N° d'ordre : 1151

UNIVERSITE DES SCIENCES ET TECHNIQUES DE LILLE

THESE

présentée à

l'UNIVERSITE DES SCIENCES ET TECHNIQUES DE LILLE

pour obtenir

LE TITRE DE DOCTEUR DE 3ème CYCLE

Mireille **CLERBOUT**

COMMUTATIONS PARTIELLES ET FAMILLES DE LANGAGES

Thèse soutenue le 22 mars 1984 , devant la Commission d'Examen

MEMBRES DU JURY	G. JACOB	Président
	M. LATTEUX	Rapporteur
	J.M. AUTEBERT	Examinateur
	L. BOASSON	"
	M. DAUCHET	"

P R O F E S S E U R S CLASSE EXCEPTIONNELLE

M.	CONSTANT Eugène	I.E.E.A.
M.	FOURET René	Physique
M.	GABILLARD Robert	I.E.E.A.
M.	MONTREUIL Jean	Biologie
M.	PARREAU Michel	Mathématiques
M.	TRIDOT Gabriel	Chimie
M.	VIVIER Emile	Biologie
M.	WERTHEIMER Raymond	Physique

P R O F E S S E U R S 1ère classe

M.	BACCHUS Pierre	Mathématiques
M.	BEAUFILS Jean-Pierre (dét.)	Chimie
M.	BIAYS Pierre	G.A.S.
M.	BILLARD Jean (dét.)	Physique
M.	BOILLY Bénoni	Biologie
M.	BOIS Pierre	Mathématiques
M.	BONNELLE Jean-Pierre	Chimie
M.	BOUGHON Pierre	Mathématiques
M.	BOURIQUET Robert	Biologie
M.	BREZINSKI Claude	I.E.E.A.
M.	CELET Paul	Sciences de la Terre
M.	CHAMLEY Hervé	Biologie
M.	COEURE Gérard	Mathématiques
M.	CORDONNIER Vincent	I.E.E.A.
M.	DEBOURSE Jean-Pierre	S.E.S.
M.	DYMENT Arthur	Mathématiques

PROFESSEURS 1ère classe (suite)

M. ESCAIG Bertrand	Physique
M. FAURE Robert	Mathématiques
M. FOCT Jacques	Chimie
M. GRANELLE Jean-Jacques	S.E.S.
M. GRUSON Laurent	Mathématiques
M. GUILLAUME Jean	Biologie
M. HECTOR Joseph	Mathématiques
M. LABLACHE COMBIER Alain	Chimie
M. LACOSTE Louis	Biologie
M. LAVEINE Jean Pierre	Sciences de la Terre
M. LEHMANN Daniel	Mathématiques
Mme LENOBLE Jacqueline	Physique
M. LHOMME Jean	Chimie
M. LOMBARD Jacques	S.E.S.
M. LOUCHEUX Claude	Chimie
M. LUCQUIN Michel	Chimie
M. MIGEON Michel Recteur à Grenoble	E.U.D.I.L.
M. MIGNOT Fulbert (dét.)	Mathématiques
M. PAQUET Jacques	Sciences de la Terre
M. PROUVOST Jean	Sciences de la Terre
M. ROUSSEAU Jean-Paul	Biologie
M. SALMER Georges	I.E.E.A.
M. SEGUIER Guy	I.E.E.A.
M. SIMON Michel	S.E.S.
M. STANKIEWICZ François	S.E.S.
M. TILLIEU Jacques	Physique
M. VIDAL Pierre	I.E.E.A.
M. ZEYTOUNIAN Radyadour	Mathématiques

P R O F E S S E U R S 2ème classe

M. ANTOINE Philippe	Mathématiques (Calais)
M. BART André	Biologie
Mme BATTIAU Yvonne	Géographie
M. BEGUIN Paul	Mathématiques
M. BELLET Jean	Physique
M. BERZIN Robert	Mathématiques
M. BKOUCHE Rudolphe	Mathématiques
M. BODARD Marcel	Biologie
M. BOSQ Denis	Mathématiques
M. BRASSELET Jean-Paul	Mathématiques
M. BRUYELLE Pierre	Géographie
M. CAPURON Alfred	Biologie
M. CARREZ Christian	I.E.E.A.
M. CAYATTE Jean-Louis	S.E.S.
M. CHAPOTON Alain	C.U.E.E.P.
M. COQUERY Jean-Marie	Biologie
Mme CORSIN Paule	Sciences de la Terre
M. CORTOIS Jean	Physique
M. COUTURIER Daniel	Chimie
M. CROSNIER Yves	I.E.E.A.
M. CURGY Jean-Jacques	Biologie
Mle DACHARRY Monique	Géographie
M. DAUCHET Max	I.E.E.A.
M. DEBRABANT Pierre	E.U.D.I.L.
M. DEGAUQUE Pierre	I.E.E.A.
M. DELORME Pierre	Biologie
M. DELORME Robert	S.E.S.
M. DE MASSON D'AUTUME Antoine	S.E.S.
M. DEMUNTER Paul	C.U.E.E.P.

PROFESSEURS 2ème classe (Suite 1)

M.	DENEL Jacques	I.E.E.A.
M.	DE PARIS Jean-Claude	Mathématiques (Calais)
Mle	DESSAUX Odile	Chimie
M.	DEVRAINNE Pierre	Chimie
M.	DHAINAUT André	Biologie
Mme	DHAINAUT Nicole	Biologie
M.	DORMARD Serge	S.E.S.
M.	DOUKHAN Jean-Claude	E.U.D.I.L.
M.	DUBOIS Henri	Physique
M.	DUBRULLE Alain	Physique (Calais)
M.	DUBUS Jean-Paul	I.E.E.A.
M.	FAKIR Sabah	Mathématiques
M.	FONTAINE Hubert	Physique
M.	FOUQUART Yves	Physique
M.	FRONTIER Serge	Biologie
M.	GAMBLIN André	G.A.S.
M.	GLORIEUX Pierre	Physique
M.	GOBLOT Rémi	Mathématiques
M.	GOSSELIN Gabriel (dét.)	S.E.S.
M.	GOUDMAND Pierre	Chimie
M.	GREGORY Pierre	I.P.A.
M.	GREMY Jean-Paul	S.E.S.
M.	GREVET Patrice	S.E.S.
M.	GUILBAULT Pierre	Biologie
M.	HENRY Jean-Pierre	E.U.D.I.L.
M.	HERMAN Maurice	Physique
M.	JACOB Gérard	I.E.E.A.
M.	JACOB Pierre	Mathématiques
M.	JEAN Raymond	Biologie
M.	JOFFRE Patrick	I.P.A.

PROFESSEURS 2ème classe (suite 2)

M. JOURNEL Gérard	E.U.D.I.L.
M. KREMBEL Jean	Biologie
M. LANGRAND Claude	Mathématiques
M. LATTEUX Michel	I.E.E.A.
Mme LECLERCQ Ginette	Chimie
M. LEFEVRE Christian	Sciences de la Terre
Mle LEGRAND Denise	Mathématiques
Mle LEGRAND Solange	Mathématiques (Calais)
Mme LEHMANN Josiane	Mathématiques
M. LEMAIRE Jean	Physique
M. LHENAFF René	Géographie
M. LOCQUENEUX Robert	Physique
M. LOSFELD Joseph	C.U.E.E.P.
M. LOUAGE Francis (dét.)	E.U.D.I.L.
M. MACKE Bruno	Physique
M. MAIZIERES Christian	I.E.E.A.
M. MESSELYN Jean	Physique
M. MESSERLIN Patrick	S.E.S.
M. MONTEL Marc	Physique
Mme MOUNIER Yvonne	Biologie
M. PARSY Fernand	Mathématiques
Mle PAUPARDIN Colette	Biologie
M. PERROT Pierre	Chimie
M. PERTUZON Emile	Biologie
M. PONSOLLE Louis	Chimie
M. PORCHET Maurice	Biologie
M. POVY Lucien	E.U.D.I.L.
M. RACZY Ladislas	I.E.E.A.
M. RAOULT Jean François	Sciences de la Terre
M. RICHARD Alain	Biologie

PROFESSEURS 2ème Classe (suite 3)

M.	RIETSCH François	E.U.D.I.L.
M.	ROBINET Jean-Claude	E.U.D.I.L.
M.	ROGALSKI Marc	Mathématiques
M.	ROY Jean-Claude	Biologie
M.	SCHAMPS Joël	Physique
Mme	SCHWARZBACH Yvette	Mathématiques
M.	SLIWA Henri	Chimie
M.	SOMME Jean	G.A.S.
Mle	SPIK Geneviève	Biologie
M.	STAROSWIECKI Marcel	E.U.D.I.L.
M.	STERBOUL François	E.U.D.I.L.
M.	TAILLIEZ Roger	Institut Agricole
Mme	TJOTTA Jacqueline (dét.)	Mathématiques
M.	TOULOTTE Jean-Marc	I.E.E.A.
M.	TURRELL Georges	Chimie
M.	VANDORPE Bernard	E.U.D.I.L.
M.	VAST Pierre	Chimie
M.	VERBERT André	Biologie
M.	VERNET Philippe	Biologie
M.	WALLART Francis	Chimie
M.	WARTEL Michel	Chimie
M.	WATERLOT Michel	Sciences de la Terre
Mme	ZINN JUSTIN Nicole	Mathématiques

CHARGES DE COURS

M. ADAM Michel	S.E.S.

CHARGES DE CONFERENCES

M. BAFCOP Joël	I.P.A.
M. DUVEAU Jacques	S.E.S.
M. HOFLACK Jean	I.P.A.
M. LATOUCHE Serge	S.E.S.
M. MALAUSSENA DE PERNO Jean-Louis	S.E.S.
M. NAVARRE Christian	I.P.A.
M. OPIGEZ Philippe	S.E.S.

Il est important pour moi de témoigner ici de ma reconnaissance à ceux et à celles qui, de près ou de loin, ont fait en sorte que ce travail existe.

Tout particulièrement, je remercie de tout coeur Michel Latteux ; par sa passion pour la recherche et son sens des relations, il a été sans doute beaucoup plus pour moi qu'un directeur de thèse. Il m'a sans cesse aidée, conseillée mais aussi rassurée parfois...

Je remercie également Gérard Jacob qui a bien voulu être le président du jury, Luc Boasson et Jean-Michel Autebert qui ont toujours suivi de près ce travail, et Max Dauchet qui m'a fait le plaisir de participer au jury.

La réalisation pratique de cette thèse est due à Françoise Tailly pour la frappe, et je la remercie pour sa patience, et à Henri Glanc qui en a fait le tirage avec sa gentillesse habituelle.

