pour la fonction $f_{(ab,cde)(ac,bde)(ad,bce)}$, l'autre cas se démontre de manière identique. La condition est clairement nécessaire. Pour montrer qu'elle est suffisante, nous utiliserons le résultat plus général exprimé dans le *lemme 6.5*, lui-même étant une conséquence du *lemme 6.4* suivant:

Lemme 6.4: *Soient X un ensemble fini, Y et Z deux sous-ensembles de X et R,R' deux ordres stricts définis sur X tels que $R' \cap (Z \times Z) = R \cap (Z \times Z)$ et $R' \cap (Y \times Y) \subset R \cap (Y \times Y)$. Alors $R'' = (R' \cup (R \cap (Y \times Y)))^+$ est un ordre strict sur X qui vérifie: $R \cap (Y \times Y) = R'' \cap (Y \times Y)$ et $R \cap (Z \times Z) \subset R'' \cap (Z \times Z)$.*

Preuve: Soit (x,x') un élément de $R''-(R' \cup (R \cap (Y \times Y)))$. Alors, il existe $x_0, x_1, \ldots, x_{n+1}$ vérifiant $x_0 = x$, $x_{n+1} = x'$ et $\forall\, i \in [0,n]$, $(x_i, x_{i+1}) \in R' \cup (R \cap (Y \times Y))$. Nécessairement, n est supérieur ou egal à 1. Supposons que n soit le plus petit entier vérifiant ces propriétés, alors on déduit que pour tout i dans $[1,n]$, x_i est un élément de Y. En effet, si il existe $j \in [1,n]$, tel que $x_j \notin Y$, on déduit que $(x_{i-1}, x_i) \in R'$ et $(x_i, x_{i+1}) \in R'$. Comme R' est un ordre, on a alors $(x_{i-1}, x_{i+1}) \in R'$ et n ne serait pas le plus petit entier vérifiant les conditions. Cela entraine que pour tout i dans $[1, n-1]$, (x_i, x_{i+1}) est un élément de $R \cap (Y \times Y)$ puisque $R' \cap (Y \times Y) \subset R \cap (Y \times Y)$. Comme R est transitive, on obtient que n est inférieur ou

égal à 2 et que x et x' ne peuvent appartenir tous les deux à Y puisque (x,x') n'est pas un élément de $R \cap (Y \times Y)$. Cette dernière conséquence entraine $R \cap (Y \times Y) = R'' \cap (Y \times Y)$. Clairement, $R \cap (Z \times Z) \subset R'' \cap (Z \times Z)$, il nous reste donc à vérifier que R'' est bien un ordre strict sur X. Par construction, R'' est transitive. Nous devons montrer que R'' est irréflexive. Supposons qu'il existe un élément x dans X tel que $(x,x) \in R''$. Comme $R \cap (Y \times Y) = R'' \cap (Y \times Y)$, x ne peut pas être un élément de Y. De plus, comme R' et R sont des ordres stricts, on déduit $(x,x) \in R''-(R' \cup (R \cap (Y \times Y)))$. D'après ce qui précède, deux cas sont à envisager:

1) Il existe $y \in Y$ tel que $(x,y) \in R' \cup (R \cap (Y \times Y))$ et $(y,x) \in R' \cup (R \cap (Y \times Y))$. Comme $x \notin Y$, on obtient $(x,y) \in R'$ et $(y,x) \in R'$ ce qui est impossible.

2) Il existe $y,y' \in Y$ tels que $(x,y) \in R' \cup (R \cap (Y \times Y))$, $(y,y') \in R \cap (Y \times Y)$, et $(y',x) \in R' \cup (R \cap (Y \times Y))$. Comme $x \notin Y$, on déduit, $(x,y) \in R'$ et $(y',x) \in R'$. Comme R' est transitive, on aurait alors $(y',y) \in R'$. Or $R' \cap (Y \times Y)$ est inclus dans $R \cap (Y \times Y)$ et $(y,y') \in R \cap (Y \times Y)$. Cette contradiction nous permet alors de conclure que R'' est bien un ordre strict sur X, ce qui termine la preuve du lemme.

□

On déduit de ce résultat le lemme suivant:

Lemme 6.5: *Soient X un ensemble fini, $\{Y,Z\}$, une partition de X et R,S deux ordres stricts définis sur X tels que $S \cap (Z \times Z) = R \cap (Z \times Z)$ et $S \cap (Y \times Y) \subset R \cap (Y \times Y)$. Alors il existe deux ordres stricts totaux sur X, R' et S' vérifiant $R \subset R'$, $S \subset S'$, $S' \cap (Z \times Z) = R' \cap (Z \times Z)$ et $S' \cap (Y \times Y) = R' \cap (Y \times Y)$.*

Preuve: Nous allons montrer ce lemme par induction sur n, le nombre d'éléments manquant dans R et S pour que ces ordres soient totaux Si $n=0$, on peut prendre $R'=R$ et $S'=S$. Si maintenant $n>0$, envisageons deux cas:

1) Si $S \cap (Y \times Y)$ est strictement inclus dans $R \cap (Y \times Y)$, par application du *lemme 6.4*, la relation $S''=(S \cup (R \cap (Y \times Y)))^+$ est un ordre strict sur X, contenant S et qui vérifie $R \cap (Y \times Y) = S'' \cap (Y \times Y)$ *et* $R \cap (Z \times Z) \subset S'' \cap (Z \times Z)$. On peut alors appliquer la relation de récurrence à S'' et R puisque le nombre de relations manquant dans R et S'' pour que ces ordres soient totaux est strictement plus petit que n.

2) Si $S \cap (Y \times Y) = R \cap (Y \times Y)$, on peut supposer que R n'est pas une relation totale, dans le cas contraire, on pourait appliquer le même raisonnement sur la relation S. Il existe donc $y \in X$ tel que l'ensemble des éléments qui lui sont incomparables par la relation R est non vide. Soit x, un élément minimal de cet ensemble. Posons $R''=(R \cup \{(x,y)\})^+$, clairement R'' est une relation irréflexive, par définition de x. Supposons $x \in Y$, nous allons montrer que dans ce cas $R'' \cap (Z \times Z) = S \cap (Z \times Z)$. En effet, soit un élément (z,z') de $(R'' \cap (Z \times Z))-(S \cap (Z \times Z))$. Comme $S \cap (Z \times Z) = R \cap (Z \times Z)$, on déduit que $(z,x) \in S$ et ($z'=y$ ou $(y,z') \in S$). Par définition de x, on obtient $(z,y) \in R$, sinon x ne serait pas minimal dans l'ensemble

des éléments incomparables à y par R ou (y,x) serait un élément de R. Donc $(z,y) \in S$ et $(z,z') \in S$, ce qui entraine $R'' \cap (ZxZ)=S \cap (ZxZ)$. De plus, par construction, on a $S \cap (YxY) \subset R'' \cap (YxY)$. Symétriquement, dans le cas où $x \in Z$, on obtient $R'' \cap (YxY)=S \cap (YxY)$ et $S \cap (ZxZ) \subset R'' \cap (ZxZ)$. Dans les deux cas, le nombre de relations manquant dans R'' et S pour que ces deux ordres soient totaux est strictement plus petit que n. On peut donc appliquer l'hypothèse de récurrence à R'' et S, ce qui termine la preuve du lemme.

□

Remarque: On retrouve ainsi le résultat de la *proposition 1.10*: La fonction $f_{(ab,cd)(ac,bd)(ad,bc)}$ réalise la commutation totale. En effet, pour tout couple de mots commutativement équivalents u,v de $\{a,b,c,d\}^*$, les ordres R et S déterminés respectivement par les graphes $O(u\, f_{(ab,cd)})$ et $O(v\, f_{(ad,bc)})$ vérifient $R \cap \{a,c\}x\{a,c\}=S \cap \{a,c\}x\{a,c\}=\varnothing$ et $R \cap \{b,d\}x\{b,d\}=S \cap \{b,d\}x\{b,d\}=\varnothing$. D'après le *lemme 6.5*, il est donc toujours possible de trouver deux mots w_1 et w_2 tels que $w_1 \in u\, f_{(ab,cd)}$, $w_2 \in v f_{(ad,bc)}$ et $w_1 \in w_2\, f_{(ac,bd)}$.

Nous sommes maintenant en mesure de montrer que la condition du *lemme 6.3* est suffisante: En effet si les relations R_m et S_m sont des relations d'ordres, elles vérifient les conditions du *lemme 6.5*, ce qui permet de conclure grâce au *lemme 4.1*: $v \in u f_{(ab,cde)(ac,bde)(ad,bce)}$.

Nous allons expliciter le *lemme 6.3* pour la fonction $f_{(ab,cde)(ac,bde)(cd,abe)}$ afin de comparer cette dernière à la fonction $f_{(bd,ace)}$. Nous pouvons en effet énoncer:

Lemme 6.6: *Pour tout mot u de $\{a,b,c,d,e\}^*$, l'image de u par $f_{(bd,ace)}$ est incluse dans l'image de u par $f_{(ab,cde)(ac,bde)(cd,abe)}$.*

Preuve: Soient u,v deux mots commutativement équivalents de $\{a,b,c,d,e\}^*$ tels que v n'est pas dans l'image de u par $f_{(ab,cde)(ac,bde)(cd,abe)}$. Nous allons montrer que dans ce cas v ne peut être dans l'image de u par $f_{(bd,ace)}$. Nous reprenons ici la construction des relations R_m et S_m utilisée pour le *lemme 6.3*, les relations R_0 et S_0 étant determinées par les graphes $O(u\, f_{(ab,cde)})$ et $(O(v\, f_{(cd,abe)})$. D'après le *lemme 6.3*, comme v n'est pas dans l'image de u par $f_{(ab,cde)(ac,bde)(cd,abe)}$, la relation R_m ou la relation S_m n'est pas irréflexive. Soit p le plus petit entier tel que R_p ou S_p ne soit pas irréflexive. Nous supposerons qu'il s'agit de S_p, la démonstration étant identique dans l'autre cas. Vérifions tout d'abord que $p>1$: Soient $(x,x') \in S_1\text{-}(S_0 \cup R_0(bde))$ et n le plus petit entier vérifiant: $\exists\, x_0,x_1,\ldots,x_{n+1}$ tels que $x_0=x$, $x_{n+1}=x'$ et $\forall\, i \in [0,n]$, $(x_i,x_{i+1}) \in S_0 \cup R_0(bde)$. Alors pour tout i dans $[1,n]$, $x_i \in Z_{bde}$: en effet supposons qu'il existe $j \in [1,n]$ tel que $x_j \notin Z_{bde}$. Alors, $(x_{j-1},x_j) \in S_0$ et $(x_j,x_{j+1}) \in S_0$ donc $(x_{j-1},x_{j+1}) \in S_0$ ce qui est incompatible avec la définition de n. On déduit $n<3$, d'où trois cas possibles:

- $\exists\, x_1,x_2 \in Z_{bde}$ tels que $(x,x_1) \in S_0$, $(x_2,x') \in S_0$, $(x_1,x_2) \in R_0(de)$,
- $\exists\, x_1 \in Z_{bde}$ tels que $(x,x_1) \in S_0$, $(x_1,x') \in R_0(de)$,
- $\exists\, x_1 \in Z_{bde}$ tels que $(x,x_1) \in R_0(de)$, $(x_1,x') \in S_0$.

Comme $S_0(de)=\varnothing$, x ne peut être égal à x' donc $p>1$. Nous supposerons dans la suite que p est un entier pair, la démonstration serait identique dans l'autre cas. Il existe donc $x \in Z$, $q \in \mathbb{N}$ tel que $p=2q+2$ et $(x,x) \in S_{2q+2}$. De la même façon que précédemment, on déduit trois cas possibles:

- $\exists\, x_1,x_2 \in Z_{ac}$ tels que $(x,x_1) \in S_{2q+1}$, $(x_2,x) \in S_{2q+1}$, $(x_1,x_2) \in R_{2q+1}(ac)$,
- $\exists\, x_1 \in Z_{ac}$ tels que $(x,x_1) \in S_{2q+1}$, $(x_1,x) \in R_{2q+1}(ac)$,
- $\exists\, x_1 \in Z_{ac}$ tels que $(x,x_1) \in R_{2q+1}(ac)$, $(x_1,x) \in S_{2q+1}$.

Dans tous les cas, on obtient: $\exists\ (a,i_{2\,q+1}^{a}),(c,i_{2q+1}^{c}) \in Z$ tels que $(((c,i_{2q+1}^{c}),(a,i_{2q+1}^{a})) \in R_{2q+1}\text{-}S_{2q+1}$ et $((a,i_{2q+1}^{a}),(c,i_{2q+1}^{c})) \in S_{2q+1}\text{-}R_{2q+1})$ ou $(((c,i_{2q+1}^{c}),(a,i_{2q+1}^{a})) \in S_{2q+1}\text{-}R_{2q+1}$ et $((a,i_{2q+1}^{a}),(c,i_{2q+1}^{c})) \in R_{2q+1}\text{-}S_{2q+1})$, en effet les autres possibilités entraineraient la non irréflexivité d'une des deux relations R_{2q+1} et S_{2q+1}. Nous pouvons nous placer dans le deuxième cas sans nuire à la généralité de la démonstration et nous allons montrer par induction la propriété suivante notée (P): si $((a,i_{2q+1}^{a}),(c,i_{2q+1}^{c})) \in R_{2q+1}\text{-}S_{2q+1}$ et R_{2q+1} est irréflexive alors il existe $((a,i_1^a),(b,i_0^b)) \in R_0$, $((e,i_0^e),(c,i_1^c)) \in R_0$, $((b,i_0^b),(e,i_0^e)) \in S_0$, $((b,i_2^b),(a,i_1^a)) \in S_0$, $((c,i_1^c),(d,i_2^d)) \in S_0$, $((a,i_3^a),(b,i_2^b)) \in R_0$, $((d,i_2^d),(c,i_3^c)) \in R_0$, ..., $((a,i_{2q+1}^{a}),(b,i_{2q}^{b})) \in R_0$, $((d,i_{2q}^{d}),(c,i_{2q+1}^{c})) \in R_0$.

Grâce aux résultats obtenus précédemment sur la construction de S_1, on montre facilement que le propriété (P) est vérifiée pour $q=0$. Considérons maintenant $((a,i_{2q+3}^{a}),(c,i_{2q+3}^{c})) \in R_{2q+3}\text{-}S_{2q+3}$, avec R_{2q+3} irréflexive. Grâce aux résultats précédents, on déduit deux cas possibles: $\exists\, ((b,i_{2q+2}^{b}),(d,i_{2q+2}^{d})) \in S_{2q+2}$ et $(((a,i_{2q+3}^{a}),(b,i_{2q+2}^{b})) \in R_{2q+2}$, $((d,i_{2q+2}^{d}),(c,i_{2q+3}^{c})) \in R_{2q+2})$ ou $(((a,i_{2q+3}^{a}),(d,i_{2q+2}^{d})) \in R_{2q+2}$, $((b,i_{2q+2}^{b}),(c,i_{2q+3}^{c})) \in R_{2q+2})$. Considérons tout d'abord le deuxième cas: si $((d,i_{2q+2}^{d}),(c,i_{2q+3}^{c})) \in R_0$, alors, $((a,i_{2q+3}^{a}),(c,i_{2q+3}^{c})) \in R_{2q+2}$, ce qui est contraire aux hypothèses puisque $R_{2q+2}(a,c)=S_{2q+2}(a,c)$, et si $((c,i_{2q+3}^{c}),(d,i_{2q+2}^{d})) \in R_0$, on aurait $((b,i_{2q+2}^{b}),(d,i_{2q+2}^{d})) \in R_{2q+3}$ et $((d,i_{2q+2}^{d}),(b,i_{2q+2}^{b})) \in R_{2q+3}$, ce qui est incompatible avec l'hypothèses R_{2q+3} est irréflexive. Pour la même raison, on déduit que dans le premier cas, $((a,i_{2\,q+3}^{a}),(b,i_{2q+2}^{b})) \in R_0$, et $((d,i_{2q+2}^{d}),(c,i_{2q+3}^{c})) \in R_0$. Le même raisonnement appliqué symétriquement à $(((b,i_{2q+2}^{b}),(d,i_{2q+2}^{d})) \in S_{2q+2})$ nous permet ensuite d'appliquer la relation de récurrence et de conclure que la propriété (P) est vérifiée.

Symétriquement, nous obtenons pour $((c,i_{2q+1}^{c}),(a,i_{2q+1}^{a})) \in S_{2q+1}\text{-}R_{2q+1}$, il existe $((c,j_1^c),(d,j_0^d)) \in S_0$, $((e,j_0^e),(a,j_1^a)) \in S_0$, $((d,j_0^d),(e,j_0^e)) \in R_0$, $((d,j_2^d),(c,j_1^c)) \in R_0$, $((a,j_1^a),(b,j_2^b)) \in R_0$, $((c,j_3^c),(d,j_2^d)) \in S_0$, $((b,j_2^b),(a,j_3^a)) \in S_0$, ..., $((c,i_{2q+1}^{c}),(d,j_{2q}^{d})) \in S_0$, $((b,j_{2q}^{b}),(a,i_{2q+1}^{a})) \in S_0$. Si on suppose maintenant que $v \in u\, f_{(bd,ace)}$, on doit avoir $(u\ num_{/u/})\ \Pi_{Z_{bd}}=(v\ num_{/u/})\ \Pi_{Z_{bd}}$ et $(u\ num_{/u/})\ \Pi_{Z_{ace}}=(v\ num_{/u/})\ \Pi_{Z_{ace}}$. Cela entraine que $((e,i_0^e),(c,i_1^c)) \in S$ et $((e,j_0^e),(a,j_1^a)) \in R$, où S et R sont les ordres déterminés respectivement par les graphes des occurrences de v et de u. On déduit ensuite $((b,i_0^b),(d,i_2^d)) \in S$ et $((d,j_0^d),(b,j_2^b)) \in R$. Une simple récurrence sur q permet alors de montrer $(((b,i_{2q}^{b}),(d,j_{2q}^{d})) \in S)$ et $(((d,j_{2q}^{d}),(b,i_{2q}^{b})) \in R)$ce qui est incompatible avec l'hypothèse: $(u\ num_{/u/})\ \Pi_{Z_{bd}}=(v\ num_{/u/})\ \Pi_{Z_{bd}}$. Cette contradiction termine la preuve du *lemme 6.6*.

□

Cette comparaison conduit à formuler une réponse négative à une question de [9]:

Lemme 6.7: *Les deux mots (ab,cde)(ac,bde)(cd,abe) et (ab,cde)(ace,bd)(cd,abe) de* Ψ_X^* *sont équivalents, donc les compositions de fonctions de commutation partitionnée ne s'écrivent pas toujours de façon unique comme la composée irréductible de fonctions de commutation partitionnée.*

Preuve: Nous avons vu que *(ab,cde)(ac,bde)(cd,abe)* était plus grand que *(ace,bd)*. Cela entraine que *(ab,cde)(ab,cde)(ac,bde)(cd,abe) (cd,abe)* est plus grand que *(ab,cde)(ace,bd)(cd,abe)*. Maintenant, comme *(ab,cde) (ab,cde)(ac,bde)(cd,abe)(cd,abe)* est équivalent à *(ab,cde)(ac,bde)(cd,abe)*, on déduit *(ab,cde)(ac,bde)(cd,abe)* est plus grand que *(ab,cde)(ace,bd)(cd,abe)*. De façon symétrique au *lemme 6.6*, on peut montrer que *(ab,cde)(ace,bd)(cd,abe)* est plus grand que *(ac,bde)*. Comme précedemment, on déduit alors que *(ab,cde)(ace,bd)(cd,abe)* est plus grand que *(ab,cde)(ac,bde)(cd,abe)*, donc que ces deux mots sont équivalents.

□

Nous allons maintenant étudier les mots de longueur 4 de Ψ_X*. Un grand nombre de ces mots se réduisent directement grâce aux règles de simplification. Deux cas sont à envisager:

(1) Un mot obtenu par concaténation de *(ab,cde)(ac,bde)(cd,abe)* avec une lettre de Ψ_X .

(2) Un mot obtenu par concaténation de *(ab,cde)(ac,bde)(ad,bce)* avec une lettre de Ψ_X .

Dans les deux cas, on sait que si on concatène au mot considéré une lettre qui n'est pas de type 3, ou une lettre apparaissant déjà dans ce mot, on obtient alors un mot qui n'est pas irréductible. Pour le cas *(1)*, il nous faut donc vérifier que les mots obtenus par la concaténation de *(ab,cde)(ac,bde)(cd,abe)* avec une lettre de l'ensemble *{(ad,bce), (ae,bcd), (bc,ade), (bd,ace), (be,acd), (ce,abd), (de,abc)}* ne sont pas irréductibles. Considérons les différents cas:

(ab,cde)(ac,bde)(cd,abe)(ad,bce) est équivalent à *(ab,cde)(a,be,c,d)* d'après la *proposition 1.10* en considérant *b* et *e* comme une seule lettre. On obtient ensuite, en appliquant la règle 1.9, *(ab,cde)(ac,bde)(cd,abe)(ad,bce)* est équivalent à *(a,b,c,d,e)*.

(ab,cde)(ac,bde)(cd,abe)(ae,bcd) est équivalent à *(ab,cde)(ac,bde)(cd,b,ae)* d'après la règle 1.9. En appliquant de nouveau cette règle, on obtient: *(ab,cde)(ac,bde)(cd,abe)(ae,bcd)* est équivalent à *(a,b,c,de)(c,d,b,ae)* donc à *(a,b,c,d,e)*.

(ab,cde)(ac,bde)(cd,abe)(bc,ade) est équivalent à *(ab,cde)(ace,bd)(cd,abe)(bc,ade)* d'après le *lemme 6.7*. Or *(ace,bd) (cd,abe)(bc,ade)* est équivalent à *(ae,b,c,d)* d'après la *proposition 1.10* où on considère *a* et *b* comme une seule lettre. Donc *ab,cde)(ac,bde)(cd,abe)(bc,ade)* est équivalent à *(ab,cde)(ae,b,c,d)* donc à *(a,b,c,d,e)* d'après la règle1.9.

(ab,cde)(ac,bde)(cd,abe)(bd,ace) est équivalent à *(ab,cde)(ace,bd)(cd,abe)(bd,ace)*. D'après la règle 1.8, on obtient alors *(ab,cde)(c,d,b,ae)* et comme précédemment *(ab,cde)(ac,bde)(cd,abe)(bd,ace)* est équivalent à *(a,b,c,d,e)*.

(ab,cde)(ac,bde)(cd,abe)(be,acd) est équivalent à *(ab,cde)(ac,bde)(cd,a,be)* d'après la règle 1.9. En appliquant de nouveau la règle 1.9, on obtient *(ab,cde)(ac,bde)(cd,abe)(be,acd)* est équivalent à *(ab,cde)(a,c,d,be)* donc à *(a,b,c,d,e)*.

(ab,cde)(ac,bde)(cd,abe)(ce,abd) ne peut être réduit en utilisant directement les règles de la première section ou les résultats précédents. Nous montrerons cependant dans le *lemme 6.8* que la fonction associée à ce mot réalise la commutation totale.

(ab,cde)(ac,bde)(cd,abe)(de,abc) est en fait du même type que le mot précédent, en effet, *(ab,cde)(ac,bde)(cd,abe)(de,abc)* est équivalent à *(ab,cde)(ace,bd)(cd,abe)(bc,ade)* d'après le *lemme 6.7*. Si on applique sur *(ab,cde)(ace,bd)(cd,abe)(de,abc)* la permutation qui échange *a* et *b* et qui échange *c* et *d*, on obtient *(ab,cde)(ac,bde)(cd,abe)(ce,abd)*. Le résultat du *lemme 6.8* s'appliquera donc également pour *(ab,cde)(ac,bde)(cd,abe)(de,abc)*.

Lemme 6.8: *La fonction associée au mot (ab,cde)(ac,bde)(cd,abe)(ce,abd) réalise la commutation totale sur X={a,b,c,d,e}.*

Preuve: Soient u et v deux mots commutativement équivalents de $\{a,b,c,d,e\}^*$. Considérons $w_1=(u \Pi_{ab})(u \Pi_{cde})$, $w_2=(v \Pi_{ab})(u \Pi_{de})(v \Pi_c)$ et $w_3=(v \Pi_{abd})(v \Pi_{ce})$. Clairement, $w_1 \in u f_{(ab,cde)}$ et $w_3 \in v f_{(ce,abd)}$. De plus, $w_2 \in w_1 f_{(ac,bde)}$, en effet, $w_2 \Pi_{ac}=(v \Pi_a)(v \Pi_c)=(u \Pi_a)(u \Pi_c)=w_1 \Pi_{ac}$, et $w_2 \Pi_{bde}=(v \Pi_a)(u \Pi_{de})=(u \Pi_a)(u \Pi_{de})=w_1 \Pi_{ade}$. De même, $w_2 \in w_3 f_{(cd,abe)}$: $w_2 \Pi_{cd}=(u \Pi_d)(v \Pi_c)=(v \Pi_d)(v \Pi_c)=w_3 \Pi_{cd}$, et $w_2 \Pi_{abe}=(v \Pi_{ab})(u \Pi_e)=(v \Pi_{ab})(v \Pi_e)=w_3 \Pi_{abe}$. Donc $v \in u f_{(ab,cde)} f_{(ac,bde)} f_{(cd,abe)} f_{(ce,abd)}$ et $f_{(ab,cde)(ac,bde)(cd,abe)(ce,abd)}$ réalise la commutation totale.

□

Pour le cas *(2)*, il faut vérifier que les mots obtenus par la concaténation de *(ab,cde)(ac,bde)(ad,bce)* avec une lettre de l'ensemble *{(ae,bcd), (bc,ade), (bd,ace),(be,acd),(cd,abe),(ce,abd),(de,abc)}* ne sont pas irréductibles. Comme pour le cas *(1)* tous les mots à étudier, à l'exception de *(ab,cde)(ac,bde)(ad,bce)(ae,bcd)*, se réduisent grâce a la règle 1.9, la *proposition 1.10* et le *lemme 6.7.* et nous conjecturons:

Conjecture 6.9: *La fonction associée au mot (ab,cde)(ac,bde)(ad,bce)(ae,bcd) réalise la commutation totale sur X={a,b,c,d,e}.*

Remarques:

-La concaténation de *(ab,cde)(ac,bde)(ad,bce)(ae,bcd)* avec une lettre quelconque de Ψ_X différente de *(ae,bcd)* et de *(abcde)* réalise clairement la commutation totale.

-Une réponse positive à cette conjecture nous permettrait de conclure qu'il n'existe pas de mot de longueur strictement plus grande que 3, dans $\Psi_{\{a,b,c,d,e\}^*}$, qui soit irréductible, et que toute composition de fonctions de commutation partitionnée définie sur un alphabet de cinq lettres peut être simulée par une composition d'au plus trois fonctions de commutation partitionnée.

7. La décision $f=com$.

Il semble très difficile d'établir une caractérisation des compositions de fonctions de commutation partitionnée qui soit simple et facile à vérifier. Les résultats obtenus sur des alphabets de 3, 4 ou 5 lettres et les différents cas possibles dans l'application du *lemme 4.1* nous incitent en particulier à émettre les conjectures suivante:

Conjecture 7.1: *Soit un alphabet* $X=\{y,x_1,x_2,\ldots,x_n\}$, *où* $n\geq 2$. *Alors la fonction de commutation associée au mot* $\phi=(yx_1,X-\{y,x_1\})\ (yx_2,X-\{y,x_2\})\ldots(yx_n,X-\{y,x_n\})$ *réalise la commutation totale.*

Conjecture 7.2: *Soit f une composition de fonctions de commutation partitionnée définie sur un alphabet X. Alors f réalise la commutation totale si et seulement si f réalise la commutation totale pour l'ensemble des mots de X* contenant une seule occurrence de chaque lettre de X.*

Une réponse positive à cette deuxième conjecture entrainerait également la vérification de la première. Il est en effet très facile de vérifier que la fonction de commutation, définie sur un alphabet $X=\{y,x_1,x_2,\ldots,x_n\}$ où $n\geq 2.$, et associée au mot $\phi=(yx_1,X-\{y,x_1\})\ (yx_2,X-\{y,x_2\})\ldots(yx_n,X-\{y,x_n\})$ réalise la commutation totale sur l'ensemble des mots de X^* ne contenant qu'une seule occurrence de chaque lettre de X. Les deux résultats suivant sont une contribution très partielle à la recherche d'une caractérisation des compositions réalisant la commutation totale.

Lemme 7.3: *Soient* $X=\{x_1,x_2,\ldots x_n\}$ *un alphabet et f une composition de fonctions de commutation partitionnée définie sur X, alors f réalise la commutation totale si et seulement si f réalise la commutation totale pour tout mot du langage* $(x_1x_2\ldots x_n)^*$.

Preuve: La condition est clairement nécessaire. Réciproquement, soit f une composition de fonctions de commutation partitionnée définie sur X réalisant la commutation totale sur $(x_1x_2\ldots x_n)^*$. Considérons un mot u quelconque de X^*. Alors il existe au moins un mot u' de $(x_1x_2\ldots x_n)^*$ admettant u comme sous-mot: on peut prendre par exemple $(x_1x_2\ldots x_n)^{/u/}$. Il existe donc une projection sélective p telle que $u'p=u$. Soit maintenant un mot v de X^* commutativement équivalent à u. D'après le *lemme 2.2*, on sait que $com\ p=p\ com$. On déduit alors qu'il existe un mot v' de X^* vérifiant $v' \in u\ com$ et $v'p=v$. Par hypothèse, $v' \in u'f$. Le *lemme 2.2* s'applique également pour f, et comme $v \in u'fp$, on déduit que $v \in u'pf=uf$. Donc f réalise la commutation totale sur X^*.

□

Proposition 7.3: *Soit* $X=\{x,y,z_1,z_2,\ldots,z_k\}$ *un alphabet. Soient* $\phi=(xyw_1,w_1')(xyw_2,w_2')\ldots(xyw_n,w_n') \in \Psi_X^*$ *et* $(xw_{n+1},yw_{n+1}') \in \Psi_X$ *où pour tout i de* $[1,n+1]$, w_i *et* w_i' *sont des mots de* X^*. *Alors,* $\phi(xw_{n+1},yw_{n+1}') \equiv (x,y,z_1,z_2,\ldots,z_k)$ *si et seulement si* $\phi \equiv (xy,z_1,z_2,\ldots,z_k)$.

Preuve: Si $\phi \equiv (xy,z_1,z_2,\ldots,z_k)$, on obtient alors $\phi(xw_{n+1},yw_{n+1}') \equiv (xy,z_1,z_2,\ldots,z_k)(xw_{n+1},yw_{n+1}')$ et en appliquant la règle 1.9, $\phi(xw_{n+1},yw_{n+1}') \equiv (x,y,z_1,z_2,\ldots,z_k)$. Réciproquement supposons que ϕ n'est pas équivalent à $(xy,z_1,z_2,\ldots,z_k)$, alors ϕ est strictement inférieur à $(xy,z_1,z_2,\ldots,z_k)$. Il existe donc deux mots u et v de X^* vérifiant: $u\ com=v\ com$, $u\ \Pi_{xy}=v\ \Pi_{xy}$ et $v \notin u f_\phi$. Soit h le morphisme de X^* vers $(X \cup \{x',y'\})^*$ défini par $\forall\ z \in alph(w_{n+1})$, $z\ h=y'x'zx'y'$, $\forall\ z \in alph(w_{n+1}')$, $z\ h=x'y'zy'x'$, et $x\ h=x$, $y\ h=y$. Soit h' le morphisme de $(X \cup \{x',y'\})^*$ vers X^* défini par $\forall\ z \in X$, $z\ h'=z$, $x'\ h'=x$ et $y'\ h'=y$. Enfin, posons $f=f_{\phi(xw_{n+1},yw_{n+1}')}$, $f'=f_\phi$ et $f''=f_{(xy,z_1,\ldots,z_k)}$, $g=f\ h'^{-1}$, $g'=f'\ h'^{-1}$ et $g''=f''\ h'^{-1}$

Soit $v'=v\ h$. Comme $u \in vf'$, et que f'' est la restriction de g à X, on déduit que $u \in v\ g''$. D'autre part, $v=v'\ \Pi_X$, on déduit alors du *lemme 2.2* qu'il existe un mot u' de $(X \cup \{x',y'\})^*$ vérifiant $u' \in v'\ g''$ et $u'\ \Pi_X=u$. Supposons maintenant que $v' \in u'\ g$, alors il existe un mot w' dans $(X \cup \{x',y'\})^*$ tel que $w' \in u'\ g'$ et $w' \in v'(f_{(xw_{n+1},yw_{n+1}')}, h'^{-1})$. En particulier, w' vérifie $w'\ \Pi_{xyx'y'}=u'\ \Pi_{xyx'y'}=v'\ \Pi_{xyx'y'}$. Soient a et b deux lettres distinctes de $alph(w')$, nous allons montrer que $w'\ \Pi_{ab}=v'\ \Pi_{ab}$. Envisageons quatre cas:

1) $\{a,b\} \subset \{x,y,x',y'\}$. Alors, comme $w'\ \Pi_{xyx'y'}=v'\ \Pi_{xyx'y'}$, on obtient bien $w'\ \Pi_{ab}=v'\ \Pi_{ab}$.

2) $\{a,b\} \subset \{x,x'\} \cup alph(w_{n+1})$ ou $\{a,b\} \subset \{y,y'\} \cup alph(w_{n+1}')$. Alors, comme $w' \in v'(f_{(xw_{n+1},yw_{n+1}')}, h'^{-1})$, on déduit $w'\ \Pi_{ab}=v'\ \Pi_{ab}$.

3) $((a=x$ *ou* $a=x')$ *et* $b \in alph(w_{n+1}'))$ *ou* $((a=y$ *ou* $a=y')$ *et* $b \in alph(w_{n+1}))$ *ou* $((b=x$ *ou* $b=x')$ *et* $a \in alph(w_{n+1}'))$ *ou* $((b=y$ *ou* $b=y')$ *et* $a \in alph(w_{n+1}))$. Supposons $a=x$ et $b \in alph(w_{n+1}')$, la démonstration serait identique dans les autres cas. Par construction de v', tout x et tout b apparaissant dans v' sont séparés par au moins un y'. Comme $w'\ \Pi_{xy'}=v'\ \Pi_{xy}$ et $w'\ \Pi_{by'}=v'\ \Pi_{by}$, on déduit $w'\ \Pi_{xby'}=v'\ \Pi_{xby}$, d'où $w'\ \Pi_{ab}=v'\ \Pi_{ab}$.

4) $(a \in alph(w_{n+1})$ *et* $b \in alph(w_{n+1}'))$ *ou* $(a \in alph(w_{n+1}')$ *et* $b \in alph(w_{n+1}))$. Supposons $a \in alph(w_{n+1})$ et $b \in alph(w_{n+1}')$. Par construction de v', une occurrence de a et une occurrence de b consécutives dans v' sont séparés, soit par $x'y'x'y'$ si a est avant b dans v, soit par $y'x'y'x'$ dans le cas contraire. Comme $w'\ \Pi_{x'y'}=v'\ \Pi_{x'y'}$, $w'\ \Pi_{by'}=v'\ \Pi_{by'}$, et $w'\ \Pi_{x'a}=v'\ \Pi_{x'a}$,, on déduit $w'\ \Pi_{ab}=v'\ \Pi_{ab}$.

Nous avons donc montré: $\forall\ \{a,b\} \subset X \cup \{x',y'\}$, $w'\ \Pi_{ab}=v'\ \Pi_{ab}$, donc $w'=v'$ et $v' \in u'\ g'$. D'après le *lemme 2.2*, on déduit alors $v'\ \Pi_X \in (u'\ \Pi_X)\ g'$, c'est à dire $v \in uf'$, ce qui est en contradiction avec les hypothèses. On déduit donc que v' n'est pas dans l'image de u' par g. On montre alors facilement, par induction sur n et en utilisant le *lemme 1.7* du chapitre 2 que $v'\ h'$ n'est pas dans l'image par f de $u'\ h'$. Nous avons donc montré que f ne réalisait pas la commutation totale, ce qui termine la preuve de la proposition.

□

8. Fonctions de rangement.

Dans cette section, nous nous intéressons aux propriétés de "*rangement*" des compositions de fonctions de commutation partitionnée. Il a été montré dans [8] que pour toute composition de fonctions de commutation partitionnée, il existe une composition de fonctions de commutation 2-partitionnée qui lui est équivalente. Nous allons donc restreindre l'étude à ces dernières.

Soient X un alphabet. Pour tout ordre strict total $<$ défini sur X, nous noterons: $K_< = x_1^* x_2^* \ldots x_n^*$, où $X=\{x_1,x_2,\ldots,x_n\}$ et $\forall\, i \in [1,n-1].\ x_i < x_{i+1}$, et $R_<$, l'application de X^* dans X^* définie par: $\forall\, u \in X^*,\ u R_< = (u\ com) \cap K_<$. Avec ces notations, nous introduisons les définitions suivantes:

Définition 8.1: *Soient X un alphabet, $<$ un ordre strict total défini sur X, f une fonction de $X^* \longrightarrow 2^{X^*}$ et $L \subset X^*$. Nous dirons que f **range L pour** $<$ si et seulement si $\forall\, u \in X^*,\ u R_< \in uf$.*

Définitions 8.2: *Soient X un alphabet, f une fonction de $X^* \longrightarrow 2^{X^*}$, et $<$ un ordre strict total défini sur X. Nous dirons que f **est une fonction de rangement pour** $<$ si f range X^* pour $<$.*

De manière plus générale, f est une **fonction de rangement** si il existe un ordre strict total $<$, défini sur X, tel que f soit une fonction de rangement pour $<$, et f est une **fonction de rangement universel** si f est une fonction de rangement pour tous les ordres stricts totaux définis sur X.

Exemple 8.3: Soit $X=\{y,x_1,x_2,\ldots,x_n\}$ où $n \geq 2$. On peut montrer facilement que $f=f_{(yx_1,X-\{y,x_1\})} f_{(yx_2,X-\{y,x_2\})} \cdots f_{(yx_n,X-\{y,x_n\})}$ est une fonction de rangement universel: Pour alléger l'écriture, posons pour $i \in [1,n]$, $f_i = f_{(yx_i,X-\{y,x_i\})}$. Soit $<$ un ordre strict total défini sur X. On peut supposer que $y<x_n$, la démonstration serait similaire dans le cas contraire. Soit $u \in X^*$, posons $w_1 = (u\ \Pi_{yx_1})(u\ \Pi_{X-\{y,x_1\}})$ et pour tout $i>1$, posons $w_i = (u\ \Pi_y)((u\ R_<)\ \Pi_{x_1\ldots x_i})(u\ \Pi_{x_{i+1}\ldots x_n})$. Alors, pour tout i vérifiant $1<i<n$, il est clair que $w_i \in w_{i-1} f_i$, en effet $w_i\ \Pi_{yx_i} = w_{i-1}\ \Pi_{yx_i} = yx_i$ et $w_i\ \Pi_{X-\{y,x_i\}} = w_{i+1}\ \Pi_{X-\{y,x_i\}} = ((u\ R_<)\ \Pi_{x_1\ldots x_{i-1}})(u\ \Pi_{x_{i+1}\ldots x_n})$. On obtient donc pour $i=n$, $w_n = (u\ \Pi_y)((u\ R_<)\ \Pi_{x_1\ldots x_{n-1}})(u\ \Pi_{x_n}) \in uf$. Maintenant, comme $w_n \in (u\ R_<) f_n$, et que $f f_n = f$, on déduit que $u\ R_< \in uf$. Donc f est une fonction de rangement universel.

Plus généralement, les compositions de fonctions de commutation partitionnée ne sont pas des fonctions de rangement, c'est à dire qu'elles ne sont des fonctions de rangement pour aucun ordre strict défini sur X, que dans des cas bien particulier. Plus précisemment, nous pouvons énoncer:

Lemme 8.4: *Soit $f=f_1f_2...f_n$ une composition de fonctions de commutation partitionnées définies sur un alphabet X. Alors f est une fonction de rangement si et seulement si pour toute paire de lettres distinctes de X, il existe i dans $[1,n]$ tels que ces deux lettres commutent par f_i.*

Preuve: La condition est clairement nécessaire. Nous allons établir la réciproque par induction sur n. Pour $n=1$, la seule fonction de commutation partitionnée vérifiant la condition est la fonction de commutation totale qui est donc une fonction de rangement universel. Supposons la propriété vérifiée pour n, et considérons la composition de fonctions de commutation partitionnée $f=f_1f_2...f_nf_{n+1}$. Si $f'=f_1f_2...f_n$ vérifie la propriété alors, par hypothèse de récurrence, f', donc f, sont des fonctions de rangement. Sinon, il existe des groupes de lettres de X qui ne commutent que par f_{n+1}. Posons $X_1,X_2,...X_k$ ces ensembles de lettres qui n'ont jamais commuté entre elles dans f'. Soit h le morphisme de X^* dans $(X-(X_1 \cup ... \cup X_k) \cup \{z_1,z_2,...,z_k\})^*$ défini par: $\forall\ i \in [1,k],\ \forall\ x \in X_i,\ x\,h=z_i$, et $\forall\ x \in X-(X_1 \cup ... \cup X_k),\ x\,h=x$. posons $f''=f'\,h$. Clairement h est compatible avec f' et $f'=f''\,h^{-1}$. Maintenant, f'' vérifie la condition et range $(X-(X_1 \cup ... \cup X_k) \cup \{z_1,z_2,...,z_k\})^*$ pour un ordre strict total $<$ défini sur $X-(X_1 \cup ... \cup X_k) \cup \{z_1,z_2,...,z_k\}$. On déduit donc que $v'=(u\,h)\,R_<$ est dans l'image de $u\,h$ par f''. D'après le *lemme 1.4* du chapitre 2, il existe un mot v de X^* vérifiant $v\,h=v'$ et $v \in u f'$. Ce mot v est donc élément d'un langage $x_1^*\cdots x_{j_1}^*X_{i_1}^*x_{j_1+1}^*\cdots X_{i_k}^*x_{j_k+1}^*\cdots x_{j_q}^*$, où les x_t sont les éléments de $X-(X_1 \cup ... \cup X_k)$. On en déduit ensuite, comme toutes les lettres des ensembles X_i commutent par f_{n+1}, qu'on peut par cette dernière fonction ranger séparément les X_i^* pour obtenir un mot commutativement équivalent à u et d'une forme voulue. La composition f est donc bien une fonction de rangement.

□

Le lemme précédant permet de savoir si il existe au moins un ordre pour lequel une composition de fonctions de commutation partitionnée f donnée, définie sur un alphabet X est une fonction de rangement. Cependant il ne permet pas de connaitre tous les ordres sur lesquels cette composition range X^*. Il est cependant décidable de savoir si f range X^* sur un ordre $<$ donné comme l'indique la proposition suivante:

Proposition 8.5: *Soit f une composition de fonctions de commutation partitionnée définie sur un alphabet X. La fonction f est une fonction de rangement pour un ordre donné $<$ si et seulement si f range sur cet ordre l'ensemble des mots de X^* contenant deux occurrences de chaque lettre de X.*

Preuve: La condition est évidemment nécessaire. Réciproquement, soient f une composition de fonctions de commuation partitionnée vérifiant la condition et u un mot de X^*. Nous allons montrer que f range tout mot w de X^* sur $<$ par induction sur $n=sup(|w|_x, x \in X)$. Cela est le cas si $n \leq 2$, par hypothèse et d'après le *lemme 2.2*. Suposons maintenant que f range tous les mots w de X^* vérifiant $sup(|w|_x, x \in X) \leq n$ avec $n \geq 2$ et considérons un mot u vérifiant $sup(|u|_x, x \in X) = n+1$. On peut considérer sans nuire à la généralité de la démonstration qu'il n'existe qu'une seule lettre a de X vérifiant $|u|_a = n$. Alors il existe $u_1, u_2, u_3, u_4 \in X^*$ vérifiant $u = u_1 a u_2 a u_3 a u_4$ avec $|u_1|_a = |u_1|_a = |u_1|_a = 0$, donc $u' = u\ num_{|u|}$ s'écrit $u' = u_1'(a,1)u_2'(a,2)u_3'(a,3)u_4'$. Soient h le morphisme de $alph(u')^*$ vers X^* défini par $\forall\ (x,i) \in alph(u'), (x,i)\ h = x$ et $u'' = u_1'(a,1)u_2'u_3'(a,3)u_4'$.. Comme pour toute lettre x de X, $u''\ h$ contient au plus n occurrences de x, on déduit des hypothèses de récurrence f range $u''\ h$ sur $<$. D'après le *lemme 2.2*, il existe donc un mot v'' qui est dans l'image de u'' par $f\ h^{-1}$ et qui vérifie de plus $v''\ h = (u''\ h)\ R_<$. Donc v'' s'écrit $v_1''(a,1)(a,3)v_2''$. Comme $u'' = u' \prod_{alph(u')-\{(a,2)\}}$, on déduit, toujours du *lemme 2.2*, qu'il existe un mot v' vérifiant $v' \in u'(f\ h^{-1})$ et $v'' = v' \prod_{alph(u')-\{(a,2)\}}$, donc $v' = v_1''(a,1)(a,2)(a,3)v_2''$.Maintenant, comme $v'\ h \in uf$, d'après le *lemme 2.2* et que $v''\ h = (u''\ h)\ R_<$, on déduit $v'\ h \in uf$ et $v'\ h = u\ R_<$. Donc f range u sur $<$.