Et puis, il y a le "labo", et son ambiance : plus que des relations de travail, j'y retrouve des ami(e)s, à tous les "niveaux hiérarchiques", et c'est une chance extraordinaire ...

COMMUTATIONS PARTIELLES

ET

FAMILLES DE LANGAGES

INTRODUCTION

L'intérêt de plus en plus vif porté à la mise en oeuvre du parallélisme dans les calculs et les programmes a fait naître le besoin d'avoir des modèles formels mieux adaptés pour décrire et étudier les systèmes et les processus parallèles. Le modèle mathématique qui décrit le mieux le parallélisme dans un processus est celui des Réseaux de Pétri, développés dans les années 60-62 par C.A. Pétri, pour modéliser les concepts d'actions asynchrones et concurrentes.

En "déployant" un réseau de Pétri représentant une tâche ou un programme, on obtient un ensemble de processus associés aux différentes exécutions possibles : son comportement. Cette situation est analogue à celle plus classique de la théorie des automates : un automate peut représenter un programme séquentiel et le langage associé décrit son comportement : un ensemble de processus séquentiels. On peut alors se poser les problèmes habituels d'analyse (à partir d'une "machine", trouver son comportement) et de synthèse (trouver un système vérifiant un comportement donné).

Dans le but de déterminer plus facilement le comportement d'un réseau de Pétri, Mazurkiewicz a utilisé un outil très simple : les "traces" et les langages traces [16]. Cette notion fut introduite pour la première fois par Cartier et Foata en 1969 [7] pour l'étude des problèmes de réarrangements dans les mots. Tandis qu'un mot, au sens usuel, est une suite de noms d'actions, une trace est définie comme un ensemble partiellement ordonné de noms d'actions.

Un premier pas pour associer un modèle formel à un système parallèle consiste en effet à associer un nom à chaque action élémentaire et à décrire par une relation binaire sur l'ensemble des noms d'actions le fait que deux actions (ou plus) peuvent être exécutées simultanément.

Si X désigne alors l'ensemble des noms (l'alphabet) et C la relation définie sur X, on associe à (X,C) une relation d'équivalence notée $\sim$ sur X^*

qui sera la plus petite relation vérifiant la propriété suivante : si $w = uxyv$, $w' = uyxv$ et $(x,y) \in C$, alors w et w' sont équivalents ($w \sim w'$). Une trace se définit alors comme une classe d'équivalence pour cette relation et un langage trace est un ensemble de traces, autrement dit, un sous ensemble du monoïde libre partiellement commutatif correspondant à la structure quotient $X^*/\sim$.

A partir de ces définitions, plusieurs approches correspondant à des concepts différents sont alors envisagées : elles relèvent des problèmes de combinatoire [12], de l'étude de certaines catégories de Réseaux de Pétri [11], mais surtout de la théorie des langages formels (voir l'analogie avec la définition d'un langage comme un sous-ensemble d'un monoïde libre). Le développement de cet aspect est à l'origine de nombreuses recherches et publications où sont étudiées entre autre les propriétés des langages traces réguliers et algébriques ([2], [3]), une classification des familles de langages traces, semblable à la classification fondamentale de Chomsky et leurs propriétés de clôture pour les opérations usuelles ([18],[19]).

Un autre point de vue, que l'on trouve par exemple dans [8] et que nous adopterons ici est l'étude des langages définis non plus comme un ensemble de traces, mais comme l'union de traces. L'objet de ce travail consiste plus particulièrement en l'étude des langages qui se définissent comme la clôture par une relation de commutation (au sens défini plus haut) d'un langage.

On pourra alors remarquer que ces langages se décrivent alors assez simplement par la donnée d'un langage plus simple et de la relation qu'on lui applique, alors que sans ces outils, il paraît plus difficile de les expliciter.

Nous allons définir trois catégories de relations de commutation sur un alphabet :

1) Les relations de semi-commutation qui ont l'unique propriété d'être irréflexives ($\forall\, x \in X$, x n'est pas en relation avec x). Par exemple, soient les deux actions "produire", représentée par la lettre P, et "consommer", représentée par la lettre C ; les enchaînements possibles de ces deux actions pourront être identifiés aux mots du langage défini comme la fermeture du langage rationnel $(PC)^*$ par la relation $\{(C,P)\}$: on peut remplacer tout

occurrence du facteur CP par le facteur PC, c'est-à-dire, on peut produire plus vite qu'on ne consomme. L'ensemble de ces relations sera noté SC.

2) La deuxième famille est constituée des relations dites de commutation partielle (notée CP) qui sont les relations de semi-commutation symétriques. Par exemple, deux instructions (ou deux actions) du type $A \leftarrow 1$, $B \leftarrow 1$ pourront être exécutées dans un ordre quelconque. Ces relations font l'objet des études précédemment citées.

3) Les relations associées à une partition de l'alphabet sur lequel elles sont définies en sont un cas particulier. Si π est une partition de l'alphabet X, la relation $C\pi$ associée à π est définie de la façon suivante : (a,b) appartient à $C\pi$ si et seulement si a et b ne sont pas dans la même classe pour π. On peut aussi définir une telle relation comme le complémentaire d'une relation d'équivalence. L'ensemble de ces relations sera noté P.

Soient X un alphabet, C une relation de semi-commutation, et w, w', deux mots de X^*. Nous dirons que w se dérive directement en w' par C et nous noterons $w \underset{C}{\rightarrow} w'$ si $w = uxyv$, $w' = uyxv$ et (x,y) appartient à C. $\overset{*}{\underset{C}{\rightarrow}}$ désigne la clôture réflexive et transitive de $\underset{C}{\rightarrow}$. A une relation C, nous pourrons alors associer une fonction de commutation f, définie sur X^* par $f(w) = \{w' \mid w \overset{*}{\underset{C}{\rightarrow}} w'\}$. Cette définition s'étend facilement aux langages en posant, si L est un langage défini sur X, $f(L) = \bigcup_{w \in L} f(w)$, et aux familles de langages : $f(\mathcal{L}) = \{f(L) \mid L \in \mathcal{L}\}$.

Dans le cas des relations de commutation partitionnée, l'image d'un mot w par une fonction f associée à une partition $\pi = \{X_1,\dots,X_k\}$ de l'alphabet X s'exprime très simplement à l'aide d'opérations mieux connues : en effet, f(w) est l'ensemble des mots obtenus en faisant le shuffle des k projections de w sur chaque X_i. D'où l'intérêt tout particulier qu'on portera à ces fonctions de commutation...

Ainsi, après avoir précisé les notations et les définitions dans le premier chapitre, nous allons étudier les propriétés des fonctions des différents types dans le deuxième chapitre : nous commencerons par donner quelques propriétés sur les dérivations associées à une relation de semi-commutation et notamment nous établirons que pour toute relation de semi-commutation C, w se dérive en w' par C si et seulement si, pour tout couple de lettres (x,y) de $X \times X$, la projection de w sur $\{x,y\}$ se dérive par C en la projection de w' sur $\{x,y\}$. Cette propriété est démontrée dans

[8], dans le cas des relations de commutation partielle. Ensuite, nous montrerons qu'une fonction de commutation partitionnée se factorise très simplement en un homomorphisme linéairement effaçant suivi de l'opération Twin Shuffle, notée TS, qui à un mot w fait correspondre le shuffle de w et de sa recopie sur un alphabet disjoint. Nous parviendrons alors à exprimer une fonction de commutation partielle à l'aide d'une fonction de commutation partitionnée, d'un homomorphisme et de son inverse. La factorisation d'une fonction de semi-commutation sera un peu plus longue et délicate, mais elle n'utilisera que des fonctions de commutation partielle, des homomorphismes non effaçants et des homomorphismes inverses.

Puisque toute fonction de semi-commutation se simule à l'aide d'opérations plus simples et de transductions rationnelles, il semble alors naturel d'étudie les propriétés des cônes rationnels (familles de langages fermées par transductions rationnelles) fermés par les fonctions de semi-commutation.

Ainsi, nous débuterons le troisième chapitre en étudiant un exemple de cône rationnel fidèle clos pour l'opération Twin Shuffle, il s'agit de la famille des langages MULTIRESET, définie comme le plus petit cône rationnel fidèle fermé par intersection et contenant le langage COPY = $\{ww,\ w \in \{a,b\}^*\}$ (voir [5]). Nous établirons alors, pour un cône rationnel fidèle, $\mathcal{L}$, fermé par intersection, les équivalences des trois propriétés suivantes : $\mathcal{L}$ contient le langage COPY ; $\mathcal{L}$ est fermé pour l'opération Twin Shuffle ; $\mathcal{L}$ est fermé pour les opérations de commutation partitionnée (donc partielle et de semi-commutati Ces propriétés nous permettront par ailleurs de retrouver un résultat déjà étab dans [5] : un cône rationnel fidèle clos par intersection est fermé par homomorphisme linéairement effaçant si et seulement si il contient le langage COPY, mais la méthode sera ici beaucoup plus algébrique que dans [5] où la mise en oeuvre de machines complexes rend les démonstrations plus lourdes. Ces résultats sont également à rapprocher de ceux de [17] où les auteurs montrent que la famille des langages FIFO (reconnus en temps quasi réel par des automates à files) n'est autre que la fermeture par autoshuffle de la famille des langages rationnels et que le langage TS = $\{w \sqcup\!\sqcup \tilde{w},\ w \in \{a,b\}^*\}$. (le "Complete Twin Shuffle") est un générateur fidèle des langages FIFO. Ensuite, nous nous intéressons au plus petit cône rationnel fermé par commutation partitionnée et nous établissons que la famille des langages MULTIRESET est le plus petit cône rationnel fidèle possédant cette propriété, et que la famille des langages récursivement énumérables est le plus petit cône rationnel fermé par commutation partitionnée. Nous achevons cette partie en donnant une nouvelle caractérisation des familles de langages MULTIRESET, BNP ([6]) et Q([4]). Par exemple, MULTIRESET = $H^{S\alpha}(P(RAT) \wedge RAT)$, c'est-à-dire, tout langage L de la famille MULTIRESET peut se mettre sous la forme $L = h(f(R) \wedge K)$ où h est un homomorphisme strictement alphabétique, f une fonction de commutation partitionnée et R et K sont deux langages rationnels.

Le quatrième chapitre aborde les questions naturelles que soulève l'introduction de nouvelles familles de langages. Nous montrerons ainsi que les familles de langages SC^k (RAT) pour $k \geq 0$ (on applique successivement à un langage rationnel k fonctions de semi-commutation) constituent une hiérarchie infinie. La conclusion sera la même pour les familles CP^k (RAT) et P^k(RAT), et cette dernière propriété nous permettra d'établir plusieurs résultats sur la composition des fonctions de commutation partitionnée. Nous terminons par quelques problèmes de décidabilité, concernant pour la plupart les langages de P(RAT).