□

Le résultat précédent permet donc, étant donné une composition de fonctions de commutation partitionnée, de connaitre tous les ordres pour lesquels cette composition est une fonction de rangement. Il suffit de tester, pour tous les ordres possibles, si la composition range tous les mots contenant exactement deux occurrences de chaque lettre de l'alphabet. Cette recherche risque cependant d'être particulièrement longue dans beaucoup de cas, en effet, si la composition à examiner est une fonction de rangement universel, on ne poura en être assuré qu'après avoir effectué tous les tests. La proposition suivante va nous permettre de calculer effectuvement et d'une façon plus simple tous les ordres pour lesquels une composition de fonctions de commutation partitionnée est une fonction de rangement.

Proposition 8.6: *Soit $f = f_1 f_2 \ldots f_n$ une composition de fonctions de commutation 2-partitionnées définies sur un alphabet X, où f_i est la fonction associée à la partition de X:$\{X_i, Y_i\}$. Alors f est une fonction de rangement pour un ordre donné $<$ si et seulement si $f_2 \ldots f_n$ range $X_1^* Y_1^*$ ou $Y_1^* X_1^*$ pour cet ordre.*

Preuve: Posons $f'=f_2...f_n$. Il est clair que si f' range $X_1^*Y_1^*$ ou $Y_1^*X_1^*$ sur $<$, alors f range X^* sur $<$, puisque, pour un mot u quelconque de X^*, les mots $(u \prod_{X_1})(u \prod_{Y_1})$ et $(u \prod_{Y_1})(u \prod_{X_1})$ sont dans l'image de u par f_1. Réciproquement, posons $X=\{x_1,x_2,...,x_p\}$ avec $x_1<x_2<...<x_p$ et supposons que f' ne range ni $X_1^*Y_1^*$ ni $Y_1^*X_1^*$. Alors, il existe deux mots $u=u_1u_2$ et $v=v_1v_2$ où u_1,v_2 sont dans X_1^* et u_2,v_1 sont dans Y_1^*, tels que f ne range ni u ni v. Soient $Z=\{z_1,z_2,...,z_p\}$ un alphabet disjoint de X, g le morphisme de X^* vers Z^* défini par $\forall\ x_i \in X$, $x_i\ g=z_i$, et h le morphisme de $(X \cup Z)^*$ vers X^* défini par $\forall\ x_i \in X$, $x_i\ h=x_i$ et $\forall\ z_i \in Z$, $z_i\ h=x_i$. Considérons $w=u_1((v_2v_1)\ g)u_2$, et $w'=w_1w_2...w_n$ où pour tout i de $[1,n]$, w_i est un mot de $\{x_i,z_i\}^*$ défini par $w_i=x_i^{|u|_{x_i}}z_i^{|u|_{z_i}}$ si $x_i \in X_1$ et $w_i=z_i^{|u|_{z_i}}x_i^{|u|_{x_i}}$ sinon. Il est clair que pour tout mot $w'' \in w\ (f_1\ h^{-1})$, $w''\ \prod_X = u_1u_2$ ou $w''\ \prod_Z = (v_1v_2)\ g$. Comme $w' \in ((u_1v_2v_1u_2)\ R_<)\ h^{-1})$, que $(u_1u_2)\ R_<$ n'est pas dans l'image de u_1u_2 par f', que $(v_1v_2)\ R_<$ n'est pas dans l'image de v_1v_2 par f', on déduit du *lemme 1.4* du chapitre 2 que w' n'est pas dans l'image de w par $(f\ h^{-1})$. Donc $w'\ h$ n'est pas dans l'image par f de $w\ h$. Comme $w'\ h=(w\ h)\ R_<$, on déduit que f ne range pas le mot $u_1v_2v_1u_2$ sur $<$, ce qui termine la preuve de la proposition.

□

On peut étendre cette proposition de la façon suivante: Soit X un alphabet. A tout langage K inclus dans X^* de la forme $Z_1^*Z_2^*...Z_p^*$ où les Z_i sont des sous-alphabets de X, disjoints deux à deux et non nécessairement différents de l'ensemble vide, on associe le langage $S(K)$, construit sur un alphabet $Z=\{z_1,z_2,..,z_p\} \cup \{z_1',z_2',..,z_p'\}$, défini par $S(K)=z_1z_2..z_p \sqcup z_1'z_2'..z_p'$. Pour toute fonction de commutation 2-partitionnée f définie sur X et associée à une partition $\{Y,Y'\}$, on définit une application l_f de Z vers 2^{X^*} par: $\forall\ z_i \in Z$, $z_i\ l_f=(Z_i \cap Y)^*$ et $\forall\ z_i' \in Z$, $z_i'\ l_f=(Z_i \cap Y)^*$. Enfin, on étend cette application de façon compatible avec le produit sur Z^* et on définit $L_f(K)=\{w\ l_f / w \in K\}$. Avec ces notations, nous pouvons énoncer:

Proposition 8.7: *Soit $f=f_1f_2...f_n$ une composition de fonctions de commutation 2-partitionnées définies sur un alphabet X, ou f_i est la fonction associée à la partition de X:$\{Y,Y'\}$. Alors f range $K=Z_1^*Z_2^*...Z_p^*$ sur un ordre donné $<$ si et seulement si $f_2...f_n$ range sur cet ordre au moins un des langages de l'ensemble $L_f(K)$.*

Preuve: La condition est suffisante. En effet soit L un langage de $L_f(K)$ tel que $f'=f_2...f_n$ range L sur $<$. Par définition des langages de $L_f(K)$, pour tout mot u de K, $(u\ f_1) \cap L$ est différent de l'ensemble vide. Donc f range u sur $<$. Réciproquement, comme dans la preuve de la *proposition 8.6*, nous allons supposer que f' ne range aucun langage de $L_f(K)$ et construire avec cette hypothèse un mot de K qui ne peut pas être rangé sur $<$ par f. Posons $L_f(K)=\{L_1,L_2,...,L_q\}$, alors il existe des mots $u_1,u_2,...,u_q$ vérifiant $\forall\ i \in [1,q]$, $u_i \in L_i$ et f' ne range pas u_i sur $<$. Comme nous l'avons vu dans la démonstration de la *proposition 8.6*, il suffit de trouver un mot w vérifiant: $\forall\ w' \in (wf_1)$, $\exists\ i \in [1,q]$ tel que u_i est un sous-mot de w, cela assurera alors que f ne range pas w puisque f' ne range pas u_i. Pour tout i de $[1,q]$,

posons $u_{i,k,Y,1}=u_i \Pi_{Z_k\cap Y}$, $u_{i,k,Y',0}=u_i \Pi_{Z_k\cap Y'}$, $u_{i,k,Y,0}=\varepsilon$ et $u_{i,k,Y',1}=\varepsilon$ si $u_i \Pi_{Z_k}=(u_i \Pi_{Z_k\cap Y})(u_i \Pi_{Z_k\cap Y'})$ sinon $u_{i,k,Y,0}=u_i \Pi_{Z_k\cap Y}$, $u_{i,k,Y',1}=u_i \Pi_{Z_k\cap Y'}$, $u_{i,k,Y,1}=\varepsilon$ et $u_{i,k,Y',0}=\varepsilon$. Posons, pour $k \in [1,p]$, pour $\alpha=Y$ ou $\alpha=Y'$ et pour $\beta \in \{1,0\}$, $w_{k,\alpha,\beta}=u_{1,k,\alpha,\beta}u_{2,k,\alpha,\beta}\ldots u_{q,k,\alpha,\beta}$, $w_k=w_{k,Y,1}w_{k,Y,0}w_{k,Y',1}w_{k,Y',0}$ et enfin $w=w_1w_2\ldots w_p$. Par construction, $w \in K$. D'autre part tout mot w' de wf_1 admet en sous-mot soit $w_{k,Y,1}w_{k,Y',0}$ soit $w_{k,Y',1}w_{k,Y,0}$ pour tout $k \in [1,p]$. On déduit alors: $\forall\, w' \in wf_1, \exists\, i \in [1,q]$ tel que u_i soit un sous-mot de w' ce qui entraine f n'est pas une fonction de rangement pour $<$.

□

Comme toute composition de fonctions de commutation partitionnée est équivalente à une composition de fonctions de commutation 2-partitionnée (voir règle 1.9 et [8]), cette proposition nous donne le moyen d'obtenir tous les ordres pour lesquels une composition donnée est une fonction de rangement. Si $f=f_1f_2\ldots f_n$ est une composition de fonctions de commutation 2-partitionnée définie sur un alphabet X, il suffit de calculer la suite d'ensembles de langages $K_1,K_2,\ldots,K_n$ où $K_1=L_{f_1}(X^*)$ et pour tout i de $[1,n-1]$, $K_{i+1}=\bigcup_{L\in K}(L_{f_{i+1}}(L))$. A tout langage de la forme $x_1^*x_1^*\ldots x_m^*$ appartenant à K_n où pour tout i de $[1,m]$ x_i est une lettre de X correspont alors un ordre $x_1<x_2<\ldots<x_n$ pour lequel, d'après la proposition précédente, f est une fonction de rangement.

Exemple 8.8: Soient $X=\{a,b,c,d,e,f\}$ et f la composition de fonctions de commutation associée au mot $(ab,cd,ef)(adf,bec)$ de Ψ_X^*. Tout d'abord, grâce à la règle 1.9, nous obtenons: $(ab,cd,ef)(adf,bec)\equiv(abcd,ef)(ab,cdef)(adf,bec)$. Nous pouvons maintenant construire K_1,K_2,K_3 en notant plus simplement par $(z_1z_2\ldots z_p)$ où les z_i sont des lettres de X, le langage $\{z_1,z_2,\ldots,z_p\}^*$. Nous obtenons alors:

$K_1=\{(abcd)(ef),(ef)(abcd)\}$,

$K_2=\{(ab)(cd)(ef),(cd)(ab)(ef),(cd)(ef)(ab),(ef)(ab)(cd),(ab)(ef)(cd)\}$,

et enfin, $K_3=((a)(d)(f)$ ш $(b)(c)(e))\cup((d)(a)(f)$ ш $(c)(b)(e))\cup((d)(f)(a)$ ш $(c)(e)(b))\cup((f)(a)(d)$ ш $(e)(b)(c))\cup((a)(f)(d)$ ш $(b)(e)(c))$. Ce qui nous permet de déduire que f est une fonction de rangement pour les ordres qui vérifient: ($a<d<f$ et $b<c<e$) ou ($d<a<f$ et $c<b<e$) ou ($d<f<a$ et $c<e<b$) ou ($f<a<d$ et $e<b<c$) ou ($a<f<d$ et $b<e<c$).

Pour terminer la section consacrée aux fonctions de rangement, signalons que la *conjecture 7.2* selon laquelle une composition de fonctions de commutation partitionnée réalise la commutation totale si et seulement si elle réalise la commutation totale sur l'ensemble des mots contenant une seule occurrence de chaque lettre de l'alphabet conduit à émettre la conjecture suivante concernant les fonctions de rangement universel:

Conjecture 8.9: *Les seules fonctions de rangement universel sont les fonctions de commutation totale.*

Chapitre 4: Semi-commutations et langages algébriques.

1. Semi-commutations et reconnaissabilité

2. Fonctions de semi-commutations algébrico-rationnelles

3. Le cas des langages de la forme u^*f

1. Semi-commutations et reconnaissabilité

Un certain nombre de propriétés ont été initialement montrées pour les fonctions de commutation partielle. Ainsi R. Cori et D. Perrin ont prouvé dans [18] que, si f est une fonction de commutation partielle définie sur un alphabet X, et L_1, L_2 deux langages rationnels inclus dans X^* et fermés pour f, alors $L_1 \cup L_2$ et $L_1 L_2$ sont des langages rationnels. Le résultat reste vrai dans le cas des fonctions de semi-commutation :

Lemme 1.1: [12] *Soit f une fonction de semi-commutation définie sur un alphabet X. Soient $L_1 \subset X^*$ et $L_2 \subset X^*$ deux langages rationnels vérifiant $L_1 = L_1 f$ et $L_2 = L_2 f$, alors $L_1 \cup L_2$ et $L_1 L_2$ sont des langages rationnels.*

De même, plusieurs conditions suffisantes pour que l'image de l'étoile d'un rationnel R par une fonction de commutation partielle soit rationnelle ont été successivement établies. Tout d'abord par M.P. Flé et G. Roucairol ([21]) dans le cas où R est fini et rigide: une partie L de X^* est **rigide** (pour f) si pour tout mot w de L et toutes lettres distinctes a,b de $alph(w)$, les deux lettres a et b ne commutent pas par f. Ce résultat a été étendu par R. Cori et D. Perrin ([18]) dans le cas où R est rigide mais non nécessairement fini, puis par R. Cori et Y. Métivier ([17]).

Enfin, Y. Métivier a généralisé dans [27] cette dernière preuve au cas des fonctions de semi-commutation. La condition obtenue, montrée également, mais par des techniques différentes, par M. Clerbout et M. Latteux dans [12] utilise une propriété du graphe de non-commutation de f pour $alph(R)$ qui est le graphe de non-commutation de la restriction de f à $alph(R)$ comme le graphe de commutation de f pour $alph(R)$ est le graphe de commutation de la restriction de f à $alph(R)$.

Proposition 1.2: [27] et [12] *Soit f une fonction de semi-commutation définie sur un alphabet X, et $R \subset X^*$, un langage rationnel vérifiant $R = Rf$. Si pour tout mot w de R, le graphe de non-commutation de alph(w) pour f est fortement connexe, alors $R^* f$ est un langage rationnel.*

En utilisant cette proposition, Y. Métivier a donné dans [27] une condition suffisante pour que la fermeture d'un langage rationnel par une fonction de semi-commutation reste rationnel. La condition porte sur les **facteurs itérants** du rationnel considéré. Si X est un alphabet, un mot w de X^* est un facteur itérant d'un langage L inclus dans X^* s'il existe u,v dans X^* tels que $uw^*v \subset L$.

Proposition 1.3: [27] *Soit f une fonction de semi-commutation définie sur un alphabet X, et $R \subset X^*$, un langage rationnel. Si pour tout facteur itérant w de R, le graphe de non-commutation de alph(w) pour f est fortement connexe, Rf est un langage rationnel.*

Nous terminerons cette section en remarquant que la condition de la *proposition 1.2* n'est pas une condition nécessaire. En effet, prenons $R=\{a,b,ba\}$ et f la fonction de semi-commutation associée à la relation $\{(a,b)\}$. Clairement, $Rf=R$ et le graphe de non-commutation du mot ba n'est pas fortement connexe, cependant $R^*f=\{a,b\}^*$ est un langage rationnel. Le résultat suivant, dont on peut trouver une preuve dans [12] et dans [20], montre qu'on obtient une condition nécessaire et suffisante dans le cas particulier où R ne contient qu'un mot.

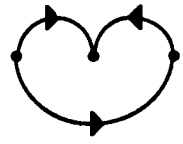
Proposition 1.4: [12] et [20] *Soit f une fonction de semi-commutation définie sur un alphabet X, et $w \in X^*$. w^*f est un langage rationnel si et seulement si le graphe de non-commutation de f pour alph(w) est fortement connexe.*

2. Fonctions de semi-commutations algébrico-rationnelles

Dans cette section, nous caractérisons les fonctions de semi-commutation f telles que pour tout langage rationnel R, Rf soit un langage algébrique. Nous appelons les fonctions de semi-commutation vérifiant cette propriété des fonctions **algébrico-rationnelles**.

Nous verrons que la condition s'exprime de manière assez simple: Une fonction de semi-commutation f définie sur un alphabet X est algébrico-rationnelle si et seulement si le graphe de commutation de X pour f de contient pas de sous-graphe du type

ni du type



les nœuds étant tous distincts.

Ce résultat entraine que toutes les fonctions de semi-commutation qu'on peut définir sur un alphabet de deux lettres sont des fonctions algébrico-rationnelles, ce que nous allons tout d'abord vérifier, puis nous effectuerons une

récurrence portant sur la taille de l'alphabet pour démontrer le résultat dans le cas général.

2.1. Le cas d'un alphabet de deux lettres.

Posons $X=\{a,b\}$. Les quatre fonctions de semi-commutation qui peuvent être définie sur X sont:

$f_1=Id$, la fonction qui ne fait rien commuter,
f_2 associée à la règle $ab \longrightarrow ba$,
f_3 associée à la règle $ba \longrightarrow ab$,
f_4 associée aux règles $ab \leftrightarrow ba$, la commutation totale sur X.

Dans [8], M. Clerbout a établi une condition nécessaire et suffisante pour qu'un mot w' soit dans l'image par f_2 d'un mot w de X^*:

Lemme 2.1: [8] *Soient w et w' deux mots de X^*, alors $w' \in w f_2$ si et seulement si $w' \in w\,com$ et $\forall (u,v) \in FG(w)xFG(w'), |u|=|v| \Longrightarrow |u|_b \leq |v|_b$.*

Nous pouvons alors énoncer:

Proposition 2.2: *Toute fonction de semi-commutation f définie sur $X=\{a,b\}$ est algébrico-rationnelle.*

Preuve: Si $f=f_1$, le résultat est évident: si $R \subset X^*$ est un élément de RAT, $R f=R \in RAT \subset ALG$. Si $f=f_4$, M. Latteux a déjà établi dans [22] que: $\forall R \in RAT$, $R f_4=R\,com \in Ocl$. Nous allons donc montrer que $R f_2$ est algébrique, pour tout langage rationnel R. Le résultat se démontre de manière symétrique pour f_3.

Soit h le morphisme défini sur $\{a,b,\bar{b}\}$ par: $a\,h=a$, $b\,h=b$, $\bar{b}\,h=\varepsilon$, et soit g le morphisme défini sur le même alphabet par: $a\,g=a$, $b\,g=\varepsilon$, $\bar{b}\,g=b$. On a alors $\forall u \in X^*$, $u f_2=\left((u\,h^{-1}) \cap (D_1'^*(\bar{b},b) \sqcup a^*) \right) g$. En effet, soit $u' \in u f_2$. Notons $\bar{u}'$ le mot formé à partir de u' en marquant toutes les occurrences de la lettres b. Posons $v=(u \sqcap \bar{u}') \cap (\bar{b}^* b^* a)^*$. D'après le *lemme 2.1*, il est clair que $v \Pi_{\bar{b}b} \in D_1'^*(\bar{b},b)$ donc: $v \in (u\,h^{-1}) \cap (D_1'^*(\bar{b},b) \sqcup a^*)$ et $u'=v\,g$. D'autre part, soit $w \in (u\,h^{-1}) \cap (D_1'^*(\bar{b},b) \sqcup a^*)$. Tout facteur gauche α de w vérifie: $|\alpha|_{\bar{b}} \geq |\alpha|_b$. Donc $w\,g \in (w \Pi_{ab}) f_2=u f_2$. Comme $D_1'^*(\bar{b},b) \in Ocl$ qui est fermée par transduction rationnelle, tout langage rationnel a donc son image par f_2 dans Ocl et f_2 est algébrico-rationnelle.

□

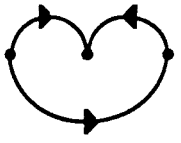
Remarque: Si L est un langage algébrique, $L f_1 = L \in ALG$, et il existe un langage rationnel R tel que $L f_4 = R f_4$ donc $L f_4 \in ALG$. Par contre $L f_2$ n'est pas toujours algébrique: Soit $L = \{ (ba)^n b^n , n \geq 0 \}$. $L \in ALG$, mais $(L f_2) \cap b^*a^*b^* = \{b^{n+k}a^n b^{n-k}, n \geq k \geq 0\} = L_1$, et $FG(b^*L_1) = \{b^n a^p b^q, n \geq p \geq q \geq 0\} \notin ALG$.

2.2. Le cas d'un alphabet quelconque.

Sur un alphabet de cardinalité supérieure à eux, il est clair qu'il faudra imposer des conditions sur une fonction de semi-commutation pour qu'elle soit algébrico-rationnelle.

Définition 2.3: *Soit f une fonction de semi-commutation définie sur un alphabet X. Nous dirons que f vérifie la condition notée (K) si le graphe de commutation de X pour f ne contient aucun sous-graphe du type*

ni du type

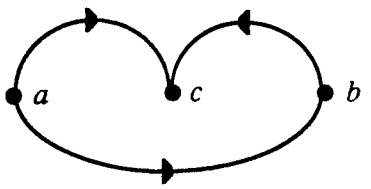


les nœuds de chacun de ces sous-graphes étant tous distincts.

Cette condition est équivalente à la condition: Pour tout couple (a,b) dans C, il n'existe pas de lettres c et d non nécessairement distinctes vérifiant $(a,c) \in C$ et $(d,b) \in C$. Avec cette définition nous pouvons énoncer:

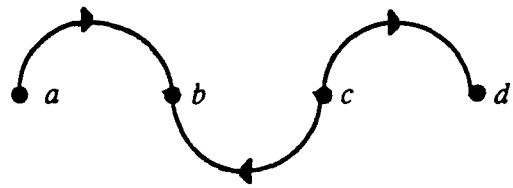
Proposition 2.4: *Si une fonction de semi-commutation est algébrico-rationnelle, elle vérifie alors la condition (K).*

Preuve: Soit f la fonction de semi-commutation définie sur $X = \{a, b, c\}$ par le graphe de commutation suivant:



Soit $R = (abc)^*$. Alors, $Rf \cap c^*b^*a^* = \{c^n b^n a^n, n \geq 0\} \notin ALG$.

Soit g la fonction de semi-commutation définie sur $\{a,b,c,d\}$ par le graphe de commutation suivant:



Soit $R'=(cd)^*(ab)^*$. Alors $R'\, g \cap d^*b^*c^*a^*=\{d^n b^p c^n a^p,\ n\geq 0,\ p\geq 0\} \notin ALG$.

Une fonction de semi-commutation ne vérifiant pas la condition (K) ne sera donc jamais algébrico-rationnelle.

□

Le reste de cette sous-section va être consacré à la preuve de la réciproque de cette proposition. Le principe général de la preuve sera une récurrence sur la taille de l'alphabet sur lequel on travaille. Cependant, pour certaines fonctions de semi-commutation, le résultat se démontre directement. Nous allons donc résoudre ces cas particuliers séparément. Mais, tout d'abord, énoncons un lemme que nous utiliserons souvent par la suite, dans lequel nous utilisons la notation suivante: Si X est un alphabet, w un mot de X^*, et t un entier positif, nous noterons $w(t)$ le facteur gauche de longueur t du mot w.

Lemme 2.5: *Soient X un alphabet, $u \in X^+$, $a \notin X$, $i>0$, $w \in a(u \sqcup a^i)$, $w' \in u \sqcup a^i a^+$. Alors il existe un entier t_0 positif tel que:*

1) $w(t_0)\ com = w'(t_0)\ com$

2) $\forall\, 0<s<t_0,\ |w'(s)|_a < |w(s)|_a$

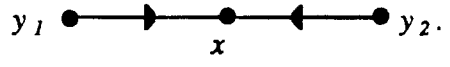
Preuve: Soit t_0 le plus petit entier t positif tel que $|w'(t)|_a \geq |w(t)|_a$. t_0 existe puisque, si $t=|w|$, on a $|w'(t)|_a \geq |w(t)|_a$. Alors $|w'(t_0)|_a = |w(t_0)|_a$ et donc $|w'(t_0)\ \Pi_X| = |w(t_0)\ \Pi_X|$, ce qui entraîne $w'(t_0)\ \Pi_X = w(t_0)\ \Pi_X$ puisque ces deux mots sont des facteurs gauches de u. Donc $w(t_0)\ com = w'(t_0)\ com$ et la propriété 2 est vérifiée par construction de t_0.

□

Nous allons donc nous intéresser aux fonctions suivantes: celles qui permettent à une lettre de commuter avec toute les autres dans le même sens:

Définition 2.6: *Une fonction de semi-commutation f définie sur un alphabet X vérifie la propriété (P) si et seulement si il existe une lettre x de X telle que pour toute lettre y de X, $yx \in xyf$ ou telle que pour toute lettre y de X, $xy \in yxf$.*

Posons donc f une fonction de semi-commutation définie sur X et vérifiant la condition (K) et la propriété (P), c'est à dire qu'il existe une lettre $x \in X$ telle que $\forall\, y \in X$, $yx \in xyf$. Le deuxième cas ($xy \in yxf$) se traiterait de manière symétrique. Nous pouvons alors cerner plus précisément ce que peut faire la fonction f: $\forall\, y_1, y_2 \in X\text{-}\{x\}$, $y_1y_2 \notin y_2y_1 f$. En effet, le graphe de commutation de X pour f contient déjà



On ne peut donc ajouter aucun arc entre y_1 et y_2 puisque f vérifie la condition (K). En revanche, on peut avoir des commutations du type $yx \longrightarrow xy$, $y \in X\text{-}\{x\}$. Nous pouvons alors décomposer l'alphabet X en trois ensembles disjoints: $X=X_1 \cup X_2 \cup \{x\}$, avec $X_1=\{y \in X\text{-}\{x\} / xy \in yxf\}$, $X_2=\{y \in X / xy \notin yxf\}$.

Cela veut dire que dans un mot w de X^* les occurrences de la lettre x vont se "déplacer" dans les deux sens; dans chaque facteur de w composé de lettres de X_1, mais une occurrence de x ne traversera une lettre de X_2 que de gauche à droite. En remplaçant dans w les nouvelles positions des occurrences marquées de la lettre x ($\bar{x}$ au lieu de x) d'un mot w' de wf, on pourra retrouver des mots de $D_1^*(x,\bar{x}) \sqcup X_1^*$ et des mots de $D_1'^*(x,\bar{x}) \sqcup (X_1 \cup X_2)^*$. C'est ce que nous allons démontrer dans le lemme suivant, où, pour un mot w de X^*, nous notons $\bar{w}$ l'image de w par le morphisme qui marque la lettre x: $m: X \longrightarrow X \cup \{\bar{x}\}$, $x\,m=\bar{x}$, et $\forall\, y \in X\text{-}\{x\}$, $y\,m=y$.

Lemme 2.7: *Soit f une fonction de semi-commutation définie sur X, vérifiant la condition (K) et pour laquelle il existe une lettre x telle que $\forall\, y \in X$, $yx \in xyf$. Soient $u \in X^*$ et $u' \in uf$. Alors il existe un mot v de $u \,\bar{m}\, \bar{u}'$ tel que: $v \in \big((D_1^*(x,\bar{x}) \sqcup X_1^*)(D_1'^*(x,\bar{x}) \sqcup (X_1 \cup X_2)^*)\big)^*$ où $X_1=\{y \in X\text{-}\{x\} / xy \in yxf\}$, $X_2=\{y \in X / xy \notin yxf\}$.*

Preuve: Raisonnons par induction sur la longueur du mot u. Si $|u|=0$, le résultat est évident. Si $|u|>0$, on a toujours $u\,\Pi_{X_1 \cup X_2} = u'\,\Pi_{X_1 \cup X_2}$ car les lettres de $X_1 \cup X_2$ ne peuvent pas commuter entre elles. On peut d'autre part supposer que u et u' commencent par deux lettres différentes: si u et u' commencent par une même lettre de $X_1 \cup X_2$, on applique l'hypothèse de récurrence aux mots privés de leur première lettre, et on a terminé; si u et u' commencent par la lettre x, on a $u=xw$, $u'=xw'$, et par hypothèse de récurrence, il existe $v_1 \in (w \,\bar{m}\, \bar{w}') \cap \big((D_1^*(x,\bar{x}) \sqcup X_1^*)(D_1'^*(x,\bar{x}) \sqcup (X_1 \cup X_2)^*)\big)^*$. Alors le mot $x\bar{x}v_1$ est dans $(u \,\bar{m}\, \bar{u}') \cap \big((D_1^*(x,\bar{x}) \sqcup X_1^*)(D_1'^*(x,\bar{x}) \sqcup (X_1 \cup X_2)^*)\big)^*$. On peut également supposer que $|u|_{X_2} \neq 0$: si $|u|_{X_2}=0$, $u\,f=(u\,\Pi_{X_1}) \sqcup x^i$, avec $i=|u|_x$ et donc $u \,\bar{m}\, \bar{u}' \subset D_1^*(x,\bar{x}) \sqcup X_1^*$. Donc u ou u' commence par x. Envisageons deux cas:

i) u' commence par x.

Alors u commence par une lettre de X_1 (si u commence par une lettre de X_2, u' aussi). On peut donc écrire: $u=u_1cu_2$, $u'=u_1'cu_2'$, $/u_1/_{X_2}=/u_1'/_{X_2}$, $c \in X_2$, et on a $u_1 \Pi_{X_1}=u_1' \Pi_{X_1}$ et $/u_1/_x \geq /u_1'/_x$ car $xc \notin cxf$. Donc $u_1 \in (u \Pi_{X_1}) \sqcup x^j$, $u_1' \in x((u \Pi_{X_1}) \sqcup x^i)$ avec $j=/u_1/_x$, $i+1=/u_1'/_x$, $j \geq i+1$, et on déduit ainsi du *lemme 2.5* qu'il existe des décompositions: $u_1=v_1w_1$, $u_1'=v_1'w_1'$, avec $v_1 \neq \varepsilon$ et $v_1\ com=v_1'\ com$ donc: $v_1 \amalg v_1' \cap D_1^*(x,\bar{x}) \sqcup X_1^* \neq \emptyset$. Alors, $(v_1 \amalg v_1')(w_1cu_2 \amalg \bar{w}_1'c\bar{u}_2') \subset u \amalg u'$. De plus, $v_1'w_1'cu_2' \in v_1w_1cu_2f$ et $v_1\ com=v_1'\ com$ impliquent $w_1'cu_2' \in w_1cu_2f$ (*lemme 3.3 du chapitre 2*). En appliquant l'hypothèse de récurrence à w_1cu_2, on obtient le résultat.

ii) u commence par x.

On peut alors écrire: $u \in x((u \Pi_{X_1 \cup X_2}) \sqcup x^i)$, $u' \in (u' \Pi_{X_1 \cup X_2}) \sqcup x^j$, avec $j=i+1=/u/_x$. En utilisant le *lemme 2.5*, on a $u=w_1w_2$, $u'=w_1'w_2'$, avec $w_1 \neq \varepsilon$, $w_1\ com=w_1'\ com$ et donc $w_2' \in w_2f$. Soit $\{v\}=w_1 \amalg \bar{w}_1' \cap (x^*\bar{x}^*(X_1 \cup X_2))^*$. Il est clair que $v \in D_1'^*(x,\bar{x}) \sqcup (X_1 \cup X_2)^*$, grâce à la propriété 2 du *lemme 2.5*, et $v(w_2 \amalg \bar{w}_2') \subset u \amalg u'$. En appliquant l'hypothèse de récurrence à w_2, on a terminé. □

Nous pouvons maintenant énoncer:

Proposition 2.8: *Soit f une fonction de semi-commutation définie sur un alphabet X, vérifiant la condition (K) et pour laquelle il existe une lettre x telle que: $\forall\ y \in X$, $yx \in xyf$. Alors il existe des morphismes h et g et deux sous-ensembles de X: X_1 et X_2 tels que: $\forall\ u \in X^*$, $uf=[(u\ h^{-1}) \cap ((D_1^*(x,\bar{x}) \sqcup X_1^*)(D_1'^*(x,\bar{x}) \sqcup (X_1 \cup X_2)^*))^*]\ g$. Donc f est algébrico-rationnelle.*

Preuve: Posons $X_1=\{y \in X-\{x\},\ xy \in yxf\}$ et $X_2=X-(X_1 \cup \{x\})$. Soient h et g les morphismes définis sur $X \cup \{\bar{x}\}$ par: $\forall\ y \in X_1 \cup X_2$, $y\ h=y$, $y\ g=y$, $x\ h=x$, $x\ g=\varepsilon$, $\bar{x}\ h=\varepsilon$, $\bar{x}\ g=x$. Posons $L=(D_1^*(x,\bar{x}) \sqcup X_1^*)(D_1'^*(x,\bar{x}) \sqcup (X_1 \cup X_2)^*)$. Nous allons montrer que $\forall\ u \in X^*$, $uf=(u\ h^{-1} \cap L^*)\ g$. Soit $u' \in uf$. Il est clair que $u \amalg \bar{u}' \subset u\ h^{-1}$ et, d'après le *lemme 2.7*, il existe un mot $v \in (u \amalg \bar{u}') \cap L^*$. D'autre part, $v\ g=u'$. Donc: $u' \in [(u\ h^{-1}) \cap ((D_1^*(x,\bar{x}) \sqcup X_1^*)(D_1'^*(x,\bar{x}) \sqcup (X_1 \cup X_2)^*))^*]\ g$.

Réciproquement, soit $u' \in u\ h^{-1} \cap L^*$. Il existe donc un entier p tel que $u' \in u\ h^{-1} \cap L^p$. Nous allons faire une récurrence sur p: Si $p=0$, $u'=\varepsilon$, donc $u=\varepsilon$ et $u'\ g \in uf$. Si $p \geq 1$, $u' \in u\ h^{-1} \cap LL^{p-1}$. Donc $u'=u_1'u_2'v'$ avec $u_1' \in D_1^*(x,\bar{x}) \sqcup X_1^*$, $u_2' \in D_1'^*(x,\bar{x}) \sqcup (X_1 \cup X_2)^*$ et $v' \in L^{p-1}$. On peut alors écrire $u=u_1u_2v$ avec $u_1=u_1' \Pi_X$, $u_2=u_2' \Pi_X$, et $v=v' \Pi_X$. De plus, $u_1'\ g \in u_1\ f$ (avec les seules commutations $xy \leftrightarrow yx$, $y \in X_1$) puisque $u_1' \Pi_{X_1}=u_1 \Pi_{X_1}$ et $/u_1'/_x=/u_1/_x=/u_1'/_{\bar{x}}$. De même $u_2'\ g \in u_2\ f$: en effet, il est clair que $(u_2'\ g)\ com=(u_2\ f)\ com$ et $\forall 0<t<|u_2'\ g|$, $/u_2'\ g\ (t)/_x \geq /u_2\ g\ (t)/_x$. Et, par hypothèse de récurrence, $v'\ g \in v\ f$ $(v' \in u\ h^{-1} \cap L^{p-1})$. Donc: $u'\ g=(u_1'\ g)(u_2'\ g)(v'\ g) \in (u_1\ f)(u_2\ f)(v\ f) \subset u_1u_2v\ f=u\ f$.

□

Si, par une fonction de semi-commutation f, aucune lettre ne peut tout traverser, les mélanges vont se faire plus localement. Pour obtenir l'image d'un mot par f, l'idée est alors de se ramener à faire des commutations sur des facteurs construits sur des sous-alphabets plus petits. C'est ce que détaille le lemme suivant.

Lemme 2.9: *Soit f une fonction de semi-commutation définie sur un alphabet X, vérifiant la condition (K) mais pas la propriété (P). Alors, pour tout mot u de X^*, pour tout mot v de uf, il existe des décompositions $u=u_1u_2$ et $v=v_1v_2$ avec $u_1\neq\varepsilon$, $alph(u_1)\subsetneq X$ et $v_1 \in u_1 f$.*

Preuve: Si $alph(u)\subsetneq X$, le résultat est évident. Sinon, posons $u=au'$, $v=dv'$, $a,d \in X$. Si $a=d$ on peut prendre $u_1=v_1=a$. Si $a\neq d$, posons $D=\{z \in X-\{a\} / za \in azf\}$, et $Y=X-(D \cup \{a\})$. Alors: $d \in D$ donc $D\neq\emptyset$, $Y\neq\emptyset$ car f ne possède pas la propriété (P), et si z_1 et z_2 sont dans D, le graphe de f contient le sous-graphe z_1 ●—◀—●—▶—● z_2 (arc étiqueté a), on ne peut donc avoir aucune commutation entre z_1 et z_2 car f vérifie la condition (K). Envisageons deux cas:

i) Il n'existe pas de lettres $y \in Y$ et $z \in D$ telles que $zy \in yzf$.

Posons alors $u=awyu''$, $v=dw'y'v''$ avec $w,w' \in (D \cup \{a\})^*$ et $y,y' \in Y$. Aucune occurrence de la lettre a dans aw ne peut franchir le y car: $\forall x \in Y$, $xa \notin axf$. On a donc $|aw|_a\leq|w'|_a$ d'où $|w|_a<|w'|_a$. D'autre part, pour deux lettres (d_1,y_1) de DxY, on peut avoir $y_1d_1 \in d_1y_1f$, mais comme $d_1y_1 \notin y_1d_1f$, on a $dw' \Pi_D \in FG(w \Pi_D)$. Alors, $u=u'yu''$, $u' \in a((dw' \Pi_D) \sqcup a^i)(w'' \sqcup a^{i'})$, en effet, $w \Pi_D=(dw' \Pi_D)w''$. $v=v'y'v''$, $v' \in (dw' \Pi_D) \sqcup a^j$, avec $j=|w'|_a$, $i+i'=|w|_a$, donc $j>i$. D'après le *lemme 2.5*, on peut alors écrire: $u'=u_1u_2'$, $v'=v_1v_2'$, avec $u_1\neq\varepsilon$ et $u_1 com=v_1 com$ (donc $v_1 \in u_1 f$). De plus, $alph(u_1) \subset (D \cup \{a\}) \subsetneq X$. Le couple (u_1,v_1) convient donc.

ii) Il existe une lettre $y \in Y$ et une lettre $d_1 \in D$ telles que $d_1y \in yd_1f$.

Alors, $D=\{d\}$. En effet, si D contient deux lettre distinctes d_1 et d_2, on aura dans le graphe de commutation de X pour f le sous-graphe y ●—▶—●—◀—●—▶—● d_2 (d_1, a), ce qui est contraire à l'hypothèse.

Posons alors $E=\{z \in X-\{d\} / dz \in zdf\}$ et $Z=X-(E \cup \{d\})$. On a: $y \in E$ donc $E\neq\emptyset$, $Z\neq\emptyset$ car f ne possède pas la propriété (P), et si z_1 et z_2 sont deux lettres distinctes de E, le graphe de f contient le sous-graphe z_1 ●—▶—●—◀—● z_2 (d), on ne peut donc avoir aucune commutation entre z_1 et z_2 car f vérifie la condition (K). Posons alors $u=awzu''$, $v=dw'z'v''$ avec $w,w' \in (E \cup \{d\})^*$ et $z,z' \in Z$. Aucune occurrence de la lettre d dans u'' ne peut franchir z ou z', mais on peut avoir des règles du type $dt \longrightarrow td$ pour une lettre t de Z.. On a donc $|dw'|_d\leq|aw|_d$ d'où $|w'|_d<|w|_d$. D'autre part, $\forall x \in E-\{a\}$, $\forall z \in Z$, on a dans le graphe de commutation de X pour f le sous-graphe a ●—▶—●—◀—● x (d). on ne peut donc pas rajouter un arc de x vers z. De plus $za \notin azf$ car $D=\{d\}$ et $d \notin Z$. Donc, $\forall x \in E$, $\forall z \in Z$, $zx \notin xzf$. Mais on peut avoir la règle $z_1x_1 \longrightarrow x_1z_1$ pour un couple (x_1,z_1) de ExZ, on a donc $(aw \Pi_E) \in FG(w' \Pi_E)$. Alors, $u=u'zu''$, $u' \in (aw \Pi_E) \sqcup d^j$, $v=v'z'v''$, $v' \in d((aw \Pi_E) \sqcup d^i)(w'' \sqcup d^{i'})$, en effet, $dw' \Pi_E=(aw \Pi_E)w''$ avec $j=|aw|_d$,

$i+i'=|w'|_d$, donc $j>i$. D'après le *lemme 2.5*, on peut alors écrire: $u'=u_1u_2'$, $v'=v_1v_2'$, avec $u_1\neq\varepsilon$ et $u_1\ com=v_1\ com$ (donc $v_1 \in u_1 f$). De plus, $alph(u_1) \subset (E \cup \{d\}) \subsetneq X$. Le couple (u_1,v_1) convient donc, ce qui termine la preuve.

□

Nous pouvons maintenant énoncer la résultat principal de cette section:

Proposition 2.10: *Soit f une fonction de semi-commutation définie sur un alphabet X, vérifiant la condition (K), alors f est algébrico-rationnelle.*

Preuve: Nous raisonnons par induction sur $|X|$, la cardinalité de l'alphabet X. Si $|X|=2$, le résultat est donné par la *proposition 2.2*. Si $|X|>2$, alors soit f vérifie la propriété (P) et le résultat est donné par la *proposition 2.8*, sinon, nous allons montrer que pour tout rationnel R, Rf est un langage algébrique. Soit $R \in RAT$. Il existe alors un automate déterministe $M=(X,Q,q_0,*,T)$ qui reconnait R. Si $q,q' \in Q$, posons $R_{q,q'}=\{w \in X^* / q*w=q'\}$. Soit s, la substitution définie sur $Q\times Q$ par: $\forall (q,q') \in Q\times Q$, $(q,q')s = \bigcup_{x\in X} (R_{q,q'} \cap (X-\{x\})^*)$.

Par hypothèse de récurrence, s est une substitution algébrique. Soit T le langage rationnel, défini sur $Q\times Q$ par: $T=\{ (q_0,q_1)(q_1,q_2)\ldots(q_{p-1},q_p) / p\geq 1$, $\forall i \in [1,p]$, $q_i \in Q$, $q_p \in T\}$. Nous allons montrer que $Rf=Ts$, et nous aurons terminé puisque le résultat d'une substitution algébrique appliquée à un langage rationnel est un langage algébrique.

i) $Ts \subset Rf$?

Soit $w=(q_0,q_1)(q_1,q_2)\ldots(q_{p-1},q_p) \in T$. $\forall w' \in ws$, $w'=u_0u_1\ldots u_{p-1}$ avec $u_i \in (q_i,q_{i+1})s$. Donc: $\forall i, \exists x_i \in X, \exists v_i \in R_{q_i,q_{i+1}} \cap (X-\{x_i\})^*$ tels que $u_i \in v_i f$, et $w' \in (v_0 f)(v_1 f)\ldots(v_{p-1}) \subset v_0v_1\ldots v_{p-1} f$. Comme $q_0*v_0v_1\ldots v_{p-1}=q_p \in T$, $v_0v_1\ldots v_{p-1} \in R$. Ainsi, $ws \subset Rf$.

ii) $Rf \subset Ts$?