P L A N

CHAPITRE I - DEFINITIONS, NOTATIONS, RAPPELS

CHAPITRE II - FACTORISATION DES FONCTIONS DE COMMUTATION

A/ Quelques lemmes de dérivation

B/ Factorisation des fonctions de commutation partitionnée

C/ Factorisation des fonctions de commutation partielle

D/ Factorisation des fonctions de semi-commutation

CHAPITRE III - CÔNES RATIONNELS FIDÈLES ET COMMUTATION PARTIELLE

A/ MULTI-RESET est fermé pour l'opération TS

B/ Propriétés des familles de langages fermées par commutation partielle

C/ Le plus petit cône rationnel fidèle clos par commutation partielle

CHAPITRE IV - HIÉRARCHIES ET PROBLÈMES DE DÉCIDABILITÉ

A/ La hiérarchie des familles de langages $CP^k(RAT)$, $k \geq 0$

B/ La hiérarchie des familles de langages $SC^k(RAT)$, $k \geq 0$

C/ La hiérarchie des familles de langages $P^k(RAT)$, $k \geq 0$

D/ Comparaisons entre familles

E/ Problèmes de décidabilité

CHAPITRE I.

DÉFINITIONS, NOTATIONS, RAPPELS

Nous supposons connus les résultats classiques de la théorie des langages (voir [1], [9]).

Si X est un alphabet et w un mot de X^*, nous noterons

- ε le mot vide de X^*,
- $|w|$ la longueur de w
- $|w|_x$ le nombre d'occurrences de la lettre x de X dans w
- $\mathrm{Alph}(w)$ l'ensemble $\{x \in X,\ |w|_x \neq 0\}$.

Soit X' une partie de l'alphabet X. La <u>projection</u> de X sur X' est l'homomorphisme noté proj (X',X) défini sur X par proj $(X,X')(x) = x$ si $x \in X'$, proj $(X,X')(x) = \varepsilon$ sinon. Si il n'y a pas d'ambiguïté sur l'alphabet de départ X, nous écrirons souvent pour un mot w de X^*, proj (w,X') au lieu de proj $(X,X')(w)$.

Un morphisme h défini sur X^*, à valeur dans Y^* est :

- <u>alphabétique</u> si $h(X) \subseteq Y \cup \{\varepsilon\}$
- <u>strictement alphabétique</u> si $h(X) \subseteq Y$
- <u>non effaçant</u> si $h(X) \subseteq Y^+$
- <u>k-limité</u> sur un langage $L \subseteq X^*$ $(k \in \mathbb{N}^*)$ si $[uwv \in L$ et $h(w) = \varepsilon]$ impliquent $|w| \leq k$
- <u>k-effaçant</u> sur un langage $L \subseteq X^*$ $(k \in \mathbb{N}^*)$ si pour tout mot $w \in X^*$, $|w| < k(|h(w)| + 1)$
- <u>ε-limité</u> (resp. <u>linéairement effaçant</u>) sur un langage $L \subseteq X^*$ si il existe un entier $k > 0$ tel que h soit k-limité (resp. k-effaçant) sur L
- <u>marqué</u> si $X \subseteq Y$ et $\forall\, x \in X$, il existe $u \in (Y \setminus X)^*$ tel que tel que $h(x) = xu$.

Une famille de langages $\mathcal{L}$ est fermée par homomorphisme linéairement effaçant (resp. ε-limité, alphabétique), si pour tout langage L de $\mathcal{L}$, et pour tout homomorphisme h linéairement effaçant sur L (resp. k-limité sur L, alphabétique), h(L) est dans $\mathcal{L}$.

Une application $\tau : X^* \to 2^{Y^*}$ est une transduction rationnelle (fidèle) si il existe un alphabet Z, un langage rationnel $R \subseteq Z^*$ et deux homomorphismes alphabétiques $h : Z^* \to X^*$ et $g : Z^* \to Y^*$ (avec g ε-limité sur R) tels que pour tout mot w de X^*, $\tau(w) = g(h^{-1}(w) \cap R)$. Une famille de langages est un cône rationnel (fidèle) si elle est fermée par transduction rationnelle (fidèle).

Soient $\mathcal{L}$ et $\mathcal{L}'$ deux familles de langages. Nous noterons :

- $C(\mathcal{L})$ le cône rationnel engendré par $\mathcal{L}$
- $C^f(\mathcal{L})$ le cône rationnel fidèle engendré par $\mathcal{L}$
- $C_\cap(\mathcal{L})$ $(C^f_\cap(\mathcal{L}))$ le plus petit cône rationnel (fidèle) fermé par intersection et contenant $\mathcal{L}$
- $\mathcal{L} \wedge \mathcal{L}' = \{L \cap L',\ L \in \mathcal{L},\ L' \in \mathcal{L}'\}$
- $\mathcal{H}^{s\alpha}(\mathcal{L}) = \{h(L),\ L \in \mathcal{L},$ h est un morphisme strictement alphabétique$\}$
- $\mathcal{H}^{\varepsilon}(\mathcal{L}) = \{h(L),\ L \in \mathcal{L},$ h est ε-limité sur L$\}$
- $\mathcal{H}^{lin}(\mathcal{L}) = \{h(L),\ L \in \mathcal{L},$ h est linéairement effaçant sur L$\}$
- $\mathcal{H}^{-1}(\mathcal{L}) = \{h^{-1}(L),\ L \in \mathcal{L},$ h est un homomorphisme$\}$.

Nous noterons :

- RAT l'ensemble des langages rationnels
- Alg l'ensemble des langages algébriques
- Lin l'ensemble des langages linéaires
- $C_1 = \{a^n b^n,\ n \geq 0\}$
- D_1^* (resp. $D_1'^*$) le langage de Dyck (resp. de semi-Dyck) sur 2 lettres.
- $COPY = \{w\bar{w}\ /\ w \in \{a,b\}^*\}$ et $MULTI\text{-}RESET = C^f_\cap(COPY)$
- $Sym = \{w\bar{w}^R\ /\ w \in \{a,b\}^*\}$ où $\bar{w}^R$ est la recopie du miroir de w sur un alphabet disjoint
- $BNP = C^f_\cap(Lin)$
- $Q = C^f_\cap(Alg)$.

L'opération de Shuffle, notée $\sqcup\!\sqcup$, est définie sur X^* par :

$\forall$ u, v $\in X^*$, $u \sqcup\!\sqcup v = \{u_1 v_1 \dots u_n v_n \;/\; n \geq 1,\ u_i, v_i \in X^*,\ u_1 u_2 \dots u_n = u,\ v_1 v_2 \dots v_n = v\}$.

Pour L et $L' \subseteq X^*$, $L \sqcup\!\sqcup L' = \bigcup_{u \in L, u' \in L'} u \sqcup\!\sqcup u'$.

Le langage Twin Shuffle est défini par $TS = \{w \sqcup\!\sqcup \bar{w},\ w \in \{a,b\}^*\}$. Nous pouvons alors définir l'opérateur TS :

Si $w \in X^*$, $TS(w) = \{w \sqcup\!\sqcup \bar{w}\}$ et pour un langage $L \subseteq X^*$, $TS(L) = \bigcup_{w \in L} TS(w)$.

L'ensemble des facteurs gauches d'un mot w de X^* est défini par

$FG(w) = \{u \in X^* \;/\; w = uv\}$. Pour un langage $L \subseteq X^*$, $FG(L) = \bigcup_{w \in L} FG(w)$.

Soient X un alphabet et C un sous-ensemble de $X \times X$. C est une relation de semi-commutation si C est une relation irreflexive. L'ensemble des relations de semi-commutation sera noté SC, et un élément (x,y) d'une relation C de type SC sera noté $xy \to yx$.

Quand la relation C a la propriété d'être symétrique, nous dirons que C est une relation de commutation partielle. L'ensemble des relations de commutation partielle aura pour nom CP et un élément (x,y) d'une relation C de type CP sera plus souvent écrit sous la forme $xy \leftrightarrow yx$. Un cas particulier des relations de commutation partielle sont les relations de commutation partitionnée : elles ont la propriété suivante :

Soit $C \subseteq X \times X$ une relation de commutation partielle. C est une relation de commutation partitionnée si et seulement si $\forall\ xy \leftrightarrow yx \in C$, $\forall\ z \in X$, $xz \leftrightarrow zx \in C$ ou $yz \leftrightarrow zy \in C$. Alors, la relation R définie sur X par $xRy \iff xy \leftrightarrow yx \notin C$ est une relation d'équivalence et l'ensemble des classes d'équivalence de cette relation constitue une partition de X. Nous pouvons donc donner la définition équivalente suivante :
$C \subseteq X \times X$ est une relation de commutation partitionnée si il existe une partition $\pi = \{X_1, X_2, \dots, X_k\}$ $(k \geq 1)$ telle que
$C = \{(\alpha,\beta) \;/\; \exists\ i, j \in \{1, \dots, k\},\ i \neq j,\ \alpha \in X_i,\ \beta \in X_j\}$.
Nous parlerons alors d'une relation associée à une partition.

L'ensemble des relations de commutation partitionnée sera noté P.

Soient $C \subseteq X \times X$ une relation de semi-commutation, et w un mot de X^*. Si $w = uxyv$ et si $xy \to yx \in C$, on dira que w se dérive directement en $w' = uyxv$ et on notera $w \underset{C}{\to} w'$. $\overset{*}{\underset{C}{\to}}$ désigne la clôture réflexive et transitive de $\underset{C}{\to}$. Dans le cas des relations de commutation partielle ou partitionnée, nous écrirons plutôt $w \underset{C}{\leftrightarrow} w'$ et $w \overset{*}{\underset{C}{\leftrightarrow}} w'$ (puisque ces relations sont symétriques, $w \overset{*}{\underset{C}{\leftrightarrow}} w' \Longleftrightarrow w' \overset{*}{\underset{C}{\leftrightarrow}} w$). A toute relation de commutation $C \subseteq X \times X$, nous pouvons alors associer une fonction de commutation f définie par $\forall\, w \in X^*$, $f(w) = \{w',\ w \overset{*}{\underset{C}{\to}} w'\}$ ou $f(w) = \{w',\ w \overset{*}{\underset{C}{\leftrightarrow}} w'\}$ suivant le type de la relation C : SC, CP ou P. Nous pouvons alors définir l'image d'un langage par une fonction de commutation f :

Si $L \subseteq X^*$ est un langage, $f(L) = \underset{w \in L}{\cup} f(w)$. Dans le cas d'une fonction f associée à une relation de commutation partitionnée C, c'est-à-dire à une partition $\pi = \{X_1,\ldots,X_k\}$ de X, pour tout mot w de X^*, $f(w)$ peut se définir de la manière suivante :

$$f(w) = \mathrm{proj}\,(w,X_1) \sqcup\!\sqcup\ \mathrm{proj}\,(w,X_2) \sqcup\!\sqcup \ldots \sqcup\!\sqcup\, \mathrm{proj}(w,X_k) = \{w'/w \overset{*}{\underset{C}{\leftrightarrow}} w'\}.$$

Si k est le cardinal de la partition associée à la relation C (resp. à la fonction f), nous dirons que C (resp. f) est une relation (resp. fonction) de commutation k-partitionnée.