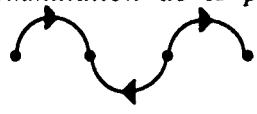
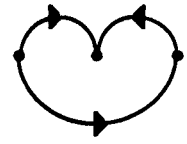
Si q_1 et q_2 sont deux états de Q, posons $T_{q,q'}=\{ (q_1,q_2)(q_2,q_3)\ldots(q_p,q_{p+1}) / p\geq 1$, $q_1=q$, $q_{p+1}=q'\}$ Nous allons montrer que pour tout mot w de $R_{q,q'}$, pour tout mot w' de wf, $w' \in T_{q,q'}s$, en faisant une récurrence sur la longueur de w. Si $alph(w) \subsetneq X$, $wf \subset (q,q')s$. Sinon, nous pouvons, grâce au *lemme 2.9*, trouver deux décompositions: $w=w_1w_2$, $w'=w_1'w_2'$, avec $w_1\neq\varepsilon$, $alph(w_1) \subsetneq X$ et $w_1' \in w_1 f$. Si $q*w_1=q_1$, on a $w_1 f \subset (q,q_1)s$ et, par hypothèse de récurrence, $w_2 f \subset T_{q_1,q'}s$. Donc $w' \in (w_1 f)(w_2 f) \subset ((q,q_1)T_{q_1,q'})s \subset T_{q,q'}s$. Nous pouvons alors conclure la démonstration de proposition: Si $w \in R$, il existe un état q de T tel que $w \in R_{q_0,q}$. Alors, pour tout mot $w' \in wf$, $w' \in T_{q_0,q}s \subset Ks$.

□

Les *propositions 2.4* et *2.10* montrent qu'en fait, l'image d'un langage rationnel par une fonction de semi-commutation algébrico-rationnelle est toujours dans *Ocl*. Nous pouvons alors énoncer:

Proposition 2.11: *Soit f une fonction de semi-commutation définie sur un alphabet X. Il y a équivalence entre*

(1) f est algébrico-rationnelle

(2) Le graphe de commutation de X pour f ne contient aucun sous-graphe du type  *ni du type* 

(3) Pour tout langage rationnel R, $Rf \in Ocl$.

(4) Pour tout langage rationnel R borné, $Rf \in ALG$.

Preuve:

(1) $\Rightarrow$ (2) par la *proposition 2.4*

(2) $\Rightarrow$ (3) car toutes les constructions qui nous ont permis d'obtenir l'image par *f d'un langage rationnel utilisent* $D_1'^*$ et D_1^* qui sont dans *Ocl* et des opérations pour lesquelles *Ocl* est fermée.

(3) $\Rightarrow$ (4) est un résultat évident.

(4) $\Rightarrow$ (1) en reprenant un des exemples de la preuve de la *proposition 2.4*.

□

En particulier, pour les fonctions de commutation partielle, le graphe de commutation associé est non-orienté, on obtient alors:

Proposition 2.12: *Une fonction de commutation partielle f définie sur un alphabet X est algébrico-rationnelle si et seulement si le graphe de commutation de X pour f ne contient aucun chemin de longueur 3.*

3. Le cas des langages de la forme $u^* f$

Pour être en mesure de caractériser les fonctions de semi-commutation f et les mots u vérifiant: $u^* f \in ALG$, nous allons tout d'abord montrer quelques résultats plus généraux qui sont pour la plupart des conséquences de propriétés énoncées dans le chapitre 2. Ainsi, nous utiliserons la conséquence suivante du *corollaire 3.7* du chapitre 2:

Lemme 3.1: *Soient X un alphabet, et f une fonction de semi-commutation définie sur X^*. Alors, $\forall u \in X^*$, $\forall u_1,u_2 \in L=FG(u^* f)$, $(u_1\ com)=(u_2\ com) \Rightarrow (u_1^{-1}L=u_2^{-1}L)$.*

Preuve: Soit $w_1 \in u_1^{-1}L$, alors: $\exists p \in /N$, $\exists w_1' \in X^*$ tels que $u_1 w_1 w_1' \in u^p f$. D'autre part, $\exists w_2 \in X^*$, $\exists q \in /N$ tels que $u_2 w_2 \in u^q f$. Supposons $p \geq q$, la démonstration se ferait de manière identique dans le cas contraire. Alors, $u_2 w_2 u^{p-q} \in u^p f$, et du *corollaire 3.7* du chapitre 2 on déduit: $\exists v_1,v_2 \in X^*$ tels que $v_1 v_2 \in u^p f$, $u_1 \in v_1 f$, $u_2 \in v_1 f$, $w_1 w_1' \in v_2 f$, et $w_2 u^{p-q} \in v_2 f$. Donc: $u_2 w_1 w_1' \in (v_1 f)(v_2 f) \subset v_1 v_2 f \subset u^p f$, et $w_1 \in u_2^{-1}L$. De manière symétrique, on peut montrer $u_2^{-1}L \subset u_1^{-1}L$.

□

On déduit également facilement du *lemme 3.3* du chapitre 2 le résultat suivant:

Lemme 3.2: *Soit f une fonction de semi-commutation définie sur un alphabet X, et u un mot de X^* vérifiant: $\forall x \in X$, $|u|_x=1$. Soient $q \in /N_+$ et $w \in L=FG(u^* f)$ vérifiant $\forall x \in X$, $|w|_x \geq q$, alors le mot w', obtenu à partir de w en effaçant la $q^{ème}$ occurrence de chaque lettre de X est un élément de L.*

Si le graphe de non-commutation de l'alphabet d'un mot u pour une fonction de semi-commutation f est fortement connexe, les mélanges des lettres dans un élément de $u^* f$ seront limités. C'est ce qu'exprime le lemme suivant.

Lemme 3.3: *Soit f une fonction de semi-commutation définie sur un alphabet X telle que le graphe de non-commutation de X pour f soit fortement connexe. Soit $u \in X^*$ tel que: $\forall x \in X$, $|u|_x=1$, alors pour tout facteur w d'un mot de u^*f, si il existe une lettre y vérifiant $|w|_y \geq 2|X|$ alors il existe une lettre x vérifiant $|w|_x \geq 1$.*

Preuve: Soient y et x, deux lettres distinctes de X. Le graphe de non-commutation de X pour f étant fortement connexe, on déduit: $\exists\, z_0,z_1,z_2,\ldots,z_p,z_{p+1},\ldots,z_q \in X$ tels que: $z_0=z_q=y$, $z_p=x$, $\forall\, i\neq j \in [0,p]$ $z_i\neq z_j$, $\forall\, i\neq j \in [p,q]$ $z_i\neq z_j$, $\forall\, i \in [1,q]$ $z_iz_{i-1} \notin z_{i-1}z_i f$. Posons $v=z_0z_1z_2\ldots.z_q$, et $t=2|X|$.De ce qui précède, on déduit: $|v|< t$ et $v f=\{v\}$. Considérons, maintenant, $w \in F(u^* f)$, vérifiant $|w|_y \geq t$, alors: $\exists\, w_1,w_2,w_3,w_4,w_5 \in X^*$, $\exists\, i,j \in IN$ tels que: $w = w_2\, y_j\, w_3\, y_{j+t}\, w_4$ et $w'=w_1w_2y_jw_3y_{j+t}w_4w_5 \in u^i f$, où y_j est la $j^{ème}$ occurrence de y dans w'. Donc: $\exists\, v_1,v_2,v_3,v_4 \in X^*$ tels que:$w' \in v_1y_jv_2u^{t-1}v_3y_{j+t}v_4 f$. Comme $\forall\, z \in X$, $|u|_z=1$, on déduit: v est un sous-mot de $y_jv_2u^{t-1}v_3y_{j+t}$, avec $z_0=y_j$ et $z_q=y_{j+t}$, et, comme $v f=\{v\}$, v est un sous-mot de $y_jw_3y_{j+t}$,donc $|w|_x \geq 1$, et la propriété est bien vérifiée.

□

On peut étendre facilement ce résultat pour obtenir:

Lemme 3.4: *Soit f une fonction de semi-commutation définie sur un alphabet X. Soit $\{X_1,X_2,\ldots,X_n\}$, l'ensemble des composantes fortement connexes du graphe de non-commutation de X pour f. Soit $u \in X^*$ tel que: $\forall\, x \in X$, $|u|_x=1$, alors: $\forall\, w \in F(u^*f)$, $\forall\, m \in IN_+$, $\forall\, k \in [1,n]$, $(\exists\, y \in X_k / |w|_y \geq 2m|X_k|) \Rightarrow (\forall\, x \in X_k, |w|_x \geq m)$.*

Nous allons caractériser les fonctions de semi-commutation f et les mots u vérifiant: $u^*f \in ALG$, en fonction du nombre de composantes fortement connexes du graphe de non-commutation de f pour $alph(u)$. Si ce nombre est strictement plus grand que 2, nous pouvons énoncer:

Proposition 3.5: *Soient f une fonction de semi-commutation définie sur un alphabet X, et u un mot de X^* vérifiant: $alph(u)=X$. Si le nombre de composantes fortement connexes du graphe de non commutation de X pour f est strictement supérieur à 2, alors u^*f n'est pas un langage algébrique.*

Preuve: Soit $\{X_1,X_2,\ldots,X_n\}$, l'ensemble des composantes fortement connexes du graphe de non-commutation de X pour f. Si n est supérieur ou égal à 3, on peut vérifier facilement que: $\exists\, i\neq j \in [1,n]$ tels que $\forall\, x \in X_i$, $\forall\, y \in X\text{-}X_i$, $xy \in yxf$, et $\forall\, x \in X_j$ $\forall\, y \in X\text{-}X_j$ $yx \in xyf$. Posons $u_1=u\,\Pi_{X_i}$, $u_3=u\,\Pi_{X_j}$, $u_2=u\,\Pi_{X-(X_i \cup X_j)}$. Comme $n\geq 3$, u_2 est différent du mot vide et: $\forall\, i \in [1,3]$, $\forall\, j\neq i \in [1,3]$, $alph(u_i)\cap alph(u_j)=\emptyset$. On déduit alors que: $u^*f\cap u_1^*u_2^*u_3^*=\{\, u_1^n u_2^n u_3^n, n \in \mathbb{N}\,\} \notin ALG$. □

Considérons, maintenant, le cas où le nombre de composantes fortement connexes du graphe de non-commutation de $alph(u)$ pour f est strictement inférieur à 3. Si ce nombre est égal à 1, la *proposition 1.4* nous indique alors que $u^*f \in RAT$. Il nous reste donc à étudier le cas où le nombre de composantes fortement connexes du graphe de non-commutation de $alph(u)$ pour f est égal à 2, et nous allons montrer que, dans ce cas, $u^*f \in ALG$.

Soient, donc, un alphabet X, une fonction de semi-commutation f définie sur X^*, et u un mot de X^*. On peut supposer, sans nuire à la généralité des démonstrations que $X=alph(u)$ et $\forall\, x \in X$, $|u|_x=1$. Soit $\{X_1,X_2\}$, l'ensemble des composantes fortement connexes du graphe de non-commutation de X pour f. Dans tout ce qui suit, nous noterons $u_1=u\,\Pi_{X_1}$ et $u_2=u\,\Pi_{X_2}$. Nous avons vu, dans la preuve de la *proposition 3.5* , que: $\forall\, x \in X_1$, $\forall\, y \in X_2$, $xy \in yxf$, ou $\forall\, x \in X_1$, $\forall\, y \in X_2$, $yx \in xyf$. Envisageons d'abord le cas où ces propriétés sont toutes deux vérifiées, il est alors facile de montrer: $u^*f=(D_1^*(x_1,x_2)\,h^{-1}) \cap (\, u_1^*f \sqcup\!\sqcup u_2^*f\,)$, où x_1 est la première lettre de u_1, x_2 est la première lettre de u_2, et h est le morphisme de X^* vers $\{x_1,x_2\}^*$, défini par: $\forall\, x \in \{x_1,x_2\}$, $x\,h=x$, $\forall\, x \in X\text{-}\{x_1,x_2\}$, $x\,h=\varepsilon$. D'après la *proposition 3.6*, $u_1^*f \in RAT$ et $u_2^*f \in RAT$, on en déduit alors: $u^*f \in ALG$.

Dans l'autre cas, la démonstration sera plus compliquée, et, pour alléger les notations, nous introduisons la définition suivante: Nous dirons qu'un alphabet X, une fonction de semi-commutation f, et un mot u vérifient l'hypothèse (H) si: $\forall\, x \in X$, $|u|_x=1$, le graphe de non-commutation de X pour f comporte deux composantes fortement connexes: X_1 et X_2, $\forall\, x \in X_1$, $\forall\, y \in X_2$, $xy \in yxf$, et $\exists\, x \in X_1$, $\exists\, y \in X_2$ *tels que* $yx \notin xyf$. (Le cas où: $\forall\, x \in X_1$, $\forall\, y \in X_2$, $yx \in xyf$, et $\exists\, x \in X_1$, $\exists\, y \in X_2$ tels que $xy \notin yxf$, se traiterait de manière identique.) Nous allons tout d'abord donner une caractérisation des mots appartenant au langage $L=FG(u^*f)$ en utilisant deux résultats énoncés précédemment: le *lemme 2.1* et le *lemme 3.3* du chapitre 2. Ces deux lemmes vont nous permettre de caractériser le langage $L=FG(u^*f)$ en liant le nombre de lettres de X_1 et celui des lettres de X_2 dans les mots de L. Dans ce qui suit, pour tout $y \in X_2$, nous noterons: $X_y=\{x \in X_1$ tq $yx \notin xyf$ et x précède y dans $u\}$, et $X'_y=\{x \in X_1$ tq $yx \notin xyf$ et x précède y dans $u\}$.

Lemme 3.6: *Soient X, f et u vérifiant l'hypothèse (H). Alors, un mot w de X^* est un élément de $L=FG(u^*f)$ si et seulement si w vérifie les propriétés suivantes: $w\,\Pi_{X_1} \in FG(u_1^*f)$, $w\,\Pi_{X_2} \in FG(u_2^*f)$, et pour toute lettre y de X_2, pour tous mots t,t' de X^* tels que $w=tt'$, on a: $\forall\, x \in X_y$, $|t|_x \geq |t|_y$, $\forall\, x \in X'_y$, $|t|_x \geq |t|_y-1$.*

Preuve: Soit $w \in L=FG(u^* f)$. En utilisant le *lemme 3.3* du chapitre 2 (lemme de projection sélective), on déduit facilement: $w \Pi_{X_1} \in FG(u_1^* f)$, et $w \Pi_{X_2} \in FG(u_2^* f)$. D'autre part, soient $y \in X_2$, $x \in X_y$, $x' \in X_{\bar{y}}$. En utilisant de nouveau le *lemme 3.3* du chapitre 2, on déduit: $w \Pi_{xy} \in FG((xy)^* f)$, et $w \Pi_{x'y} \in FG((yx')^* f)$. Comme $yx \notin xyf$ et yx' $\notin$ x'y f, on en déduit alors: $\forall t,t' \in X^*$ *tq* $w=tt'$, $/t/_x \geq /t/_y$ *et* $/t/_{x'} \geq /t/_y - 1$.

Réciproquement, soit $w \in X^*$, vérifiant: $w \Pi_{X_1} \in FG(u_1^* f)$, $w \Pi_{X_2} \in FG(u_2^* f)$, et $\forall y \in X_2$, $\forall t,t' \in X^*$ *tq* $w=tt'$, $\forall x \in X_y$ $/t/_x \geq /t/_y$ et $\forall x \in X_{\bar{y}}$, $/t/_x \geq /t/_y - 1$. Soit $p= sup\{ /w/_x, x \in X \}$, alors $\exists w_1 \in X_1^*$, $\exists w_2 \in X_2^*$ tels que: $(w \Pi_{X_1}) w_1 \in u_1^p f$, et $(w \Pi_{X_2}) w_2 \in u_2^p f$. Posons $w'=w_1 w_2$. Nous allons montrer que $ww' \in u^p f$. D'après le *lemme 3.4*, il suffit de montrer: $\forall (x,y) \in XxX$, $ww' \Pi_{xy} \in (u^p \Pi_{xy}) f$. Clairement, il suffit de vérifier cette propriété pour $y \in X_2$ et $x \in X_y \cup X_{\bar{y}}$. Pour la démonstration, on supposera $x \in X_y$, et, de façon similaire, on pourrait vérifier la propriété dans le cas $x \in X_{\bar{y}}$. Avec ces hypothèses, comme $ww' \Pi_{xy} = (w \Pi_{xy})(w_1 \Pi_x)(w_2 \Pi_y)$, on déduit du *lemme 2.1*, $ww' \Pi_{xy} \in (xy)^p f$.

□

On peut déduire du résultat précédent:

Propriété 3.7: *Soient X, f, et u vérifiant l'hypothèse (H), soient* $w,v \in L=FG(u^* f)$, $x \in X$ *vérifiant:* $\exists p,n \in \mathbb{N}$ *tels que* $w \in u^p u_1^n v$ *com et* $wx \in L$, *alors:*

i) Si $x \in X_1$, $vx \in L$.

ii) Si $x \in X_2$, $(vx \in L)$ *ou* $(u_1 vx \in L$ *et* $n \neq 0)$.

Preuve: D'après le *lemme 3.1*, on déduit: $u^p u_1^n vx \in L$, donc $u_1^n vx \in L$ et, en application du *lemme 3.6*, $u_1^n vx \Pi_{X_1} \in FG(u_1^* f)$, et $u_1^n vx \Pi_{X_2} \in FG(u_2^* f)$. D'autre part, comme $v \in L$, on déduit également du *lemme 3.6*, $\forall y \in X_2$, $\forall z \in X_y$, $\forall z' \in X_{\bar{y}}$, $\forall t,t' \in X^*$ tels que $v=tt'$, $/t/_z \geq /t/_y$, $/t/_{z'} \geq /t/_y - 1$. Si $x \in X_1$, on en déduit donc immédiatement $vx \in L$. Si $x \in X_2$, supposons $vx \notin L$, alors, clairement, $n \neq 0$. De plus, de ce qui précède, on déduit: $\forall y \in X_2$, $\forall z \in X_y$, $\forall z' \in X_{\bar{y}}$, $\forall t,t' \in X^*$ *tels que* $tt'=u_1 v$, $/t/_z > /t/_y$, $/t/_{z'} > /t/_y - 1$, d'où $u_1 vx \in L$.

□

Les trois propriétés suivantes sont d'autres conséquences faciles du *lemme 3.6*.

Propriété 3.8: *Soient X, f, et u vérifiant l'hypothèse (H),soient $L=FG(u^* f)$ et $u_1=u\,\Pi_{X_1}$, alors:$uL \subset L$, et $u_1L \subset L$.*

Propriété 3.9: *Soient X, f, et u vérifiant l'hypothèse (H),alors:$\forall w \in L=FG(u^* f)$, $\forall p \in IN$, $[\ (\forall y \in X_2, |w|_y \geq p) \Rightarrow (\exists x \in X_1 \text{ tq } |w|_x \geq p-1)\ ]$.*

Propriété 3.10: *Soient X, f, et u vérifiant l'hypothèse (H),soit $w \in L=FG(u^* f)$, soient $p=sup\{ |w|_y, y \in X_2\}$ et $q=inf\{ |w|_x, x \in X_1\}$. Si $q>p$, le mot w', obtenu à partir de w en effaçant la $q^{ème}$ occurrence de chaque lettre de X_1 est un élément de L.*

Nous allons maintenant montrer que si X, f, et u vérifient l'hypothèse (H), alors $u^* f$ est algébrique, en construisant un automate à un compteur reconnaissant $u^* f$. Soit $M=(X, P, S, s_0, z_0, \delta, T)$, l'automate à pile défini par: $P=\{z_0, a\}$ est l'alphabet de pile, $S=\{ w \in FG(u^{8|X|^2} f)\}$ est l'ensemble des états, $s_0=\varepsilon$ est l'état initial, $T=\{s_0\}$ est l'ensemble des états terminaux, et δ , la fonction de transition, est définie par les quatre règles suivantes:

$\forall w \in S, \forall x \in X, \forall n \in IN$, $(w, x, z_0a^n) \vdash_1 (wx, \varepsilon, z_0a^n)$, *si* $wx \in S$.

$\forall n \in IN$, $\forall w \in S$ vérifiant: $\forall x \in X, |w|_x \geq 1$, $(w, \varepsilon, z_0a^n) \vdash_2 (w', \varepsilon, z_0a^n)$, où w' est obtenu à partie de w en effaçant la première occurrence de chaque lettre de X.

$\forall n \in IN$, $\forall w \in S$ vérifiant: $q=inf\{ |w|_x, x \in X_1\} \geq 1$, $(w,\varepsilon,z_0a^n) \vdash_3 (w',\varepsilon,z_0a^{n+1})$, si le mot w', obtenu à partir de w en effaçant la $q^{ème}$ occurrence de chaque lettre de X_1, est un élément de S.

$\forall w \in S, \forall n \in IN_+$, $(w, \varepsilon, z_0a^n) \vdash_4 (u_1w, \varepsilon, z_0a^{n-1})$, si $u_1w \in S$ où $u_1=u\,\Pi_{X_1}$.

Soit $T(M)=\{ w \in X^* \text{ tq } (\varepsilon, w, z_0) \vdash^* (\varepsilon, \varepsilon, z_0) \}$, le langage reconnu par M par pile vide. Afin de prouver l'égalité $T(M)=u^* f$, nous allons tout d'abord montrer:

Lemme 3.11: *Pour tout mot w de X*, pour tout état v de S, et pour tout entier n, si il existe une dérivation dans M permettant de passer de* (ε, w, z_0) *à* (v, ε, z_0a^n), *alors w est un élément de* $L=FG(u^* f)$ *et il existe un entier p vérifiant:* $w \in u^p u_1^n v$ *com.*

Preuve: Nous allons établir la preuve par induction sur la longueur de la dérivation: $(\varepsilon, w, z_0) \overset{*}{\vdash} (v, \varepsilon, z_0a^n)$. Le lemme est clairement vérifié pour une longueur de dérivation nulle: en effet, dans ce cas, $v=w=\varepsilon$ et $n=0$. Supposons, maintenant, le lemme vérifié pour une longueur de dérivation l, et considérons la dérivation: $(\varepsilon, w, z_0) \overset{l+1}{\vdash} (v, \varepsilon, z_0a^n)$. Il est clair que le lemme est vérifié si la dernière règle utilisée était la règle 2, 3 ou 4, sinon: $\exists x \in X, \exists v' \in S, \exists w' \in X^*$ tels que w=w'x, v=v'x et: $(\varepsilon, w'x, z_0) \overset{l}{\vdash} (v', x, z_0a^n) \vdash_1 (v'x, \varepsilon, z_0a^n)$. Par hypothèse de récurrence, $w' \in L$, et $\exists p \in IN$ tel que $w' \in u^p u_1^n v'$ *com*. On a donc bien $w'x \in u^p u_1^n v'x$ *com*. De plus, d'après la *propriété 3.8*, $u^p u_1^n v'x \in L$. Le *lemme 3.3* nous permet alors d'affirmer: $w'x \in L$.

□

Le résultat suivant est un peu le dual du *lemme 3.11*.

Lemme 3.12: *Pour tout mot w de* $L=FG(u^* f)$, *il existe un état v dans S et un entier n tels que* $(\varepsilon, w, z_0) \overset{*}{\vdash} (v, \varepsilon, z_0a^n)$.

Preuve: Nous allons faire une preuve par induction sur la longueur du mot w. Le lemme est vérifié, de façon évidente, pour $w=\varepsilon$. Supposons, maintenant, le lemme vérifié pour $w \in L$, et considérons $x \in X$ tel que $wx \in L$. Par hypothèse de récurrence, $\exists v \in S, \exists n \in IN$ tels que: $(\varepsilon, wx, z_0) \overset{*}{\vdash} (v, x, z_0a^n)$, et, d'après le *lemme 3.11*, $\exists p \in IN$ *tq* $w \in u^p u_1^n v$ *com*. Si on peut appliquer la règle 1, le lemme est alors vérifié, sinon envisageons deux cas:

1) $x \in X_1$.

Comme $wx \in L$, on déduit de la *propriété 3.7* que $vx \in L$, donc $|v|_x=8|X|^2$, puisque la règle 1 ne peut pas s'appliquer. Alors, d'après le *lemme 3.4*, $\forall x \in X_1$, $|v|_x \geq 4|X|$. Distinguons, maintenant, deux sous-cas:

1.1) La règle 2 peut s'appliquer.

Alors, $(v, x, z_0a^n) \vdash_2 (v', x, z_0a^n)$, et , comme v' est obtenu à partir de v en effaçant la première occurrence de chaque lettre de X, $|v'|_x=8|X|^2-1$, et la règle 1 peut maintenant s'appliquer: $(v', x, z_0a^n) \vdash_1 (v'x, \varepsilon, z_0a^n)$.

1.2) La règle 2 ne peut pas s'appliquer.

Alors, $\exists\, y \in X_2$ tel que $/v/_y = 0$, et, $\forall\, y \in X_2$, $/v/_y < 2/X/$ d'après le *lemme 3.4*. Comme $\forall\, x \in X_1$, $/v/_x \geq 4/X/$, on en déduit, grâce à la *propriété 3.10*, que, si $q = inf\{ /w/_x, x \in X_1\}$, le mot v', obtenu à partir de v en effaçant la $q^{ème}$ occurrence de chaque lettre de X_1 est un élément de S. On peut donc appliquer la règle 3: $(v, x, z_0a^n) \vdash_3 (v', x, z_0a^{n+1})$, et, comme $/v'/_x = 8/X/^2 - 1$, la règle 1 peut s'appliquer: $(v', x, z_0a^{n+1}) \vdash_1 (v'x, \varepsilon, z_0a^{n+1})$.

2) $x \in X_2$.

D'après la *propriété 3.7*, $(vx \in L)$ ou $(u_1vx \in L$ *et* $n \neq 0)$. Distinguons alors deux sous-cas:

2.1) $vx \in L$.

Comme on ne peut pas appliquer la règle 1, $/v/_x = 8/X/^2$. Grâce au *lemme 3.4*, on peut écrire: $\forall\, y \in X_2$, $/v/_y \geq 4/X/$, et, en application de la *propriété 3.9*: $\exists\, z \in X_1$ tel que $/v/_z \geq 4/X/ - 1$. Comme $/X/ \geq 1$, $/v/_z \geq 2/X/$, et, d'après le *lemme 3.1*, $\forall\, y \in X_1$, $/v/_y \geq 1$. La règle 2 peut donc s'appliquer: $(v, x, z_0a^n) \vdash_2 (v', x, z_0a^n)$, où v' est obtenu à partir de v en effaçant la première occurrence de chaque lettre de X. Maintenant, d'après le *lemme 3.2*, $v'x \in S$, et la règle 1 peut s'appliquer: $(u_1v', x, z_0a^{n-1}) \vdash_1 (u_1v'x, \varepsilon, z_0a^{n-1})$.

2.2) $vx \notin L$.

Alors, $u_1vx \in L$ et $n \neq 0$. Distinguons de nouveau deux sous-cas:

2.2.1) La règle 2 peut s'appliquer.

Alors, $(v, x, z_0a^n) \vdash_2 (v', x, z_0a^n)$, où v' est obtenu à partir de v en effaçant la première occurrence de chaque lettre de X. D'après la *propriété 3.7*, $u_1v'x \in L$. De plus, $\forall\, y \in X_1$, $/v'/_y = /v/_y - 1$ donc: $/v'/_y < 8/X/^2$. On peut donc appliquer la règle 4: $(v', x, z_0a^n) \vdash_4 (u_1v', x, z_0a^{n-1})$, et on se retrouve dans le cas 2.1.

2.2.2) La règle 2 ne peut pas s'appliquer.

Supposons qu'on ne puisse pas appliquer la règle 4, alors, $\exists\, y \in X_1$ *tel que* $/v/_y = 8/X/^2$, et d'après le *lemme 3.4*, $\forall\, z \in X_1$, $/v/_z \geq 4/X/$. De plus, comme la règle 2 ne peut pas s'appliquer, $\exists\, t \in X_2$ tel que $/v/_t = 0$. On déduit alors du *lemme 3.4*, $\forall\, z \in X_2$, $/v/_z < 2/X/$, et, d'après le *lemme 3.6*, $vx \in L$. Cette contradiction nous permet donc d'affirmer que la règle 4 peut s'appliquer: $(v, x, z_0a^n) \vdash_4 (u_1v, x, z_0a^{n-1})$, on se retrouve alors dans le cas 2.1, puisque $u_1vx \in L$, ce qui termine la démonstration.

□

Nous sommes maintenant en mesure de montrer $T(M)=u^* f$: Soit $w \in T(M)$, alors $(\varepsilon, w, z_0) \vdash^* (\varepsilon, \varepsilon, z_0)$, et, d'après le *lemme 3.11*, $w \in L=FG(u^* f)$ et $w \in u^p\ com$, donc $w \in L$. Réciproquement, soit $w \in u^* f$, alors, d'après le *lemme 3.12*, $\exists\, v \in S, \exists\, n \in \mathbb{N}$, tels que $(\varepsilon, w, z_0) \vdash^* (v, \varepsilon, z_0 a^n)$. De plus, d'après le *lemme 3.11*, $\exists p \in \mathbb{N}$ tel que $w \in u^p u_1^n v\ com$. Comme $w \in u^* f$, $\exists\, q \in \mathbb{N}$, tel que $w \in u^q f$, donc: $v \in (u_1^{q-(n+p)} u_2^{q-p})\ com$, où $u_2 = u \prod X_2$, et $8/X/^2 \geq q-p$. On peut donc appliquer n fois la règle 4, et on obtient: $(v, \varepsilon, z_0 a^n) \vdash^* (u_1^n v, \varepsilon, z_0)$, avec $u_1^n v \in (u_1^{q-p} u_2^{q-p})\ com$. Enfin, en appliquant $q-p$ fois la règle 2, on a bien: $(u_1^n v, \varepsilon, z_0) \vdash^* (\varepsilon, \varepsilon, z_0)$.

□

Cette dernière démonstration nous a donc permis d'établir que si le graphe de non-commutation de f pour $alph(u)$ comportait deux composantes fortement connexes, alors $u^* f \in ALG$. Nous pouvons donc énoncer:

Proposition 3.13: *Soient X un alphabet, f une fonction de semi-commutation définie sur X^*, et u, un mot de X^*. $L=u^* f$ est un langage algébrique si et seulement si le graphe de non-commutation de f pour alph(u) comporte au plus deux composantes fortement connexes.*

On peut remarquer que, si une condition simple permet de caractériser les fonctions de semi-commutation f et les mots u vérifiant: $u^* f \in ALG$, il n'en va pas de même si on considère, au lieu d'un seul mot, un langage, même fini et très simple, comme le montre l'exemple ci-dessous:

Soient $X=\{a,b,c\}$, et f, la fonction de semi-commutation associée aux règles: $ba \rightarrow ab$, $ac \rightarrow ca$, $bc \rightarrow cb$. Soit $F=\{ab, ca\}$, on peut vérifier facilement que: $F^* f \cap a^*c^*b^*a^* = ((ab)^* (ca)^*) f \cap a^*c^*b^*a^* = \{ a^p c^q b^p a^q , p,q \in \mathbb{N} \}$, qui n'est pas un langage algébrique.

Chapitre 5: Automates asynchrones.

1. Définitions et propriétés de base.

2 Automates virtuellement asynchrones.

3 Automates 2-asynchrones.

1. Définitions et propriétés de base.

Nous donnons ici une définition équivalente à celle donnée initialement par Zielonka:

Définition 1.1: *Un automate fini asynchrone sur une famille* $F=\{X_1,X_2,\ldots,X_n\}$ *d'alphabets, appelé encore F-asynchrone, est un 6-uple* $M=(F,H,S,*,s_0,T)$ *où: H est un ensemble fini,* $S=\{s:F \rightarrow H\}$ *est l'ensemble des états,* $s_0 \in S$ *est l'état initial,* $T \subset S$ *est l'ensemble des états terminaux, et* $*$ *désigne la fonction de transition.*

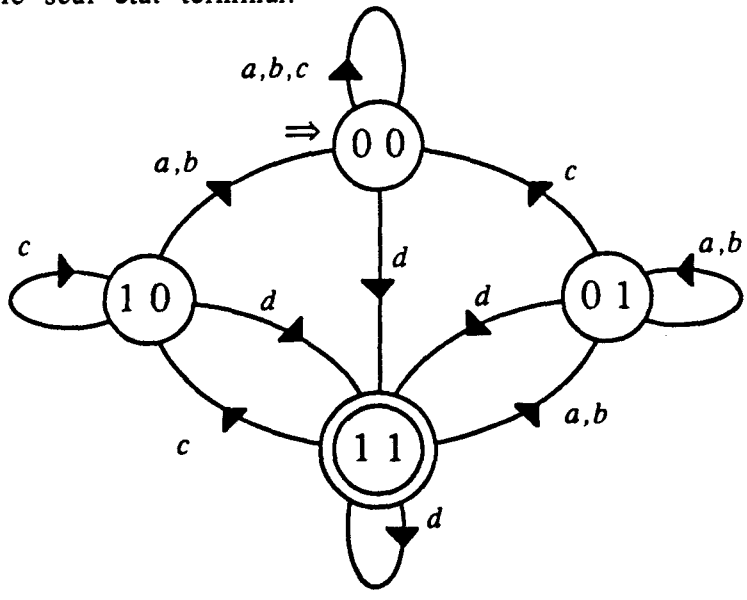
A tout x de $X=\bigcup_{i=1}^{n} X_i$ est associée une application $\delta_x: S_x \rightarrow S_x$ où $S_x=\{s:F_x \rightarrow H\}$, F_x désignant la famille $\{X_i \in F / x \in X_i\}$. On peut alors définir la fonction de transition: $\forall x \in X$, $\forall s \in S$, $s'=s*x$ est déterminé par:

$\forall X_i \in F\text{-}F_x$, $X_i s'=X_i s$,

$\forall X_i \in F_x$, $X_i s'=X_i (s_x \delta_x)$ où s_x désigne la restriction de s à F_x.

On étend alors, de façon usuelle, la fonction de transition aux mots, ce qui permet de définir le langage reconnu par cet automate, $L(M)=\{u \in X^* / s_0*u \in T\}$.

Exemple 1.2: Construisons un automate fini asynchrone sur la famille $\{X_1,X_2\}$ avec $X_1=\{a,b,d\}$, $X_2=\{c,d\}$, pour le langage rationnel $R=(a+b+c+d)^*d$. Ici, nous prendrons $H=\{0,1\}$ et un état s sera représenté par un couple dont la première composante est l'image de X_1 et la seconde est l'image de X_2. On prend $\delta_a=\delta_b=\delta_c$ avec $0\ \delta_a=0$, $1\ \delta_a=0$, a et b "travaillant" sur la première composante et c sur la seconde. La lettre d "travaille" sur les deux composantes et δ_d est donné par $00\ \delta_d=01\ \delta_d=10\ \delta_d=11\ \delta_d=11$. On peut alors représenter l'automate où 00 est l'état initial et 11 est le seul état terminal.



Remarque: Trivialement, tout automate fini sur un alphabet X peut être considéré comme asynchrone sur la famille $F=\{X\}$.

Si $M=(F,H,S,*,s_0,T)$ est un automate asynchrone sur une famille d'alphabets $F=\{X_1,X_2,...,X_n\}$, nous dirons alors que M est F-asynchrone.

Considérons deux lettres x et y de $X=\bigcup_{i=1}^{n} X_i$, telles que $F_x \cap F_y=\varnothing$. Alors: $\forall\, s \in S,\ s*xy= s*yx$. En effet, considérons $X_i \in F$. Si X_i ne contient ni x ni y, alors, par définition, $X_i\,(s*xy)=X_i\,(s*yx)=X_i\,s$. Si X_i contient x, alors, $X_i\,(s*xy)=X_i\,(s*x)=X_i\,(s*yx)$, puisque, par hypothèse, X_i ne contient pas y. Enfin, si X_i contient y, symétriquement, nous obtenons: $X_i\,(s*xy) = X_i\,(s*y) = X_i\,(s*yx)$. Soient, maintenant, la relation de commutation partielle C_F, définie sur X par: $\forall\,(x,y) \in X\times X$, $(x,y) \in C_F$, si et seulement si $F_x \cap F_y=\varnothing$, et g_F, la fonction de commutation partielle associée à C_F.. Nous dirons que g_F est la fonction de commutation partielle induite par F. De ce qui précède, on déduit: $\forall\, u \in X^*$, $\forall\, v \in u\,g_F$, $\forall\, s \in S$, $s*u=s*v$, et nous pouvons énoncer:

Lemme 1.3: *Soit M un automate F-asynchrone. Alors $L(M)$, le langage reconnu par M, est fermé pour la fonction de commutation partielle g_F induite par F.*

Nous terminerons cette section d'introduction avec le résultat fondamental de W. Zielonka qui a prouvé dans [36] la difficile réciproque du lemme précédent:

Théorème 1.4: *Soient f une fonction de commutation partielle définie sur un alphabet X, et $R \subset X^*$, un langage rationnel fermé par f. Alors il existe un automate F-asynchrone M qui reconnait R et tel que la fonction de commutation induite par F soit f.*

2. Automates virtuellement asynchrones.

Clairement, il est toujours possible, par un simple renommage des états, de considérer un automate asynchrone sur une famille d'alphabet comme un automate fini habituel. Réciproquement, cela n'est évidemment pas le cas, aussi introduisons-nous la notion d'automate **virtuellement asynchrone** sur une famille d'alphabets. Plus précisément:

Définition 2.1: *Soient* $M=(X,Q,\otimes,q_0,E)$ *un automate fini déterministe et* $F=\{X_1,X_2,\ldots,X_n\}$ *une famille d'alphabets vérifiant* $X=\bigcup_{i=1}^{n} X_i$. *L'automate M est virtuellement asynchrone sur la famille F, si il existe un automate F-asynchrone* $M'=(F,H,S,*,s_0,T)$ *qui lui est isomorphe.*

Dans cette section, nous allons nous donner les moyens de décider quand un automate est virtuellement asynchrone sur une famille d'alphabets. Soient, donc, un automate fini déterministe $M=(X,Q,\otimes,q_0,E)$ et une famille d'alphabets $F=\{X_1,X_2,\ldots,X_n\}$ vérifiant $X=\bigcup_{i=1}^{n} X_i$. Posons $\underset{1}{\equiv},\underset{2}{\equiv},\ldots,\underset{n}{\equiv}$, les plus fines relations d'équivalences, définies sur Q et vérifiant les deux propriétés suivantes:

P_1: $\forall i \in [1,n]$, $\forall q \in Q$, $\forall x \in X\text{-}X_i$, $q \underset{i}{\equiv} q\otimes x$.

P_2: $\forall p,q \in Q$, $\forall x \in X$, $(\forall X_i \in F_x, p \underset{i}{\equiv} q) \Rightarrow (\forall X_i \in F_x, p\otimes x \underset{i}{\equiv} q\otimes x)$.

Nous allons tout d'abord montrer qu'on peut effectivement construire ces relations: Posons $B_0=\{(p,X_k,q)$ *où* $p,q \in Q, X_k \in F$ *tels que* $\exists\, q_0,q_1,\ldots,q_t \in Q$ *tels que* $p=q_0$, $q=q_t$ *et* $\forall i \in [1,t\text{-}1]$, $\exists x \in X\text{-}X_k$ / $(q_i=q_{i+1}\otimes x)$ *ou* $(q_{i+1}=q_i\otimes x)\}$. Définissons la suite $B_{(i\in\mathbb{N})}$ par: $B_{i+1}=B_i \cup \{(p,X_k,q)$ / $p,q \in Q, X_k \in F$ *tels que:* $\exists x \in X_k$, $\exists p',q' \in Q$ *vérifiant:* $(p'\otimes x,X_k,p) \in B_i$, $(q'\otimes x,X_k,q) \in B_i$, *et* $\forall X_j \in F_x$, $(p',X_j,q') \in B_i\}$. Posons $B=\bigcup_{i\in\mathbb{N}} B_i$. Clairement, d'après la construction de l'ensemble B: $\forall p,q \in Q$, $\forall X_k \in F$, $((p,X_k,q) \in B) \Leftrightarrow ((q,X_k,p) \in B)$. D'autre part, comme $B_{(i\in\mathbb{N})}$ est une suite croissante majorée par l'ensemble fini $Q\times F\times Q$, il existe un entier m tel que $B_m=B_{m+1}=B$.

Nous allons montrer que nous avons bien construit les relations $\underset{1}{\equiv},\underset{2}{\equiv},\ldots,\underset{n}{\equiv}$. Plus précisément, soient $\underset{1}{\#},\underset{2}{\#},\ldots,\underset{n}{\#}$ les relations définies sur Q par: $\forall i \in [1,n]$, $\forall p,q \in Q$, $p \underset{i}{\#} q$ si et seulement si $(p,X_i,q) \in B$. Vérifions qu'il s'agit bien de relations d'équivalence. Nous avons vu que ces relations étaient réflexives et symétriques, il nous reste donc à montrer qu'elles sont également transitives. Nous allons pour cela utiliser les deux lemmes suivants.

Lemme 2.2: $\forall\, q \in Q,\ \forall\, X_k \in F,\ (\exists\, p \in Q \,/\, (p,X_k,q) \in B \text{ et } (p,X_k,q) \notin B_0)) \Rightarrow (\exists\, r \in Q,\ \exists\, x \in X_k \,/\, (r \otimes x, X_k, q) \in B_0)$.

Preuve: Soit i_0, le plus petit i vérifiant: $\exists\, p \in Q \,/\, (p,X_k,q) \in B_i$ et $(p,X_k,q) \notin B_0$. Comme $(p,X_k,q) \notin B_{i_0-1}$, $\exists\, p',q' \in Q$, $\exists\, x \in X_k$ vérifiant: $(p' \otimes x, X_k, p) \in B_{i_0-1}$, $(q' \otimes x, X_k, q) \in B_{i_0-1}$, et $\forall\, X_j \in F_X$, $(p', X_j, q') \in B_{i_0-1}$. Et par définition de i_0, on en déduit que $(q' \otimes x, X_k, q) \in B_0$.

□

Lemme 2.3: $\forall\, p,q,r \in Q,\ \forall\, X_k \in F,\ \forall\, i \in \mathbb{N},\ ((p,X_k,q) \in B_i \text{ et } (q,X_k,r) \in B_0) \Rightarrow ((p,X_k,r) \in B_i)$.

Preuve: La preuve s'appuie sur une récurrence portant sur i. Le résultat est évident pour $i=0$. Supposons le lemme vérifié jusque $i>0$ et considérons $p,q,r \in Q$ tels que $(p,X_k,q) \in B_{i+1}$ et $(q,X_k,r) \in B_0$. Si $(p,X_k,q) \in B_i$, alors le lemme est vérifié, sinon: $\exists\, x \in X_k$, $\exists\, p',q' \in Q$ tels que $(p' \otimes x, X_k, p) \in B_i$, $(q' \otimes x, X_k, q) \in B_i$, et $\forall\, X_j \in F_X$, $(p', X_j, q') \in B_i$. Comme le lemme est vérifié pour i, on en déduit que $(q' \otimes x, X_k, r) \in B_i$ et $(p,X_k,r) \in B_{i+1}$.

□

Nous sommes maintenant en mesure de montrer:

Lemme 2.4: $\forall\, p,q,r \in Q,\ \forall\, i \in [1,n],\ (p \overset{\#}{_i} q \text{ et } q \overset{\#}{_i} r) \Rightarrow (p \overset{\#}{_i} r)$

Preuve: Si $p \overset{\#}{_i} q$ et $q \overset{\#}{_i} r$, alors $(p,X_i,q) \in B$ et $(q,X_i,r) \in B$. Si $(p,X_i,q) \in B_0$ ou $(q,X_i,r) \in B_0$, alors, d'après le *lemme 2.3*, $(p,X_i,r) \in B$. Sinon, d'après le *lemme 2.2*, $\exists\, s \in Q$, $\exists\, x \in X_i$ tels que $(s \otimes x, X_i, q) \in B_0$. Comme $(p,X_i,q) \in B_m$ et $(q,X_i,r) \in B_m$, on déduit du *lemme 2.3*: $\exists\, s \in Q$, $\exists\, x \in X_i$ tels que $(s \otimes x, X_i, p) \in B_m$ et $(s \otimes x, X_i, r) \in B_m$. Comme de plus, $\forall\, X_j \in F_X$, $(s, X_j, s) \in B_m$, on obtient $(p,X_i,r) \in B_{m+1} = B_m = B$.