Enfin, si $\mathcal{L}$ est une famille de langages, nous noterons

$$SC(\mathcal{L}) = \{f(L),\ L \in \mathcal{L},\ f \text{ est une fonction de semi-commutation}\}$$
$$CP(\mathcal{L}) = \{f(L),\ L \in \mathcal{L},\ f \text{ est une fonction de commutation partielle}\}$$
$$P_k(\mathcal{L}) = \{f(L),\ L \in \mathcal{L},\ f \text{ est une fonction de commutation } k\text{-partitionnée}\}$$

et $$P(\mathcal{L}) = \underset{k \geq 1}{\cup} P_k(\mathcal{L})$$

Un cas particulier de fonction de commutation partitionnée est la fonction de commutation totale :

Si $X = \{x_1,x_2,\ldots,x_p\}$, $\pi = \{\{x_1\}, \{x_2\}, \ldots, \{x_p\}\}$ et C est la relation associée à π, il est clair que $\forall\, w \in X^*$, $\{w' \mid w \overset{*}{\underset{C}{\leftrightarrow}} w'\} = \{w' \mid \forall\, x \in X,\ |w'|_x = |w|_x\}$. Nous noterons Com (w) cet ensemble. Pour un langage $L \subseteq X^*$, Com $(L) = \underset{w \in L}{\cup}$ Com (w).

CHAPITRE II

FACTORISATION DES FONCTIONS DE COMMUTATION

A) Quelques lemmes de dérivation

B) Factorisation des fonctions de commutation partitionnée

C) Factorisation des fonctions de commutation partielle

D) Factorisation des fonctions de semi-commutation

Le but de ce chapitre est de faire le lien entre les fonctions qui viennent d'être définies dans les préliminaires et les opérations plus classiques de la théorie des langages.

Dans une première section, nous allons énoncer quelques résultats simples sur les dérivations d'un mot pour une relation de commutation donnée. Ces résultats nous permettront de prouver le dernier lemme qui nous donne une condition nécessaire et suffisante pour que, étant donnés une relation de commutation $C \subseteq X \times X$ et deux mots w, w' de X^*, w se dérive en w' par C : W se dérive en $\ddot{w}'$ par C si et seulement si pour tout couple de lettres (x, y) de $X \times X$, la projection de $\ddot{w}$ sur $\{x,y\}$ se dérive par C en la projection de w' sur $\{x,y\}$.

Dans les deuxième, troisième et quatrième sections, nous allons montrer comment simuler toute fonction de commutation (et quelque soit son type) par des opérations simples et plus usuelles : ces opérations seront les transductions rationnelles et l'opérateur TS (rappelons que $TS(w) = w \sqcup\!\sqcup \bar{w}$).

Les fonctions de commutation partitionnée seront représentées par la composition d'homomorphismes linéairement effaçant et de l'opération TS, et ceci assez simplement (section B). Nous montrerons ensuite qu'une fonction de commutation partielle se simule par la composition d'un homomorhisme marqué suivi d'une fonction de commutation partitionnée puis du même homomorphisme marqué, mais inverse (section C).

La simulation d'une fonction de semi-commutation sera un peu plus délicate et compliquée, mais on pourra remarquer qu'elle n'utilisera que des fonctions de commutation partielle ou partitionnée, des homomorphismes inverses et des homomorphismes non effaçants (section D).

II.A - Quelques lemmes de dérivation.

Dans ce paragraphe, X désignera l'alphabet sur lequel on a défini une relation de commutation C de type SC.

Les résultats énoncés seront donc à fortiori vrais pour des relations de commutation de type CP ou P.

<u>*Lemme 1.*</u>

Soient w *et* w' *deux mots de* X^* *et* X_1 *un sous-ensemble de* X. *Si* w *se dérive par* C *en* w', *alors la projection de* w *sur* X_1 *se dérive par* C *en la projection de* w' *sur* X_1 :

$$w \overset{*}{\underset{C}{\rightarrow}} w' \Rightarrow \text{proj}(w, X_1) \overset{*}{\underset{C}{\rightarrow}} \text{proj}(w', X_1)$$

<u>Preuve</u>.

Elle est basée sur une récurrence sur la longueur de la dérivation $w \overset{*}{\underset{C}{\rightarrow}} w'$. Notons ℓ cette longueur.

* Si $\ell = o$, alors $w = w'$ et par conséquent $\text{proj}(w, X_1) = \text{proj}(w', X_1)$ donc $\text{proj}(w, X_1) \overset{*}{\underset{C}{\rightarrow}} \text{proj}(w', X_1)$.

* Supposons que pour toute dérivation de longueur inférieure ou égale à ℓ, le résultat soit vrai, et considérons une dérivation de longueur $\ell+1$.

$$w \xrightarrow[C]{\ell+1} w' \Rightarrow \exists u \in X \;/\; w \overset{\ell}{\underset{C}{\rightarrow}} u \underset{C}{\rightarrow} w'$$

1) $u \underset{C}{\rightarrow} w' \Rightarrow \exists u_1, u_2 \in X^*$ et $x, y \in X$,
tels que $\begin{cases} u = u_1\, xy\, u_2 \\ w' = u_1\, yx\, u_2 \\ xy \rightarrow yx \in C \end{cases}$

2) $w \xrightarrow[c]{} u \Rightarrow$ (par hypothèse de récurrence) $proj(w,X) \overset{*}{\underset{c}{\rightarrow}} proj(u,X_1)$

Nous allons regarder différents cas, suivant que x et y appartiennent ou non à X_1.

<u>1er Cas.</u> $x \notin X_1$, $y \notin X_1$

Alors $proj(u,X_1) = proj(u_1,X_1)\ proj(u_2,X_1) = proj(w',X_1)$

Donc $proj(w,X_1) \overset{*}{\underset{c}{\rightarrow}} proj(u,X_1) = proj(w',X_1)$

<u>2ème Cas.</u> $x \notin X_1$, $y \in X_1$ (le cas symétrique sera traité de la même manière).

Alors $proj(u,X_1) = proj(u_1,X_1)\ y\ proj(u_2,X_1) = proj(w',X_1)$

Donc $proj(w,X_1) \overset{*}{\underset{c}{\rightarrow}} proj(u,X_1) = proj(w',X_1)$.

<u>3ème Cas.</u> $x \in X_1$ et $y \in X_1$

Alors $proj(w,X_1) = proj(u_1,X_1)\ xy\ proj(u_2,X_1) \xrightarrow[c]{}$
$proj(u_1,X_1)\ yx\ proj(u_2,X_1) = proj(w,X_1)$, puisque $xy \rightarrow yx \in C$

Donc $proj(w,X_1) \overset{*}{\underset{c}{\rightarrow}} proj(u,X_1) \xrightarrow[c]{} proj(w',X_1)$.

Le résultat énoncé est donc vrai pour toute dérivation.

□

<u>*Lemme 2.*</u>

Soient u, v, v_1, v_2 *des mots de* X^* *vérifiant :*

$$uv \overset{*}{\underset{c}{\rightarrow}} v_1\, u\, v_2 \text{ et } Alph(u) \cap Alph(v_1) = \emptyset$$

alors *i)* $Alph(u) \times Alph(v_1) \subseteq C$
$(\forall x \in Alph(u), \forall y \in Alph(v_1), xy \rightarrow yx \in C)$

ii) $v \overset{*}{\underset{c}{\rightarrow}} v_1 v_2$

<u>Preuve.</u>

En premier lieu, supposons que $u \in X$ et raisonnons par induction sur la longueur de la dérivation : ℓ.

* Si $\ell = 0$, alors $v_1 = \varepsilon$ et $v = v_2$. Le résultat est alors évident.

* Si $\ell > 0$, on peut alors écrire :

$$uv \overset{*}{\underset{c}{\rightarrow}} w_1\ xy\ w_2 \underset{c}{\rightarrow} w_1\ yx\ w_2 = v_1\ u\ v_2$$

avec $xy \rightarrow yx \in C$.

Plusieurs possibilités peuvent alors être envisagées.

1) $w_1 = v_1\ u\ w'_1$ et $v_2 = w'_1\ yx\ w_2$.

Alors par hypothèse de récurrence, nous avons $\{u\} \times Alph(v_1) \subseteq C$ et $v \overset{*}{\underset{c}{\rightarrow}} v_1\ w'_1\ xy\ w_2$.
Mais $xy \rightarrow yx \in C$, donc $v_1\ w'_1\ xy\ w_2 \underset{c}{\rightarrow} v_1\ w'_1\ xy\ w_2 = v_1v_2$

2) $w_1 = v_1$, $u = y$, $v_2 = xw_2$.

Par hypothèse de récurrence, $v \overset{*}{\underset{c}{\rightarrow}} v_1\ x\ w_2 = v_1v_2$ et $\{u\} \times Alph\ (v_1x) \subseteq C$, donc $\{u\} \times Alph\ (v_1) \subseteq C$.

3) $w_1y = v_1$, $u = x$, $v_2 = w_2$.
Alors $u \overset{*}{\underset{c}{\rightarrow}} w_1\ y\ w_2 = v_1v_2$ et $\{u\} \times Alph\ (w_1) \subseteq C$.
Mais comme $xy \rightarrow yx \in C \Rightarrow \{u\} \times Alph\ (w_1y) = \{u\} \times Alph\ (v_1) \subseteq C$.

4) $v_1 = w_1\ yx\ w'_2$ et $w_2 = w'_2\ u\ v_2$.

Alors $uv \overset{*}{\underset{c}{\rightarrow}} w_1\ xy\ w'_2\ u\ v_2 \Rightarrow v \overset{*}{\underset{c}{\rightarrow}} w_1\ xy\ w'_2\ v_2$ et
$\{u\} \times Alph\ (w_1\ xy\ w'_2) = \{u\} \times Alph\ (w_1\ yx\ w'_2) = \{u\} \times Alph\ (v_1) \subseteq C$.
Et comme $xy \rightarrow yx \in C$, nous avons $v \overset{*}{\underset{c}{\rightarrow}} w_1\ xy\ w'_2v_2 \underset{c}{\rightarrow} w_1\ yx\ w'_2\ v_2 = v_1v_2$.

Montrons maintenant le résultat pour un mot u de longueur quelconque. Supposons que $u = u'x$, avec $x \in X$, et raisonnons par induction sur $|u|$.

$$uv = u'(xv) \overset{*}{\underset{c}{\rightarrow}} v_1\ u'(xv_2) \Rightarrow \text{(hyp rec) } xv \overset{*}{\underset{c}{\rightarrow}} v_1\ x\ v_2$$

et $\quad Alph\ (u') \times Alph\ (v_1) \subseteq C$.

D'après la première partie de la démonstration, $xv \overset{*}{\underset{C}{\rightarrow}} v_1\ x\ v_2$ implique $v \overset{*}{\underset{C}{\rightarrow}} v_1 v_2$ et

$$\{x\} \times \mathrm{Alph}(v_1) \subseteq C.$$

En regroupant ces deux résultats, nous avons donc :

$$v \overset{*}{\underset{C}{\rightarrow}} v_1 v_2$$

et

$$\mathrm{Alph}\ (u'x) \times \mathrm{Alph}(v_1) \subseteq C$$

$\square$

Remarque.

Un cas particulier de ce lemme, quand $v_1 = \varepsilon$, sera souvent utilisé :

$$uv \overset{*}{\underset{C}{\rightarrow}} uv' \Rightarrow v \overset{*}{\underset{C}{\rightarrow}} v'.$$

On peut remarquer de manière assez évidente que la réciproque est vraie.

Le lemme suivant est un peu le dual du précédent et s'en déduit facilement :

Lemme 3.

Soient u_1, u_2, v, u', *des mots de* X^* *vérifiant* $\mathrm{Alph}(u_1) \cap \mathrm{Alph}(v) = \emptyset$ *et* $u_1\ u\ u_2 \overset{*}{\underset{C}{\rightarrow}} vu'$. *Alors*

i) $u_1 u_2 \overset{*}{\underset{C}{\rightarrow}} u'$

ii) $\forall x \in \mathrm{Alph}\ (u_1), \forall y \in \mathrm{Alph}\ (v), xy \rightarrow yx \in C$

Preuve.

Si $u_1\ v\ u_2 \overset{*}{\underset{C}{\rightarrow}} vu'$, on a alors $vu' \overset{*}{\underset{C^{-1}}{\rightarrow}} u_1\ v\ u_2$, où $C^{-1} = \{xy \rightarrow yx / yx \rightarrow xy \in C\}$
Comme $\mathrm{Alph}\ (u_1) \cap \mathrm{Alph}(v) = \emptyset$, on déduit du lemme 2 que :

i) $u' \xrightarrow[C^{-1}]{*} u_1u_2$, c'est à dire $u_1u_2 \overset{*}{\underset{C}{\rightarrow}} u'$

ii) $\forall x \in \mathrm{Alph}(v)$, $\forall y \in \mathrm{Alph}(u_1)$, $xy \rightarrow yx \in C^{-1}$ c'est à dire, $yx \rightarrow xy \in C$.

□

Le quatrième lemme n'est qu'un corollaire des deux précédents.

Lemme 4.

Soient u, u_2, v_1, v_2, p, q *des mots de* X^*.
Si $p\, u_1\, q\, u_2 \overset{*}{\underset{C}{\rightarrow}} q\, v_1\, p\, v_2$ *avec* $\mathrm{Alph}(pu_1) \cap \mathrm{Alph}(q) = \emptyset$ *et* $\mathrm{Alph}(p) \cap \mathrm{Alph}(v_1) = \emptyset$, *alors*

i) $u_1u_2 \overset{*}{\underset{C}{\rightarrow}} v_1v_2$

ii) $\forall x \in \mathrm{Alph}(pu_1)$, $\forall y \in \mathrm{Alph}(q)$, $xy \rightarrow yx \in C$

iii) $\forall x \in \mathrm{Alph}(p)$, $\forall y \in \mathrm{Alph}(v_1)$, $xy \rightarrow yx \in C$.

Le dernier lemme de cette série nous donne un moyen plus simple de reconnaître si un mot w se dérive en un mot $w' \in X^*$ par une relation C : il suffira de comparer les projections sur deux lettres de w et de w'.

Notation :

Si $X_1 \subseteq X$, nous noterons ici

$$\Pi_{X_1} = \mathrm{proj}(X, X_1).$$

Lemme 5.

Soient w *et* w' *deux mots de* X^*.

$$w \overset{*}{\underset{C}{\rightarrow}} w' \iff \forall (a,b) \in X \times X,\ \Pi_{\{a,b\}}(w) \overset{*}{\underset{C}{\rightarrow}} \Pi_{\{a,b\}}(w')$$

Preuve.

$\Longrightarrow$) C'est une conséquence du lemme 1.

$\Longleftarrow$) Nous raisonnons par induction sur la longueur de w.
Si $|w| = 0$, le résultat est évidemment vrai. Supposons que pour tout mot de longueur plus petite que n, le résultat soit vrai et considérons un mot w de longueur n+1.
Posons $w = x\, w_1$ avec $x \in X$.

1er Cas. $w' = x\, w_1'$.

Alors,

i) $\forall z, t \neq x$, par hypothèse,

$$\Pi_{\{z,t\}}(w) = \Pi_{\{z,t\}}(w_1) \overset{*}{\underset{c}{\to}} \Pi_{\{z,t\}}(w') = \Pi_{\{z,t\}}(w_1')$$

ii) $\forall z \in X$, $\Pi_{\{x,z\}}(w) = x\Pi_{\{x,z\}}(w_1) \overset{*}{\underset{c}{\to}} \Pi_{\{x,z\}}(w') = x\Pi_{\{x,z\}}(w_1')$

$\Rightarrow$ (lemme 2) $\Pi_{\{x,z\}}(w_1) \overset{*}{\underset{c}{\to}} \Pi_{\{x,z\}}(w_1')$

i) et ii) $\Rightarrow$ (hyp de récurrence) $w_1 \overset{*}{\underset{c}{\to}} w_1'$

$\Rightarrow w = xw_1 \overset{*}{\underset{c}{\to}} xw_1' = w'$

On déduit alors facilement que pour tout mot w de longueur n+1,

$$\left.\begin{array}{l} w = uv \\ w' = uv' \\ \forall (a,b) \in X \times X,\ \Pi_{\{a,b\}}(w) \overset{*}{\underset{c}{\to}} \Pi_{\{a,b\}}(w') \end{array}\right| \Rightarrow w \overset{*}{\underset{c}{\to}} w'$$

2eme Cas. $w' = yw_1'$, $y \neq x$

On peut alors écrire $w = x\, u\, y\, v$ avec $|u|_y = 0$, $u,v \in X^*$ et $w' = y\, u'\, x\, v'$ avec $|u'|_x = 0$, $u', v' \in X^*$.
Soient $w_1 = yx\, uv$ et $w_2 = yx\, u'v'$. Nous allons montrer que $w \xrightarrow[c]{*} w_1$, $w_1 \xrightarrow[c]{*} w_2$, $w_2 \xrightarrow[c]{*} w'$. Par transitivité, on aura donc le résultat cherché.

1) $w = x\,u\,y\,v \overset{*}{\underset{c}{\rightarrow}} w_1 = yx\;uv$?

$\forall z \in \mathrm{Alph}(u) \cup \{x\}$, $\Pi_{\{z,y\}}(w) = \alpha y \beta$ avec $\alpha \subseteq z^*$
et $\Pi_{\{z,y\}}(w') = y\gamma$ avec $\gamma \subseteq \{z,y\}^*$
$\Pi_{\{z,y\}}(w) \overset{*}{\underset{c}{\rightarrow}} \Pi_{\{z,y\}}(w')$ (par hypothèse) $\Rightarrow$ (lemme 2) $zy \rightarrow yz \in C$
$\Rightarrow x\,uy \overset{*}{\underset{c}{\rightarrow}} yx\,u$
$\Rightarrow w \overset{*}{\underset{c}{\rightarrow}} w_1$

2) $w_1 = yx\;uv \overset{*}{\underset{c}{\rightarrow}} w_2 = yx\;u'v'$?

- $\forall z,t \neq x, y$, $\Pi_{\{z,t\}}(w_1) = \Pi_{\{z,t\}}(w) \overset{*}{\underset{c}{\rightarrow}} \Pi_{\{z,t\}}(w') = \Pi_{\{z,t\}}(w_2)$

- $\forall z \neq x, y$, $\Pi_{\{y,z\}}(w) = \Pi_{\{z\}}(u) y \Pi_{\{y,z\}}(v) \overset{*}{\underset{c}{\rightarrow}} y\, \Pi_{\{y,z\}}(u'v')$

  $\Rightarrow$ (lemme 3) $\Pi_{\{z\}}(u)\, \Pi_{\{y,z\}}(v) \overset{*}{\underset{c}{\rightarrow}} \Pi_{\{y,z\}}(u'v')$

  $\Rightarrow y\, \Pi_{\{z\}}(u)\, \Pi_{\{y,z\}}(v) = \Pi_{\{y,z\}}(w_1) \overset{*}{\underset{c}{\rightarrow}} y\Pi_{\{y,z\}}(u'v') = \Pi_{\{y,z\}}(w_2)$

- $\forall z \neq x, y$, $\Pi_{\{x,z\}}(w) = x\, \Pi_{\{x,z\}}(uv) \overset{*}{\underset{c}{\rightarrow}} \Pi_{\{z\}}(u')\, x\, \Pi_{\{x,z\}}(v')$

  $\Rightarrow$ (lemme2) $\Pi_{\{x,z\}}(uv) \overset{*}{\underset{c}{\rightarrow}} \Pi_{\{z\}}(u')\, \Pi_{\{x,z\}}(v')$

  $\Rightarrow$ $x\, \Pi_{\{x,z\}}(uv) = \Pi_{\{x,z\}}(w_1) \rightarrow x\, \Pi_{\{z\}}(u')\, \Pi_{\{x,z\}}(v') = \Pi_{\{x,z\}}(w_2)$

- $\Pi_{\{x,y\}}(w) = x\, \Pi_{\{x\}}(u) y\, \Pi_{\{x,y\}}(v) \overset{*}{\underset{c}{\rightarrow}} y\, \Pi_{\{y\}}(u')\, x\, \Pi_{\{x,y\}}(v')$

  $\Rightarrow$ (lemme 4) $\Pi_{\{x,y\}}(u)\, \Pi_{\{x,y\}}(v) \overset{*}{\underset{c}{\rightarrow}} \Pi_{\{x,y\}}(u')\, \Pi_{\{x,y\}}(v')$

  $\Rightarrow$ $yx\, \Pi_{\{x,y\}}(u)\, \Pi_{\{x,y\}}(v) = \Pi_{\{x,y\}}(w_1) \overset{*}{\underset{c}{\rightarrow}} yx\, \Pi_{\{x,y\}}(u')\, \Pi_{\{x,y\}}(v') =$
  $\Pi_{\{x,y\}}(w_2)$

Donc

$$w_1 = yx\;uv,\; w_2 = yx\;u'v',\; |w_1| = n+1$$

et

$$\forall (a,b) \in X \times X,\; \Pi_{\{a,b\}}(w_1) \overset{*}{\underset{c}{\rightarrow}} \Pi_{\{a,b\}}(w_2).$$

D'après l'étude du premier cas, on peut conclure que $w_1 \overset{*}{\underset{c}{\rightarrow}} w_2$.