□

Les relations $\overset{\#}{_1}$, $\overset{\#}{_2}$,..., $\overset{\#}{_n}$ sont donc bien des relations d'équivalence. Clairement, d'après la construction de l'ensemble B, ces relations vérifient les propriétés P_1 et P_2. Comme les relations $\overset{\equiv}{_1}$, $\overset{\equiv}{_2}$,..., $\overset{\equiv}{_n}$ sont les plus fines relations d'équivalence vérifiant ces deux propriétés, on en déduit: $\forall\, p,q \in Q$, $\forall\, i \in [1,n]$, $p \overset{\equiv}{_i} q \Rightarrow p \overset{\#}{_i} q$. Réciproquement, on démontre très facilement, par induction sur le plus petit k tel que $(p,X_i,q) \in B_k$: $\forall\, p,q \in Q$, $\forall\, i \in [1,n]$, $p \overset{\#}{_i} q \Rightarrow p \overset{\equiv}{_i} q$. On a donc bien construit les relations $\overset{\equiv}{_1}$, $\overset{\equiv}{_2}$,..., $\overset{\equiv}{_n}$. En notant la classe d'équivalence d'un état p pour la relation d'équivalence $\overset{\equiv}{_i}$ par $[p]_i$, les propriétés P_1 et P_2 s'écrivent:

Propriété 2.5: $\forall q \in Q, \forall i \in [1,n], \forall x \in X\text{-}X_i, [p]_i=[p\otimes x]_i$.

Propriété 2.6: $\forall x \in X, \forall p,q \in Q, (\forall X_i \in F_x, [p]_i=[q]_i) \Rightarrow (\forall X_i \in F_x, [p\otimes x]_i=[q\otimes x]_i)$

La proposition suivante donne une relation liant les états d'un automate asynchrone M et les classes d'équivalence d'un automate fini "classique" isomorphe à M.

Proposition 2.7: *Soient* $M=(F,H,S,*,s_0,T_n)$ *un automate asynchrone sur une famille d'alphabets* $F=\{X_1,X_2,\ldots,X_n\}$ *vérifiant* $X=\bigcup_{i=1}^{n} X_i$, *et* $M'=(X,Q,\otimes,q_0,E)$ *un automate fini déterministe vérifiant* $M'=M\,\phi$, *où* ϕ *est un isomorphisme d'automate. Alors:* $\forall s,s' \in S, \forall i \in [1,n], ([s\,\phi]_i=[s'\,\phi]_i) \Rightarrow (X_i\,s=X_i\,s')$.

Preuve: La preuve de cette proposition s'appuie sur la construction de l'ensemble B. Soient $p=s\,\phi, q=s'\,\phi \in Q$, $i \in [1,n]$, tels que $[p]_i=[q]_i$. Soit t le plus petit entier tel que $(p,X_i,q) \in B_t$. Si $t=0$, $\exists\, r_1,r_2,\ldots,r_k \in Q$ tels que : $p=r_1$, $q=r_k$, et $\forall j \in [1,k-1]$, $\exists\, x \in X\text{-}X_i$ vérifiant $q_j=q_{j+1}\otimes x$ ou $q_{j+1}=q_j\otimes x$. Comme M est asynchrone, on déduit: $\forall j \in [1,k-1]$, $X_i.(q_j\,\phi^{-1})=X_i\,(q_{j+1}\,\phi^{-1})$, donc $X_i\,s=X_i\,s'$.

Supposons, maintenant, la proposition vérifiée jusque $t>0$ et soient $p=s\,\phi, q=s'\,\phi \in Q$, vérifiant $(p,X_i,q) \in B_{t+1}$ et $(p,X_i,q) \notin B_t$. Alors, il existe des états p' et q' dans Q, et une lettre y de X_i vérifiant: $\forall X_j \in F_y$, $(p',X_j,q') \in B_t$, $(p'\otimes y,X_i,p) \in B_t$, $(q'\otimes y,X_i,q) \in B_t$. En appliquant l'hypothèse de récurrence, on déduit $X_i\,((p'\,\phi^{-1})*y)=X_i\,s$ et $X_i\,((q'\,\phi^{-1})*y)=X_i\,s'$. De plus, comme M est asynchrone: $X_i\,((p'\,\phi^{-1})*y)=X_i\,((q'\,\phi^{-1})*y)$ donc $X_i\,s=X_i\,s'$.

□

Nous pouvons maintenant énoncer le résultat principal de cette section:

Proposition 2.8: *Soient* $M=(X,Q,\otimes,q_0,E)$ *un automate fini déterministe, complètement spécifié, et* $F=\{X_1,X_2,\ldots,X_n\}$ *une famille d'alphabets vérifiant:* $X=\bigcup_{i=1}^{n} X_i$. *Alors M est virtuellement F-asynchrone si et seulement si:* $\forall p\neq q \in Q, \exists\, i \in [1,n]$ / $[p]_i\neq[q]_i$.

En outre, quand les conditions de cette proposition sont remplies, l'automate $M'=(F,H,S,*,s_0,T)=M\,\phi$ où $\phi: Q \rightarrow S$ est l'isomorphisme d'automate défini par: $\forall q \in Q$, $s=q.\phi$ est déterminé par: $\forall i \in [1,n]$, $X_i\,s=[q]_i$, est un automate F-asynchrone isomorphe à M.

Preuve: Nous allons montrer que la condition est suffisante en vérifiant que l'automate M' défini ci-dessus est bien F-asynchrone.

- Soient $s \in S, x \in X, X_i \in F_x$. Posons $q=s\,\phi^{-1}$, alors, $X_i\,s=[q]_i$ et $X_i\,(s*x)=[q\otimes x]_i$. D'après la *propriété 2.5*, on déduit donc: $X_i\,(s*x)=X_i\,s$.

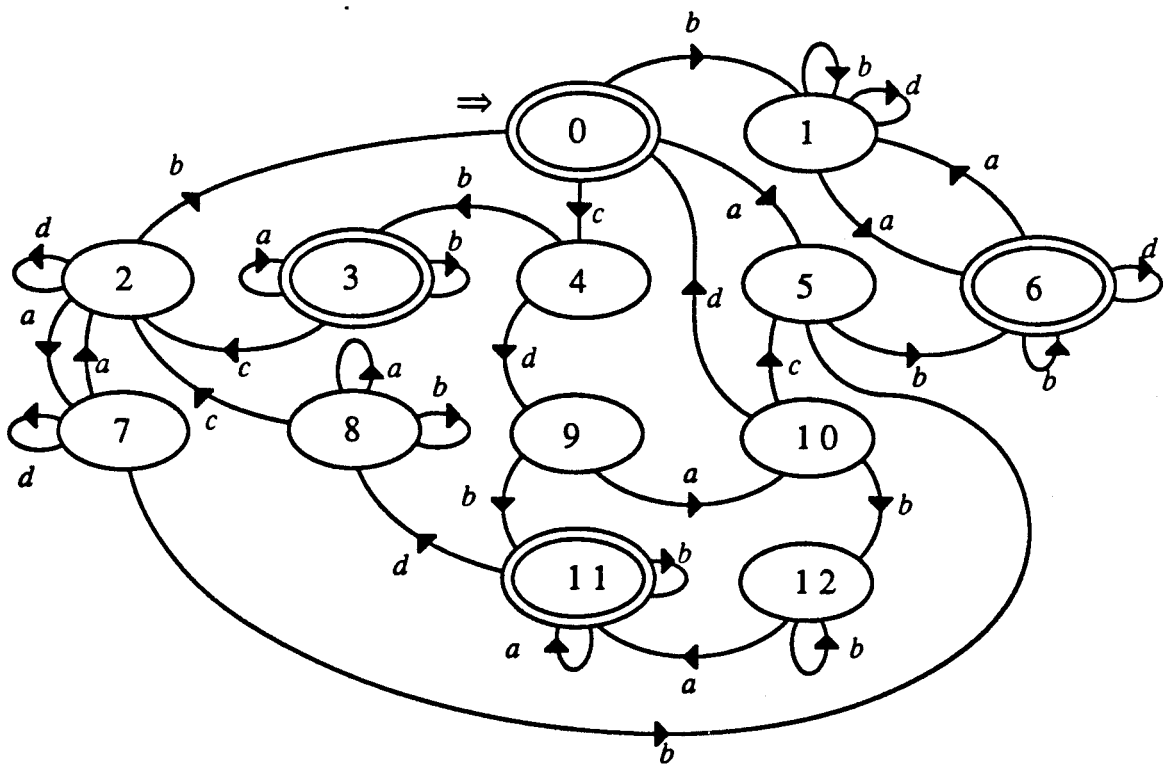
- Soient $s,s' \in S, x \in X$, vérifiant $\forall\, X_i \in F_x, X_i\,s=X_i.s'$. Posons $q=s\,\phi^{-1}$ et $q'=s'\,\phi^{-1}$. Alors, $\forall\, X_i \in F_x$, $[q]_i=[q']_i$ et d'après la *propriété 2.6* on déduit: $\forall\, X_i \in F_x$, $X_i\,(s*x)=X_i\,(s'*x)$. L'automate M' est donc bien F-asynchrone.

Réciproquement, supposons qu'il existe un automate F-asynchrone $M''=(F,D,V,\oplus,v_0,Z)$, vérifiant: $M''=M\,\varphi$ est isomorphe à M et $\exists\, q\neq q' \in Q$ tels que $\forall\, i \in [1,n]$, $[q]_i=[q']_i$. Soient $v=q\,\varphi$ et $v'=q'\,\varphi$. Comme $v\neq v'$, il existe $j \in [1,n]$ tel que $X_j\,v \neq X_j\,v'$. D'après la *proposition 2.7* ceci entraine $[q]_j \neq [q']_j$. Cette contradiction avec les hypothèses termine donc la preuve de la proposition.

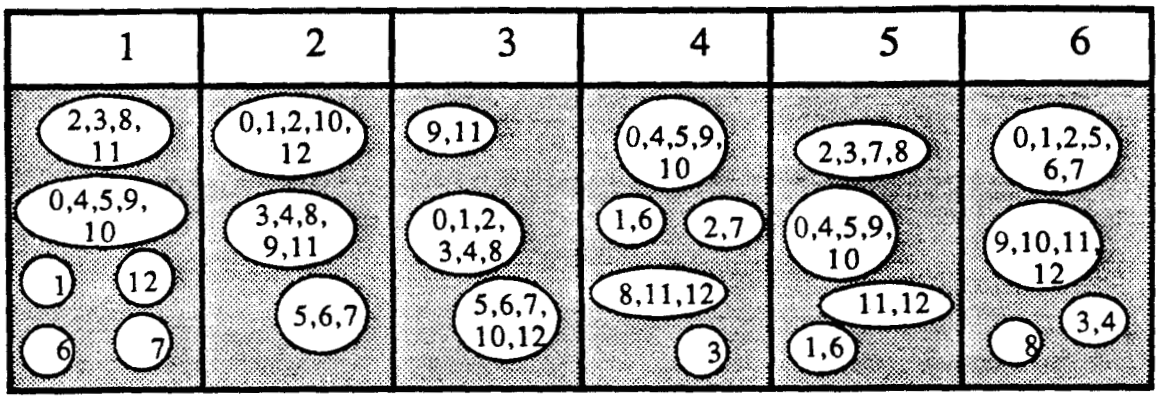
□

Nous allons appliquer ce résultat sur un exemple:

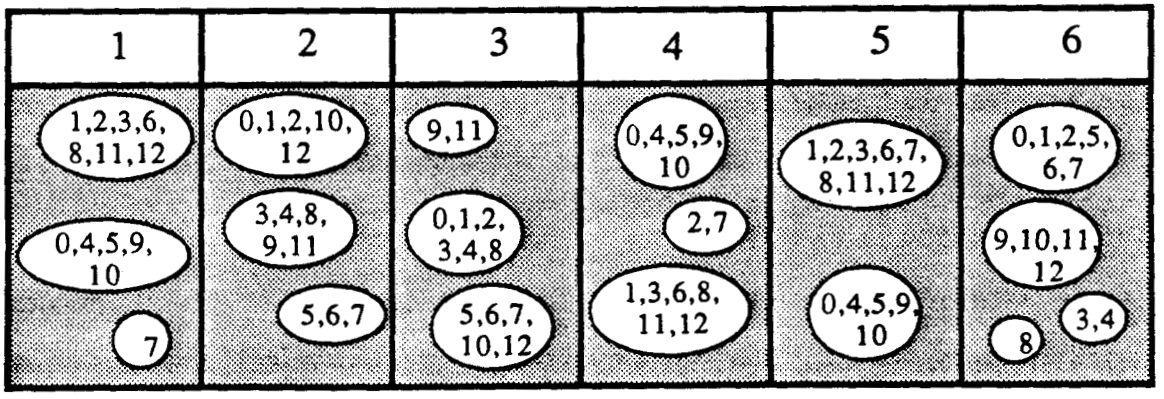
Exemple 2.9: Examinons si l'automate suivant, admettant comme alphabet d'entrée $X=\{a,b,c,d\}$ est virtuellement asynchrone sur $F=\{X_1,X_2,X_3,X_4,X_5,X_6\}$, où $X_1=\{a,b\}$, $X_2=\{a,c\}$, $X_3=\{a,d\}$, $X_4=\{b,c\}$, $X_5=\{b,d\}$ et $X_6=\{c,d\}$.



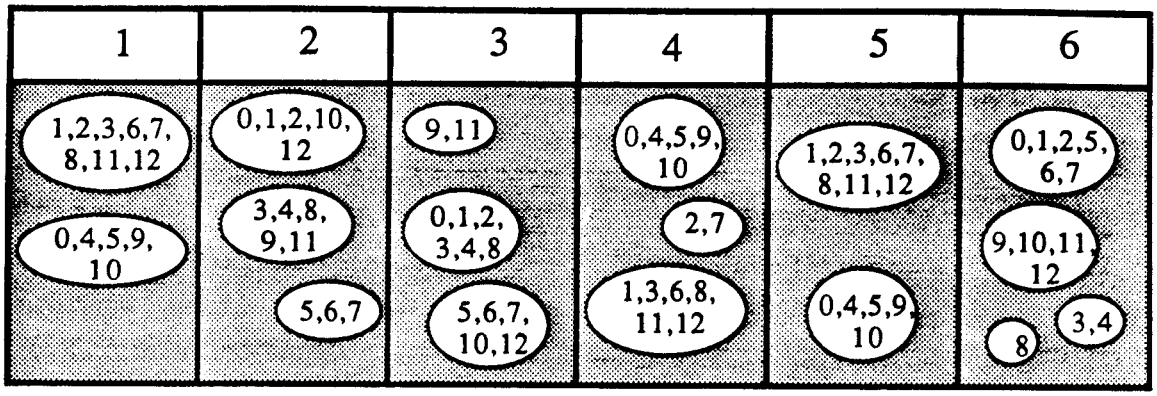
Calculons les différentes classes d'équivalence dans ce cas Pour plus de lisibilité, nous effectuerons les fermetures transitives au fur et à mesure, alors que dans l'algorithme réel ces fermetures sont "retardées". Aussi présenterons-nous les étapes successives de calcul sous formes de partitions de l'ensemble des états de l'automate pour chacune des six composantes, ces partitions convergeant vers les classes d'équivalences souhaitées. Nous obtenons ainsi dans un premier temps:



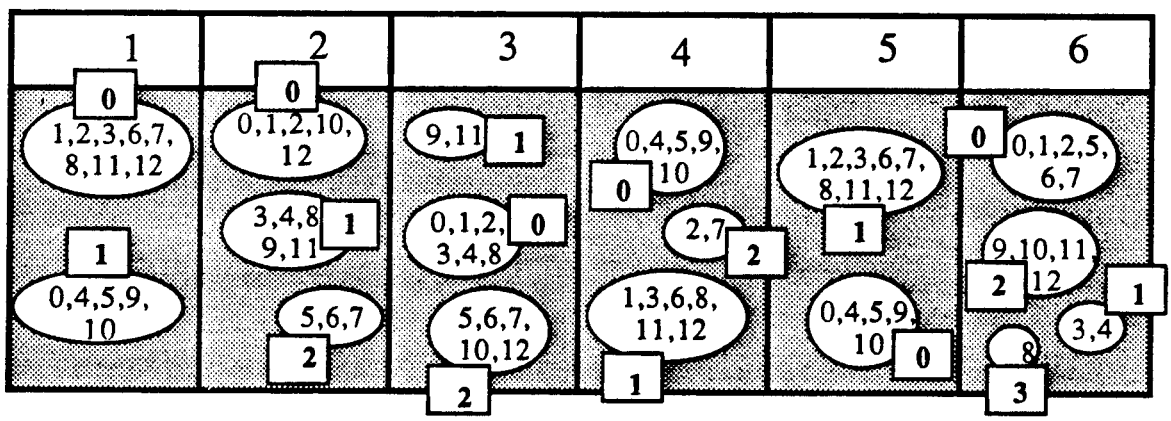
Comme les états 0,4,5,9 et 10 sont dans un même élément de partition pour les trois composantes concernant la lettre *b*, nous devons réunir dans un même élément de partition, pour chacune de ces trois composantes, les éléments de partition contenant les états 1,3,6,11,12, ateints à partir des états 0,4,5,9,10 en lisant la lettre *b*. Le même cas se produit pour les états 3,8 avec la lettre *a*, cependant, comme on boucle sur chacun de ces états en lisant *a*, ce cas n'apporte pas de changement et on obtient:

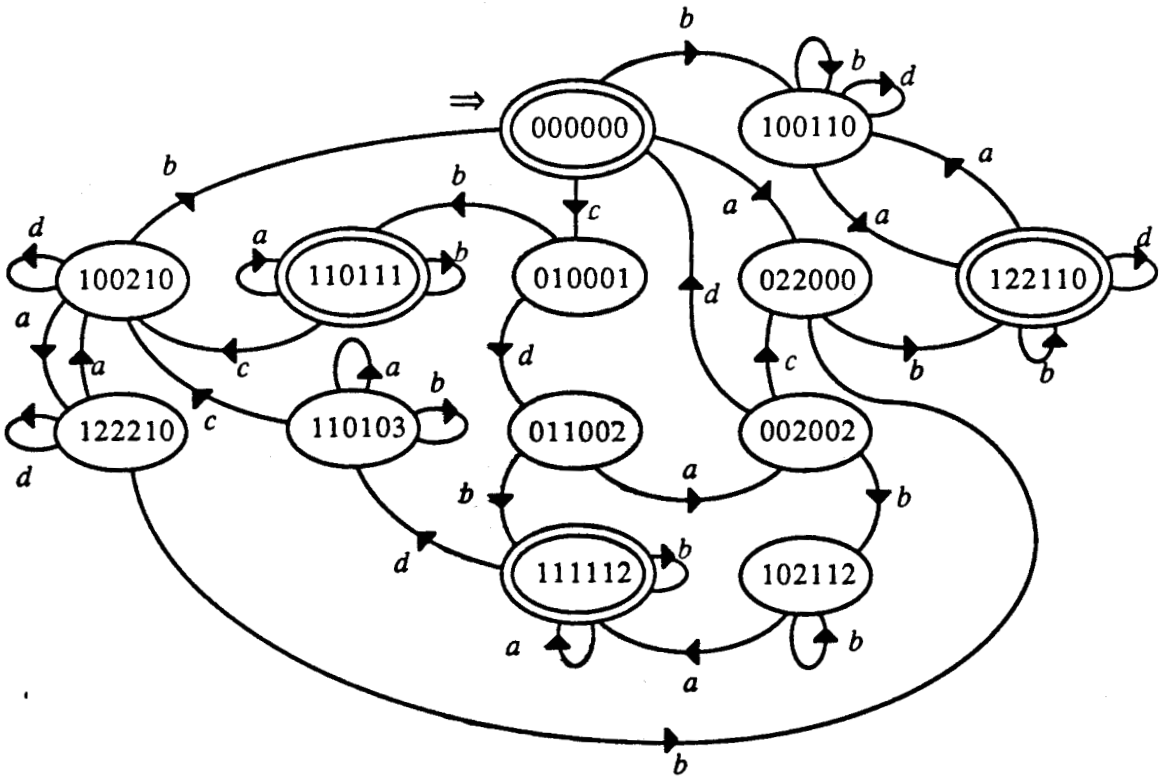


Les états 1 et 2 sont maintenant réunis dans les trois composantes concernant la lettre *a*. Le même cas se produit pour les états 1,3,6,8,11,12 avec la lettre *b*, et pour les états 1 et 2 d'une part, 6 et 7 d'autre part avec la lettre *d*. On aboutit alors aux partitions suivantes:



On constate alors qu'un nouveau calcul ne change plus ces partitions. Comme il n'existe pas deux états différents qui soient dans une même partition pour les six composantes, on en déduit que l'automate considéré est bien virtuellement asynchrone sur la famille d'alphabets donnée. En renommant les différentes classes d'equivalence comme indiqué sur le tableau suivant, on obtient alors l'automate asynchrone correspondant.





3. Automates 2-asynchrones.

L'automate asynchrone construit dans la preuve du *théorème 1.4* (voir [34]) pour une relation ou une fonction de commutation partielle f donnée est un automate asynchrone sur la famille des cliques maximales du graphe de non-commutation de la fonction f. La famille des automates 2-asynchrones correspond en quelque sorte à la famille des automates asynchrones où l'information est la plus répartie possible:

Définition 3.1: *On appelle automate 2-asynchrone tout automate asynchrone sur une famille d'alphabets qui ont une cardinalité inférieure ou égale à 2.*

La famille des automates 2-asynchrones a en fait la même puissance que la famille des automates asynchrones quelconques comme le montre la proposition suivante:

Proposition 3.2: [16] *Pour tout automate M asynchrone sur une famille F, il existe un automate M' 2-asynchrone sur une famille F', reconnaissant le même langage, et tel que la fonction de commutation partielle induite par F' soit égale à la fonction de commutation partielle induite par F.*

La preuve de cette proposition s'appuie sur la construction d'un automate 2-asynchrone qui, pour un alphabet X, reconnait la dernière lettre lue pour tous les mots de X^*. La construction de cet automate est elle même basée sur celle d'un automate 2-asynchrone reconnaissant la dernière lettre des mots construits sur un alphabet de trois lettres. La *proposition 3.2* et le *théorème 1.4* permettent alors de conclure:

Proposition 3.3: [16] *Soient f une fonction de commutation partielle définie sur un alphabet X et $R \subset X^*$ un rationnel fermé par f, alors il existe un automate 2-asynchrone sur une famille F, reconnaissant R, tel que la fonction de commutation induite par F soit égale à f.*

Dans le cas où la fonction f est la fonction où rien ne commute, on obtient alors le résultat suivant:

Corrollaire 3.4: [16] *Tout rationnel R défini sur un alphabet X peut être reconnu par un automate asynchrone sur la famille $\{\{x,y\}, x,y \in X, x \neq y\}$.*

La construction utilisée dans la preuve de la *proposition 3.2* permet d'obtenir un automate 2-asynchrone reconnaissant un rationnel quelconque, à partir d'un automate "classique" qui reconnait ce rationnel. Nous proposons ici une preuve différente du *corrollaire 3.4*.