3) $w_2 = yx\ u'v' \xrightarrow[c]{*} w' = y\ u'\ x\ v'$?

$\forall z \in Alph(u')$, $\Pi_{\{x,z\}}(w) = x\alpha$ avec $\alpha \in \{x,z\}^*$ et

$\Pi_{\{x,z\}}(w') = \Pi_{\{z\}}(u')\ x\ \Pi_{\{x,z\}}(v')$ et $\Pi_{\{x,z\}}(w) \overset{*}{\underset{c}{\rightarrow}} \Pi_{\{x,z\}}(w')$

$\Rightarrow$ (lemme 2) $\forall z \in Alph(u')$, $xz \rightarrow zx \in C$

$\Rightarrow xu' \overset{*}{\underset{c}{\rightarrow}} u'x$

$\Rightarrow w_2 = yx\ u'\ v' \overset{*}{\underset{c}{\rightarrow}} y\ u'\ x\ v' = w'.$

□

II.B - Factorisation des fonctions de commutation partitionnée.

Soit X un alphabet et $\Pi = \{X_1, X_2, \ldots, X_n\}$ une partition de X. Nous avons vu qu'à une telle partition, nous pouvons associer une relation de commutation, C_π, définie par $(x,y) \in C_\pi$ si et seulement si il existe i et $j \in \{1, 2, \ldots, n\}$, $i \neq j$ tels que $x \in X_i$ et $y \in X_j$, et une fonction de commutation notée f telle que :

$$f_\pi(w) = \{w',\ w \xrightarrow[c_\pi]{*} w'\} = \{proj(w,X_1) \sqcup\!\sqcup\ proj(w,X_2) \sqcup\!\sqcup \ldots \\ \ldots \sqcup\!\sqcup proj(w, X_n)\}$$

Ce type de fonction a un lien très étroit avec un opérateur plus connu et déjà étudié en théorie des langages : l'opérateur TS qui à un mot w associe le langage $TS(w) = \{w \sqcup\!\sqcup \bar{\bar{w}}\}$. En effet, soit $w \in X^*$ et h l'homomorphisme défini sur X^* par $h(x) = x\bar{x}$, $\forall x \in X$. Soit $\Pi = \{X, \bar{X}\}$ la partition de $X \cup \bar{X}$.
Alors il est facile de voir que $TS(w) = f_\pi(h(w))$.
Dans ce chapitre, nous allons montrer que, réciproquement, toute fonction de commutation partitionnée peut s'écrire comme la composition d'homomorphismes (linéairement effaçants), et de l'opérateur TS, ce que nous noterons

$$P \subseteq \{H^{lin}, TS\}^*.$$

Lemme 6.

Soit f *une fonction de commutation partitionnée définie sur* X *associée à une partition* $\{X_1, X_2,\dots,X_n\}$ *de* X. *Alors, il existe* (n-1) *fonctions de commutation 2-partitionnée (associées à des partitions de* X *à 2 éléments) telles que* $f = f_{n-1} \circ \dots \circ f_1$.

Preuve.

Le résultat est vrai pour n = 2.

Supposons qu'il est vrai pour une fonction de commutation (n-1) partitionnée.

Soit $\Pi = \{X_1, X_2,\dots,X_{n-2}, X_{n-1} \cup X_n\}$, g la fonction associée. Alors, il existe n-2 fonctions de commutation 2-partitionnée $f_1, f_2,\dots,f_{n-2}$ telles que $g = f_{n-2} \circ \dots \circ f_1$.

Soit h_1 la projection de X sur $X_1 \cup X_2 \cup \dots \cup X_{n-1}$ et h_2 la projection de X sur X_n. Il est facile de voir que

$$f(w) = \{h_1(w') \sqcup h_2(w'),\ w' \in g(w)\}$$

Soit f_{n-1} la fonction de commutation 2-partitionnée associée à la partition $\{X_1 \cup X_2 \cup \dots X_{n-1}, X_n\}$ de X. Alors :

$$f(w) = \{f_{n-1}(w'),\ w' \in g(w)\}$$

$$= f_{n-1} \circ f_{n-2} \circ \dots \circ f_1(w).$$

□

Nous allons maintenant montrer qu'une fonction de commutation 2-partitionnée peut être simulée par l'opération TS et un homomorphisme linéairement effaçant.

Proposition 1.

Soit X *un alphabet,* $\pi = \{X_1, X_2\}$ *une partition de* X*, et* f *la fonction de commutation partitionnée associée à* π*. Alors, il existe un homomorphisme* h *, linéairement effaçant sur* $X \cup \bar{X}$*, tel que* $\forall w \in X^*$, $f(w) = h \circ TS(w)$.

Preuve.

Soit h l'homomorphisme défini sur $X \cup \bar{X}$ par

$$h(x) = x \text{ si } x \in X_1,\ h(x) = \varepsilon \text{ si } x \in X_2,$$

$$h(\bar{x}) = \varepsilon \text{ si } \bar{x} \in \bar{X}_1,\ h(\bar{x}) = x \text{ si } \bar{x} \in \bar{X}_2.$$

Il est clair que :

$$f(w) = \mathrm{proj}(w, X_1) \sqcup\!\sqcup \mathrm{proj}(w, X_2)$$

$$= h(w \sqcup\!\sqcup \bar{w}) = h(TS(w)).$$

$\square$

II.C - Factorisation des fonctions de commutation partielle.

Nous avons vu que l'image d'un mot w de X^* par une fonction de commutation f associée à une partition de X pouvait s'exprimer facilement en utilisant les morphismes et l'opération de shuffle, qui sont des opérations classiques.

Pour étudier les fonctions de commutation partielle dans le cadre de la théorie des langages, il semble donc intéressant de pouvoir décrire l'image d'un mot w de X^* par une fonction de commutation partielle au moyen des commutations partitionnées et des transductions rationnelles fidèles.

Nous allons montrer qu'en fait une fonction de commutation partielle peut être simulée par un homomorphisme, un homomorphisme inverse et une fonction de commutation partitionnée.

Proposition 2.

Soit f *une fonction de commutation partielle définie sur l'alphabet* X. *Alors, il existe un homomorphisme marqué* h *et une fonction de commutation partitionnée* f' *tels que* $f = h^{-1} \circ f' \circ h$.

Preuve.

Soit $C \subseteq X \times X$ la relation de commutation associée à f.
Soit $Z = X \cup Y$ où $Y = \{(x,y) \in X \times X \ / \ x \neq y, (x,y) \notin C\}$. Définissons l'homomorphisme h sur X^* par :

$$\forall x \in X, \ h(x) = x(x,y_1)\ldots(x,y_t)$$

avec

$$\{y_1,\ldots,y_t\} = \{y \in X/(x,y) \in Y\}.$$

Soit π la partition de Z définie par :

$$\{x\} \in \Pi, \ \forall x \in X \text{ et } \{(x,y), (y,x)\} \in \Pi, \ \forall (x,y) \in Y.$$

Nous noterons c_π la relation de commutation correspondant à π et f la fonction associée à c_π.

La propriété suivante est une conséquence immédiate de la construction :

$$h(x)\ h(y) \xleftrightarrow[c_\pi]{*} h(y)\ h(x) \iff (x,y) \in C.$$

Nous allons maintenant montrer que $f = h^{-1} \circ f \circ h$.

$\subseteq$) Soit $w \in X^*$ et $w' \in f(w)$: $w \xleftrightarrow[c]{*} w'$.

Nous allons raisonner par induction sur la longueur de cette dérivation, ℓ, pour montrer que $h(w) \xleftrightarrow[c_\pi]{*} h(w')$.

Si $\ell = 0$, $w' = w$, $h(w') = h(w)$ et le résultat est évident.

Supposons le résultat vrai pour toute dérivation de longueur plus petite que ℓ, et considérons une dérivation de longueur $\ell+1$:

$$w \overset{\ell+1}{\underset{c}{\longleftrightarrow}} w' \Longrightarrow w \overset{\ell}{\underset{c}{\longleftrightarrow}} w_1 = u\ xy\ v \underset{c}{\longleftrightarrow} uyxv = w'$$

où

$$(x,y) \in C.$$

Par hypothèse de récurrence, $h(w) \overset{*}{\underset{c\pi}{\longleftrightarrow}} h(w_1) = h(u)\ h(x)\ h(y)\ h(v)$ et en utilisant la propriété énoncée plus haut,

$$h(w_1) = h(u_1)\ h(x)\ h(y)\ h(v) \overset{*}{\underset{c\pi}{\longleftrightarrow}} h(u)\ h(y)\ h(x)\ h(v) = h(w').$$

Donc

$$\forall w \in X^*,\ f(w) \subseteq h^{-1} \circ f' \circ h(w).$$

$\supset$) Pour montrer l'inclusion inverse, nous utilisons une récurrence sur la longueur de w.
Si $|w| = 0$, le résultat est trivial.
Posons $w = xu$, où $x \in X$. Soit $w' \in h^{-1} \circ f' \circ h(w)$, donc $h(w) \overset{*}{\underset{c\pi}{\longleftrightarrow}} h(w')$.
Nous pouvons alors écrire

$$h(w') = u_1xu_2$$

avec $|u_1|_x = 0$, $u_1 = h(w_1')$, $xu_2 = h(x)\ h(w_2')$ et $\mathrm{Alph}\ (h(x)) \cap \mathrm{Alph}\ (u_1) = \emptyset$.

En utilisant le lemme 2 du chapitre précédent, on a donc

$$h(u) \overset{*}{\underset{c\pi}{\longleftrightarrow}} h(w_1')\ h(w_2') \text{ et } \mathrm{Alph}\ (h(x)) \times \mathrm{Alph}\ (u_1) \subseteq c\pi.$$

Par hypothèse de récurrence, on déduit que $u \overset{*}{\underset{c}{\longrightarrow}} w_1'\ w_2'$ et par construction de $c\pi$,

$$\mathrm{Alph}\ (h(x)) \times \mathrm{Alph}\ (u_1) \subseteq c\pi \Longrightarrow \{x\} \times \mathrm{Alph}\ (w_1') \subseteq C.$$

Donc

$$xw_1' \overset{*}{\underset{c}{\longleftrightarrow}} w_1'x.$$

Alors,

$$w = xu \overset{*}{\underset{c}{\longleftrightarrow}} xw_1'w_2' \overset{*}{\underset{c}{\longleftrightarrow}} w_1'xw_2' = w' \text{ et } w' \in f(w).$$

$\square$

Soit h un homomorphisme marqué de X^* dans $(X \cup Y)^*$. Alors, considérons le langage rationnel $R = h(X^*)$ et la projection θ de $(X \cup Y)^*$ sur X^*, qui est ε-limitée sur R. Il est facile de constater que $h = (\cap R) \circ \theta^{-1}$ et $h^{-1} = \theta \circ (\cap R)$.

Nous pouvons alors énoncer la proposition 2 différemment :

Proposition 3.

Soit f *une fonction de commutation partielle définie sur l'alphabet* X. *Alors il existe une fonction de commutation partitionnée* f', *un homomorphisme marqué* h, *un langage rationnel* R *et une projection* θ ε*-limitée sur* R *tels que :*

$$f = \theta \circ (\cap R) \circ f' \circ h = h^{-1} \circ f' \circ (\cap R) \circ \theta^{-1}.$$

II.D - Factorisation des fonctions de semi-commutation.

Nous allons maintenant nous attacher au cas des fonctions de semi-commutation, en montrant que de telles fonctions peuvent être simulées par des transductions rationnelles fidèles et des fonctions de commutation partielle. Le problème semble plus délicat puisqu'il faut réaliser une relation non symétrique par des relations symétriques, et on ne pourra pas trouver une factorisation aussi simple que précédemment. En effet, considérons la fonction de commutation f associée à la relation $ba \to ab$, et supposons qu'il existe deux homomorphismes h et g et une fonction de commutation partielle f' associée à une relation C, tels que :

$$f = h^{-1} \circ f' \circ g.$$

Alors on peut remarquer que :

i) $f(a) = a \Rightarrow g(a) \overset{*}{\underset{C}{\Longleftrightarrow}} h(a)$

ii) $f(b) = b \Rightarrow g(b) \overset{*}{\underset{C}{\Longleftrightarrow}} h(b)$

iii) $ab \in f(ba) \Rightarrow g(ba) = g(b)\, g(a) \underset{C}{\Longleftrightarrow} h(ab) = h(a)\, h(b).$

Alors montrons que ba $\in$ f(ab) :

$$g(ab) = g(a)\, g(b) \overset{*}{\underset{C}{\longleftrightarrow}} h(a)\, g(b) \qquad \text{(en utilisant la remarque i))}$$

$$h(a)\, g(b) \overset{*}{\underset{C}{\longleftrightarrow}} h(a)\, h(b) \qquad \text{(voir ii))}$$

$$h(a)\, h(b) \overset{*}{\underset{C}{\longleftrightarrow}} g(b)\, g(a) \qquad \text{(voir iii))}$$

$$g(b)\, g(a) \overset{*}{\underset{C}{\longleftrightarrow}} h(b)\, h(a) = h(ba) \qquad \text{(voir i) et ii))}$$

et ce résultat est en contradiction avec la définition de f.
Cette fonction est la plus simple des fonctions de semi-commutation non symétriques. La construction qu'on utilisera pour trouver la succession d'opérations qui simulent une fonction de semi-commutation sera basée sur un raisonnement par récurrence sur le nombre de commutations non symétriques de la relation associée à la fonction.

Dans une première partie, nous allons donc décrire la simulation de la fonction f définie plus haut, puis, dans la deuxième partie, nous étudierons le cas général.

1) Simulation de la fonction f : ba → ab sur les mots d'un alphabet X quelconque.

Après avoir donné quelques propriétés de la fonction f (lemmes 7 et 8), nous simulerons f sur des mots d'une forme particulière (lemme 9) puis sur des mots quelconques (lemmes 10 et 11).

Lemme 7.

Soit X = {a,b} *et* f *la fonction associée à la relation* C : ba → ab.
Alors, $\forall w \in X$, $bwa \overset{*}{\underset{C}{\longrightarrow}} awb$.

Preuve.

Raisonnons par induction sur la longueur de w.
Si $|w| = 0$, alors $ba \underset{C}{\longrightarrow} ab$ par définition de C.
Supposons que $w = aw'$, $w' \in X^*$. Alors $bwa = baw'a \underset{C}{\longrightarrow} abw'a \overset{*}{\underset{C}{\longrightarrow}} aaw'b = awb$ en utilisant l'hypothèse de récurrence.

De même si $w = bw'$, $w' \in X^*$, on a alors :

$$bwa = bbwa \xrightarrow[c]{*} baw'b \text{ (HR) } \xrightarrow[c]{} abw'b = awb.$$

□

Lemme 8.

Soit $X = \{a,b\}$. *Pour tout couple de mots* (w, w') *de* $X^* \times X^*$,

$$w \xrightarrow[c]{*} w' \iff \begin{cases} \text{i) } w' \in \mathrm{Com}(w) \textit{ et} \\ \text{ii) } w = w_1w_2,\ w' = w'_1 w'_2,\ |w_1| = |w'_1| \Rightarrow |w'_1|_a \geq |w_1|_a. \end{cases}$$

(c *désigne toujours la relation* $ba \to ab$). *Nous noterons* $w \underset{a}{\leq} w'$ *la propriété ii).*

Preuve.

$\Rightarrow$ On fait une récurrence sur la longueur de la dérivation $w \xrightarrow[c]{*} w'$ qu'on notera ℓ. On peut noter que i) est toujours vérifié.

- Si $\ell = 0$ $w' = w$ et le résultat est évident.

- Pour une dérivation de longueur $\ell+1$:

$$w \xrightarrow[c]{\ell} w''=ubav \xrightarrow[c]{} uabv = w'.$$

Par hypothèse de récurrence, le résultat est vrai pour w'' : $w \underset{a}{\leq} w''$.
Il reste donc à montrer que $w'' \underset{a}{\leq} w'$. Soit $w'' = w_1w_2$, $w' = w'_1w'_2$ avec $|w_1| = |w_2|$

Si $w_1 \in FG(u) \Rightarrow w'_1 = w_1 \Rightarrow |w'_1|_a \geq |w_1|_a$.

Si $w_1 = ub \Rightarrow w'_1 = ua \Rightarrow |w'_1|_a = |w_1|_a+1 \geq |w_1|_a$

Si $w_1 = uabv_1$ avec $v_1 \in FG(v) \Rightarrow w'_1 = ubav_1 \Rightarrow |w_1|_a = |w'_1|_a$.

Donc $w'' \underset{a}{\leq} w'$, et on déduit que $w \underset{a}{\leq} w'$.

$\Leftarrow$ On fait une récurrence sur la longueur de w.

- Si $|w| = 0 \Rightarrow w = w' = \varepsilon \Rightarrow w \xrightarrow[c]{*} w'$.

- Posons $w = xw_1$, et étudions le cas où $x = a$ puis celui où $x = b$.

<u>1er Cas.</u> $w = aw_1 \Rightarrow w' = aw_1'$ car la propriété ii) est vraie pour le facteur gauche de longueur 1.

$\Rightarrow w_1' \in Com(w_1)$ et $w_1' \leqq_a w_1$

$\Rightarrow w_1 \xrightarrow[c]{*} w_1'$ (hyp. de récurrence)

$\Rightarrow w = aw_1 \xrightarrow[c]{*} w' = aw_1'$.

<u>2ème Cas.</u> $w = bw_1$. Regardons la première lettre de w'.

1) $w' = bw_1'$. Le raisonnement est le même qu'au 1er Cas.

2) $w' = aw_1'$.

Si il existe un découpage $w = uv$, $w' = u'v'$ avec $u' \in Com(u)$, $u' \in X^+$ et $v' \in Com(v)$, $v' \in X^+$, il est alors facile de voir que $u \leqq_a u'$ et $v \leqq_a v'$. En appliquant l'hypothèse de récurrence, on a donc

$$u \xrightarrow[c]{*} u' \text{ et } v \xrightarrow[c]{*} v',$$

et ainsi

$$w = uv \xrightarrow[c]{*} u'v \xrightarrow[c]{*} u'v' = w'.$$

Si on ne peut pas trouver de tels découpages, on peut alors écrire

$$w = bua \text{ et } w' = avb.$$

(On peut écarter le cas où w = bub, car alors on aurait $w' = ava$ et ainsi $|bu|_a = |w|_a > |av|_a = |w'|_a - 1$ ce qui est impossible par hypothèse). Montrons alors que $u \leqq_a v$: si il existe un couple $(u_1, v_1) \in FG(u) \times FG(v)$ vérifiant $|u_1| = |v_1|$ et $|u_1|_a > |v_1|_a$, alors $|bu_1|_a = |u_1|_a \geq |av_1| = |v_1|_a+1 \geq |bu_1|_a$ par hypothèse. Donc $|bu_1|_a = |av_1|_a$. Comme $|bu_1| = |av_1|$, on a $bu_1 \in Com(av_1)$, on pouvait alors trouver un découpage.
Ainsi $u \in Com(v)$ et $u \leqq_a v$. En appliquant l'hypothèse de récurrence, on a donc $u \xrightarrow[c]{*} v$.

$$u \xrightarrow[c]{*} v \Rightarrow w = bua \xrightarrow[c]{*} bva \xrightarrow[c]{} avb = w' \quad \text{(lemme 6)}.$$

□

Une conséquence assez immédiate de ce lemme est le corollaire suivant :

Corollaire 1.

$f((ab)^*) = D_1'^*$ *où* $D_1'^*$ *est le langage de semi-Dyck défini sur* $\{a,b\}^*$

Nous allons maintenant simuler la relation $\bar{c}c \to c\bar{c}$ sur des mots de la forme $(c\bar{c})^i$, $i \geq o$.

Notation.

Nous noterons $\{H, H^{-1}, CP\}^*$ l'ensemble des fonctions obtenues en composant des homomorphismes, des homomorphismes inverses et des fonctions de commutation partielle .

Lemme 9.

Soient $Z = \{c, \bar{c}\}$ *et* f_1 *la fonction de semi-commutation associée à la relation* $c = \{c\bar{c} \to \bar{c}c\}$. *Alors, il existe* $g \in \{H, H^{-1}, CP\}^*$ *telle que* $g = f_1$ *sur* $(c\bar{c})^*$.