Remarquons tout d'abord que, pour un rationnel défini sur un alphabet de 2 lettres, tout automate fini reconnaissant ce rationnel convient. Nous considérerons donc un alphabet $A=(0,1,..,n-1)$ avec $n\geq 3$. Soit le langage rationnel $R \subset A^*$ reconnu par un automate fini, déterministe, complètement spécifié $M=(A,Q,*,q_0,S)$. Posons $F=\{A_{ij}=\{i,j\} / i,j \in A, i\neq j\}$ et pour tout i de A, $F_i=\{A_{ik}/k\neq i\}$. Soit m le plus petit entier supérieur ou égal à n tel que m ne soit pas divisible par $m-n+3$. Posons $M=[0,...,m-1]$ et $P=MxQxM$. Notons Π_1 et Π_3 les applications de P vers M définies par: $\forall\ p=(x_1,x_2,x_3) \in P$, $p\,\Pi_1=x_1$, et $p\,\Pi_3=x_3$, et Π_2, l'application de P vers Q définie par: $\forall\ p=(x_1,x_2,x_3) \in P$, $p\,\Pi_2=x_2$. Considérons l'automate 2-asynchrone $M'=(F,P,V,\oplus,v_0,T)$, et pour tout i de A, pour tout v de V, notons v_i, la restriction de v à F_i et S_i, l'application de v dans N définie par: $\forall\ v \in V$, $v\,S_i = \sum_{A_{ij}\in F_i} ((A_{ij}\ v)\ \Pi_1)$.

Remarquons que si $v' \in V$ avec $v'_i=v_i$, alors $v'\,s_i=v\,S_i$, donc $v\,S_i$ peut être calculé à partir de v_i. Nous allons définir l'état initial v_0 et la fonction de transition de M' de manière à ce que, pour tout état accessible v dans M', toute lettre i de A puisse retrouver l'état atteint dans M par l'intermédiaire de la connaissance de la dernière lettre lue. Plus précisément, v_0 et la fonction de transition de M' vont nous permettre d'établir, par induction sur $u \in A^*$, que $v=v_0\oplus u$ vérifie la propriété suivante notée (P):

(P_1): il existe un et un seul i de A, noté $lettre(v)$ tel que pour tout k différent de i, $((A_{ik}\ v)\ \Pi_3)=v\,S_i$ (modulo m).

(P_2): pour tout k différent de $lettre(v)$, $lettre(v)=v\,S_k$ (modulo m).

(P_3): il existe e dans Q, noté $état(v)$ tel que $q_0*u=e$ et pour tout k différent de i, $((A_{ik}\ v)\ \Pi_2)=e$.

Définissons v_0 par: $\forall\ k\neq 1 \in A$, $(A_{k1}\ v_0)=(1,q_0,n-1)$ et $\forall\ k\neq 1 \in A$, $\forall\ j\neq 1 \in A$, $(A_{jk}\ v_0)=(0,q_0,0)$. Clairement, v_0 vérifie la propriété (P) avec $lettre(v_0)=1$ et $état(v_0)=q_0$. Soit $v=v_0*u$ vérifiant (P) et $v'=v\oplus i$. L'application v_i permet alors de retrouver $lettre(v)$ et $état(v)$. En effet, si pour tout k différent de i, $((A_{ik}\ v_i)\ \Pi_3)=v\,S_i$ (modulo m), alors $i=lettre(v)$ et pour tout tout k différent de i, $((A_{ik}\ v_i)\ \Pi_2)=état(v)$. Sinon $lettre(v)=v\,S_i$ (modulo m) et $état(v)=((A_{ij}\ v)\ \Pi_2)$ où $j=lettre(v)$. Pour définir $v'_i=v_i\ \delta_i$, distinguons deux cas:

Supposons $i=lettre(v)$. Définissons dans ce cas v'_i par: pour tout k différent de i: $((A_{ik}\ v'_i)\ \Pi_1)=((A_{ik}\ v_i)\ \Pi_1)$, $((A_{ik}\ v'_i)\ \Pi_2)=état(v)*i$, et $((A_{ik}\ v'_i)\ \Pi_3)=((A_{ik}\ v_i)\ \Pi_3)$. Il est clair qu'alors v' vérifie la propriété (P).

Supposons maintenant $i\neq lettre(v)$. Définissons dans ce cas v'_i par $\forall\ k\neq i$, $\forall\ k\neq j$, $((A_{ik}\ v'_i)\ \Pi_1)=i-lettre(v)+((A_{ik}\ v_i)\ \Pi_1)$(modulo m), $((A_{ij}\ v'_i)\ \Pi_1)=i-((A_{ij}\ v_i)\ \Pi_3)+((A_{ij}\ v_i)\ \Pi_1)$ (modulo m), $\forall\ k\neq i$, $((A_{ik}\ v'_i)\ \Pi_2)=état(v)*i$, et $\forall\ k\neq i$, $((A_{ik}\ v'_i)\ \Pi_3)=v'\,s_i$ (modulo m).

Montrons maintenant que $v'=v_0 \oplus u$ vérifie (P). D'après la construction de v', il suffit de prouver qu'il n'existe pas $k \neq i$ tel que pour tout l différent de k, $((A_{kl}\ v')\ \Pi_3)=v'\ S_k$ (modulo m). Alors, pour $l=i$ (différent de k), on obtient $((A_{ik}\ v')\ \Pi_3)=v'\ S_k$ (modulo m). D'autre part, par construction de v', $((A_{ik}\ v')\ \Pi_3)=v'\ S_i$ (modulo m) et $i=v'\ S_k$ (modulo m). On déduit alors $v'\ S_i=v'\ S_k=i$ (modulo m). Nous allons montrer maintenant que $((A_{ij}\ v_i)\ \Pi_3)=i$. Distinguons deux cas:

Si $k \neq j$, $((A_{ij}\ v_i)\ \Pi_3)=((A_{kj}\ v)\ \Pi_3)$ car v vérifie (P) et $j=lettre(v)$. Or $((A_{kj}\ v)\ \Pi_3)=((A_{kj}\ v')\ \Pi_3)$ car k et j sont différents de i. Par hypothèse, $((A_{kj}\ v')\ \Pi_3)=v'\ S_k$ (modulo m), donc $((A_{ij}\ v_i)\ \Pi_3)=v'\ S_{k=i}$ (modulo m).

Si $k=j$, comme A contient au moins trois lettres, il existe l dans A avec $l \neq i$ et $l \neq j$. Comme v vérifie (P) avec $j=lettre(v)$, $((A_{ij}\ v)\ \Pi_3)=((A_{lj}\ v)\ \Pi_3)$ et comme $i \neq j$ et $i \neq l$, $A_{lj}\ v$ ne peut être modifié par la lettre i, donc $((A_{lj}\ v)\ \Pi_3)=((A_{lj}\ v')\ \Pi_3)$ qui est égal à $v'\ S_j$ modulo n soit i.

Nous avons donc établi dans les deux cas que $((A_{ij}\ v_i)\ \Pi_3)=i$. D'autre part, par construction de v', pour tout l différent de i et différent de j, $((A_{il}\ v_i)\ \Pi_1)=j-i+((A_{il}\ v_i')\ \Pi_1)$ (modulo n) et $((A_{ij}\ v_i)\ \Pi_1)=((A_{ij}\ v_i)\ \Pi_3)-i+((A_{ij}\ v_i')\ \Pi_1)$ (modulo m)$=((A_{ij}\ v_i')\ \Pi_1)$ (modulo m). Comme $((A_{ij}\ v_i)\ \Pi_1) \in M$ et $((A_{ij}\ v_i')\ \Pi_1) \in M$, on déduit $((A_{ij}\ v_i)\ \Pi_1)=((A_{ij}\ v_i')\ \Pi_1)$. Considérons maintenant $v\ S_i$. Comme v vérifie (P), $v\ S_i=j$ (modulo m). D'autre part, $v\ S_i=\sum_{l \neq i}((A_{il}\ v_i)\ \Pi_1)=\sum_{l \neq j}((A_{il}\ v_i)\ \Pi_1)+((A_{ij}\ v)\ \Pi_1)$.

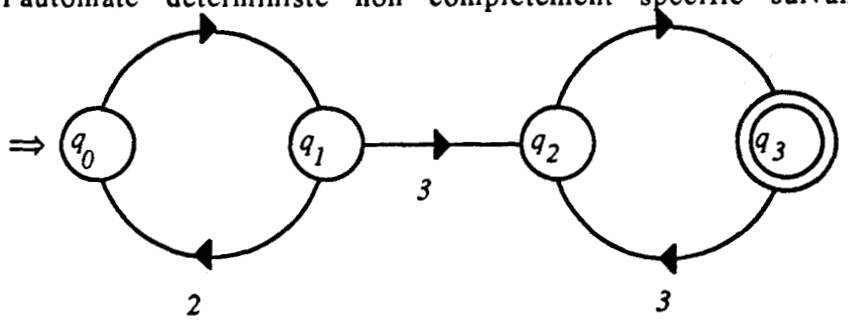
D'après ce qui précède, on déduit:

$$v\ S_i=\sum_{l \neq j}[(j-i+((A_{il}\ v_i')\ \Pi_1))]+((A_{ij}\ v_i')\ \Pi_1) \text{ (modulo } m).$$

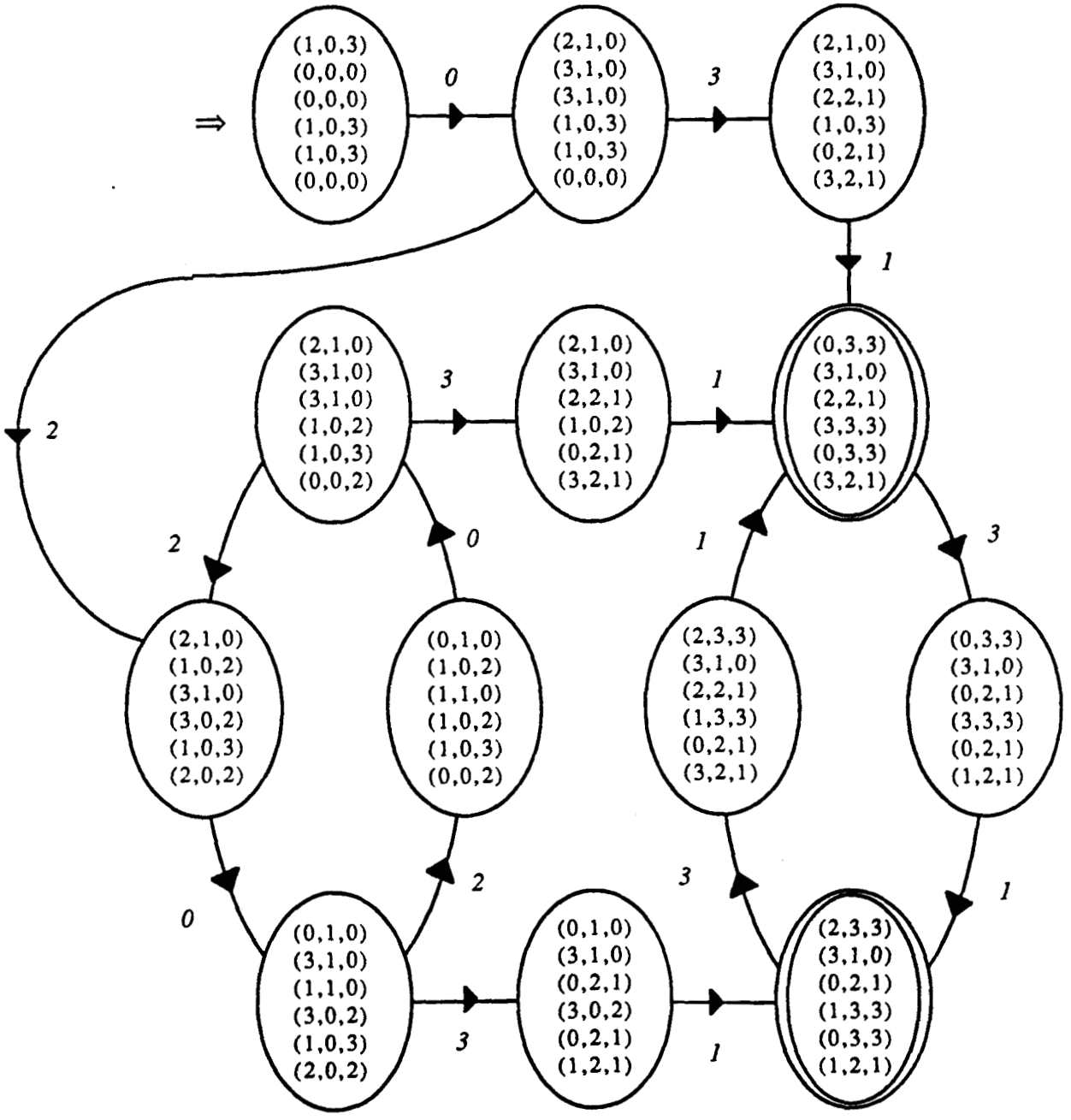
Donc, $j=(n-2)(j-1)+\sum_{l \neq i}((A_{il}\ v_i')\ \Pi_1)$ (modulo m) et $(n-3)(j-1)=0$ (modulo m), car $\sum_{l \neq i}((A_{il}\ v_i')\ \Pi_1)=v'\ S_i=i$ (modulo m). On déduit alors $(m-n+3)(j-i)=0$ (modulo m). Or m est tel que $m-n+3$ ne divise pas m (on remarquera qu'il suffit de prendre $m=n$ dans le cas où n n'est pas divisible par 3). Donc $j=i$, ce qui contredit notre hypothèse $i \neq lettre(v)$. Cette contradiction nous permet d'affirmer que v' vérifie (P). En posant $T=\{v_0 \oplus u /\ u \in A^*, état(v) \in S\}$, on déduit facilement du fait que la propriété (P) est vérifiée par tout $v=v_0 \oplus u$, légalité entre le langage reconnu par M et le langage reconnu par M'.

□

Exemple 3.5: Soit l'alphabet $A=\{0,1,2,3\}$ et $R=0(20)^*(31)^+$ le langage reconnu par l'automate déterministe non complètement spécifié suivant:



En appliquant la construction précédente, nous obtenons l'automate 2-asynchrone sur la famille $\{\{0,1\},\{0,2\},\{0,3\},\{1,2\},\{1,3\},\{2,3\}\}$, non complètement spécifié, reconnaissant R, donné ci-dessous. Chaque état v de cet automate, est représenté par $(A_{01}\ v)(A_{02}\ v)(A_{03}\ v)\ (A_{12}\ v)(A_{13}\ v)(A_{23}\ v)$.



Bibliographie

[1] I.J. Aalbersberg et G. Rozenberg, *"Theory of traces"* , Technical Report n°86-16, Department of Computer Science, University of Leiden, 1986.

[2] J. Beauquier, M. Blattner et M. Latteux, *"On commutative context-free languages"*, Journal of Computer and System Science, n°35, 1987.

[3] C. Berge, *"Graphes et hypergraphes"*, Dunod, 1970.

[4] J. Berstel, *"Transductions and Context-free languages"*, Teubner Verlag, 1978.

[5] A. Bertoni, G. Mauri et N. Sabadini, *"Unambiguous regular trace languages"*, Algebra, Combinatorics and Logic in Computer Science, Colloquia Math. Soc. J. Bolyai, n°42, pp 113-123, North Holland, Amsterdam, 1985.

[6] D. Bruschi, G. Pighizzini et N. Sabadini, *"On the existence of the minimum asynchronous automaton and on decision problems for unambiguous regular trace languages"*, pp 334-345.

[7] P. Cartier et D. Foata, *"Problèmes combinatoires de commutation et réarrangements"*, Lecture Notes in Mathematics, n°85, 1969.

[8] M. Clerbout, *"Commutations partielles et familles de langages"*, Thèse, Université de Lille I, 1984.

[9] M. Clerbout, *"Compositions de fonctions de commutation partielle"*, à paraître dans RAIRO Informatique Théorique, 1986.

[10] M. Clerbout et M. Latteux, *"Partial commutations and faithful rational transductions"*, Theoretical Computer Science, n°34, pp 241-254, 1985.

[11] M. Clerbout et M. Latteux, *"On a generalization of partial commutations"*, 4th Hungarian Computer Science Conference, pp 15-24, 1985.

[12] M. Clerbout et M. Latteux, *"Semi-commutations"*, Information and Computation, n°73, pp 59-74, 1987.

[13] M. Clerbout et Y. Roos, *"Semi-commutations algébrico-rationnelles"*, Publication Interne n°IT 126-88, LIFL, Université de Lille Flandres-Artois, 1988.

[14] M. Clerbout et Y. Roos, *"Semi-commutations et langages algébriques"*, Publication Interne n°IT 129-88, LIFL, Université de Lille Flandres-Artois, 1988.

[15] R. Cori, *"Partially abelian monoïds"*, Invited lecture, STACS, Orsay, 1986.

[16] R. Cori, M. Latteux, Y. Roos et E. Sopena, *"2-asynchronous automata"*, à paraître dans Theoretical Computer Science, 1987.

[17] R. Cori et Y. Métivier, *"Recognizable subsets of partially abelian monoïds"*, Theoretical Computer Science, n°38(, pp 179-189, 1985.

[18] R. Cori et D. Perrin, *"Automates et commutations partielles"*, RAIRO Informatique Théorique, n°19, pp 21-32, 1985.

[19] R. De Simone, *"Langages infinitaires et produit de mixage"*, Theoretical Computer Science, n°31, pp 83-100, 1984.

[20] C. Duboc, *"Commutations dans les monoïdes libres: un cadre théorique pour l'étude du parallélisme"*, Thèse, Université de Rouen, 1986.

[21] M.P. Flé et G. Roucairol, *"Maximal serializability or iterated transactions"*, Theoretical Computer Science, n°38, pp 1-16, 1985.

[22] M. Latteux, *"Cônes rationnels commutatifs"*, Journal of Computer and System Science, n°18, pp 307-333, 1979.

[23] M. Latteux et G. Rozenberg, *"Commutative one-counter languages are regular"*, Journal of Computer Science, n°29, pp 54-57, 1984.

[24] A. Mazurkiewicz, *"Concurrent program schemes and their interpretations"*, DAIMI PB 78, University of Aarhus, 1977.

[25] A. Mazurkiewicz, *"Traces, histories and graphs: instances of process monoïds"*, Lecture Notes in Computer Science, n°176, pp 115-133, 1984.

[26] Y. Métivier, *"Semi-commutations dans le monoïde libre"*, Publication interne n°I-8606, Université de Bordeaux, 1986.

[27] Y. Métivier, *" Contribution à l'étude des monoïdes de commutation"*, Thèse d'état, Université de Bordeaux, 1987.

[28] Y. Métivier, *"On recognizable subsets of free partially commutative monoïds"*, Lecture Notes in Computer Science, n° 226, pp 254-264, 1986.

[29] E. Ochmanski, *"Regular behaviour of concurrent systems"* , Bulletin of EATCS, n°27, pp 56-67, 1985.

[30] D. Perrin, *"Words over a partially commutative alphabet"*, NATO ASI Series F12, Springer, pp 329-340, 1985.

[31] Y. Roos, *"Virtually asynchronous automata"*, 2nd Conference on Automata, Languages and Programming Systems, Salgótarján, 1988.

[32] B. Rozoy, *"Un modèle de parallélisme: le monoïde distribué"*, Thèse d'état, Université de Caen, 1987.

[33] J. Sakarovitch, *"On regular trace languages"*, à paraître dans RAIRO Informatique Théorique, 1986.

[34] M. Szijártó, *"The closure of languages on a binary relation"*, IMYCS Conference, Smolenice, 1982.

[35] M. Szijártó et D. Van Hung, *"Synchronized parallel composition of languages"*, 1st Conference on Automata, Languages and Programming Systems, Salgótarján, 1986.

[36] W. Zielonka, *"Notes on asynchronous automata"*, RAIRO Informatique Théorique, n°21, pp 99-135, 1987.

Résumé:

Ce travail poursuit l'étude des fonctions de commutation dans les mots. De telles fonctions permettent de modéliser le comportement de processus concurrents: les fonctions de commutation partielle représentent le fait que certaines actions peuvent s'exécuter simultanément et les fonctions de semi-commutation formalisent des transactions du type "producteur-consommateur".

Dans le chapitre consacré aux compositions de fonctions de commutation partitionnée, nous répondons négativement à une question formulée par M. Clerbout concernant l'existence d'une forme normale pour ces compositions de fonctions. Nous y introduisons également la notion de fonction de rangement et nous montrons qu'il est décidable de déterminer si une composition de fonctions de commutation partitionnée est une fonction de rangement (universelle).

Les fonctions de semi-commutation algébrico-rationnelles sont des fonctions de semi-commutation qui transforment tout langage rationnel en langage algébrique. Elles sont introduites dans le chapitre traitant de la fermeture de langages par fonctions de semi-commutation. Nous donnons une caractérisation simple des graphes de commutation associés à ces fonctions. Nous donnons également une condition nécessaire et suffisante pour que l'image par une fonction de semi-commutation de l'étoile d'un mot soit algébrique.

Le dernier chapitre est consacré aux automates asynchrones introduits par W. Zielonka. Nous y montrons en particulier que tout rationnel peut être reconnu par un automate 2-asynchrone. Nous donnons également une condition nécessaire et suffisante, décidable, pour qu'un automate soit virtuellement asynchrone sur une famille d'alphabets, c'est à dire qu'il peut être rendu asynchrone par un simple renommage de ses états.

Mots clés:
Commutation partielle,
semi-commutation,
fonction algébrico-rationnelle,
automate asynchrone.