Preuve.

On utilise ici la propriété suivante [14] :

$$f_1((c\bar{c})^i) = D_1'^i = \{w \in D_1'^*, |w| = 2i\}$$

$$= C_1^* \sqcup\!\sqcup C_1^* \cap \{c, \bar{c}\}^{2i} \text{ où } C_1 = \{c^n\bar{c}^n, n \geq 0\}.$$

Nous allons donc construire ce langage, à partir de $(c\bar{c})^i$ en n'utilisant que les opérations permises : des homomorphismes non effaçants, des homomorphismes inverses et des commutations partielles (ou partitionnées).

Comme l'intersection avec un rationnel de la forme R^* où R est un langage rationnel peut être simulée par des homomorphismes et des homomorphismes inverses (voir [13]), on peut utiliser cette opération ici.

Soit $h : \{c, c', \bar{c}, \bar{c}'\}^* \to \{c, \bar{c}\}^*$ défini par $h(c) = h(c') = c$ et $h(\bar{c}) = h(\bar{c}') = \bar{c}$

$$L_1 = h^{-1}((c\bar{c})^i) \cap (c\bar{c} + c'\bar{c}')^* = \left\{ \begin{array}{l} (c\bar{c})^{i_1} (c'\bar{c}')^{j_1} \dots (c\bar{c})^{i_n} (c'\bar{c}')^{j_n}, \\ i_k \geq 0,\ j_k \geq 0,\ \sum_{k=1}^{n} (i_k + j_k) = i \end{array} \right\}$$

Soit f' la fonction de commutation associée à la relation

$$C' = \{ (c,\bar{c}),\ (c',\bar{c}') \} \text{ définie sur } \{ c, \bar{c}, c', \bar{c}' \}$$

et R le rationnel $(c^+\bar{c}^+ + c'^+\bar{c}'^+)^*$

$$L_2 = f'(L_1) \cap R = \left\{ \begin{array}{l} c^{i_1}\bar{c}^{i_1} c'^{j_1}\bar{c}'^{j_1} \dots c^{i_n}\bar{c}^{i_n} c'^{j_n}\bar{c}'^{j_n},\ i_k > 0,\ j_k > 0 \\ \sum_{k=1}^{n} (i_k + j_k) = i \end{array} \right\}$$

$$f'(L_1) \cap R \subseteq (C_1 C_1') \quad \text{où } C_1' = \{c'^n \bar{c}'^n,\ n \geq 0\}.$$

Soit g_c la fonction de commutation associée à la partition de

$$\{c, c', \bar{c}, \bar{c}'\} : \{\{c, \bar{c}\}, \{c', \bar{c}'\}\}.$$

Alors, il est clair qu'aux marques ' ' ' près, $g_c(L_2)$ est égal à $f_1(c\bar{c})^i)$.
Il ne reste qu'à appliquer l'homomorphisme h défini plus haut à $g_c(L_2)$ pour obtenir exactement le langage désiré.

□

<u>*Lemme 10.*</u> *Soient $\pi = \{Z_1, Z_2\}$ une partition de Z, C_π la relation de commutation partitionnée associée à π, f_π la fonction associée et C une relation de commutation sur Z telle que $C = C_\pi \cup C'$, avec $C' \subseteq Z_2 \times Z_2$. Notons f la fonction associée à C, et f' la fonction associé à C'.*

*S'il existe $g \in \{H, H', CP\}^*$ vérifiant $f' = g$ sur $A_2 \subseteq Z_2^*$
Alors, on peut trouver $g' \in \{H, H^{-1}, CP\}^*$ telle que $f = f_\pi \circ g' \circ f_\pi$ sur $A_2 \sqcup Z_1^*$.*

Preuve.

Construction de g' définie sur Z à partir de g qui est définie sur Z_2 :

Posons $g = g_n \circ \ldots \circ g_1$, avec $g_i \in \{H, H^{-1}, CP\}$, g_i étant défini sur X_i^* et à valeur dans $2^{X_{i+1}^*}$ ($X_1 = Z_2$, $X_{n+1} = Z_2$). On peut supposer que les X_i sont tous disjoints de Z_1.

Nous allons étendre chacune de ces fonctions à Z, de la manière suivante :

A $g_i : X_i^* \to X_{i+1}^*$ associons $g_i' : (X_i \cup Z_1)^* \to (X_{i+1} \cup Z_1)^*$ tel que :

i) Si $g_i \in H$: alors $g'_{i|X_1} = g_i$ et $g'_{i|Z_1}$ = Identité

ii) Si $g_i \in H^{-1}$, $g_i = h_i^{-1}$, avec $h_i \in H$, $h_i : X_{i+1}^* \to X_i^*$.
Alors $g_i' = h_i'^{-1}$ où h_i' est défini par $h'_{i|X_{i+1}} = h_{i|X_i+1}$ et $h'_{i|Z_1}$ = Identité.

iii) Si $g_i \in CP$, g_i est associée à une relation de commutation partielle $C_i \subseteq X_i \times X_i$ et $X_{i+1} = X_i$. Alors g_i' sera la fonction de commutation partielle associée à la relation $C_i' = C_i \subseteq (X_i \cup Z_1) \times (X_i \cup Z_1)$.

Posons $g' = g_n' \circ \ldots \circ g_1'$.
Il est clair que :

1) $\forall u_1 \in Z_1^*$, $g'(u_1) = u_1$

2) $\forall u_2 \in Z_2^*$, $g'(u_2) = g(u_2)$

3) $\forall v_1 \in Z_1^*$, $\forall i \in \{1,\ldots,n\}$, $\forall v_2 \in X_i^*$,
$g_i'(v_1 v_2) = g_i'(v_1)\, g_i'(v_2)$ puisqu'on est toujours sur des alphabets disjoints ($= v_1\, g_i'(v_2) = v_1\, g_i(v_2)$).
Donc, $\forall (u_1, u_2) \in Z_1^* \times Z_2^*$, $g'(u_1 u_2) = g'(u_1)\, g'(u_2) = u_1\, g(u_2)$.

Montrons que $f \subseteq f_\pi \circ g' \circ f_\pi$ sur $Z_1^* ⧢ A_2$.

Soit $w \in Z_1^* \sqcup\!\sqcup\; A_2 \Rightarrow w \in u_1 \sqcup\!\sqcup\; u_2$, $u_1 \in Z_1^*$, $u_2 \in A_2$.
Soit $w' \in f(w) \Rightarrow w \overset{*}{\underset{c}{\to}} w'$

$\Rightarrow (C_\pi \subseteq C)\ w \xrightarrow[c_\pi]{*} u_1u_2 \xrightarrow[c_\pi]{*} w \qquad \xrightarrow[c\pi]{*} w' \xrightarrow[c_\pi]{} u_1'u_2' \xrightarrow[c_\pi]{*} w'$

où u_1' (resp. u_2') désigne la projection de w' sur Z_1 (resp. Z_2)

$\Rightarrow u_1u_2 \overset{*}{\underset{c}{\to}} u_1'u_2'$

$\Rightarrow$ (lemme 1) $u_1 \overset{*}{\underset{c}{\to}} u_1'$ et $u_2 \overset{*}{\underset{c}{\to}} u_2'$

$\Rightarrow u_1' = u_1$ et $u_2 \xrightarrow[c']{*} u_2'$

$\Rightarrow u_1' = u_1$ et $u_2' \in g(u_2)$ car $u_2 \in A_2$.

$\Rightarrow u_1' \in g'(u_1) = u_1$ et $u_2' \in g'(u_2)$

$\Rightarrow u_1'u_2' \in g'(u_1)\ g'(u_2) \subseteq g'(u_1u_2) \subseteq g'(f_\pi(w))$

$\Rightarrow w' \in f_\pi(u_1'u_2') \subseteq f_\pi \circ g' \circ f_\pi(w)$

Il reste à montrer l'inclusion inverse.

Soit $w \in Z_1^* \sqcup\!\sqcup\; A_2 \Rightarrow w \in u_1 \sqcup\!\sqcup\; u_2$, $u_1 \in Z_1^*$, $u_2 \in A_2$;
Soit $w' \in f_\pi \circ g' \circ f_\pi(w) \Rightarrow w' \in f_\pi \circ g'(w_1)$, avec $w_1 \in f_\pi(w)$,
$w_1 = x_1y_1\ x_2y_2 \ldots x_ny_n$, $\forall i\ xi \in Z_1^*$, $y_i \in Z_2^*$, $x_1x_2 \ldots x_n = u_1$, $u_1y_2 \ldots y_n = u2$.

Alors $\quad g'(w_1) = g'(x_1)\ g'(y_1) \ldots g'(x_n)\ g'(y_n) = x_1\ g(y_1) \ldots x_n\ g(y_n)$
$\subseteq x_1x_2 \ldots x_n \sqcup\!\sqcup\; g(y_1)g(y_2)\ldots g(y_n) \subseteq x_1\ x_2 \ldots x_n \sqcup\!\sqcup\; g(y_1\ldots y_n)$
$\subseteq f_\pi(x_1 \ldots x_n\ f'(y_1 \ldots y_n))$ car $f' = g$ sur $A_2 \subseteq f(u_1u_2)$

$\Rightarrow f_\pi(g'(w_1) \subseteq f_\pi(f(u_1u_2)) = f(u_1u_2) \subseteq f(w)$ car $u_1u_2 \in f_\pi(w)$.

□

Soit $X = \{a,b\}$ et f la fonction associée à la relation de commutation $ba \to ab$. Nous allons simuler f sur un mot quelconque w de $\{a,b\}^*$.

Lemme 11.

La fonction f *associée à la relation* $C = \{ba \to ab\}$ *définie sur* $\{a,b\}$ *appartient à* $\{H, H^{-1}, CP\}^*$.

Preuve.

Soient $w \in \{a,b\}^*$ et $h : \{a,b\}^* \to \{a,b,c,\bar{c}\}^*$ défini par $h(b) = b$ et $h(a) = ac\bar{c}$.

Soit f_1 la fonction de commutation associée à la relation $\bar{c}c \to c\bar{c}$ définie sur $\{c,\bar{c}\}$.

Nous allons d'abord construire à partir de $h(w)$ un langage $L = \{w' \in \{a, b, c, \bar{c}\}^* / proj(w', \{a,b\}) = w \text{ et } proj(w', \{c,\bar{c}\}) \in f_1((c\bar{c})^*)\}$. Intuitivement, la place des lettres $\bar{c}$ représentera la place des a dans le mot w, et les lettres c représenteront les lettres a dans l'image par f de w.

Soit π la partition de $\{a, b, c, \bar{c}\}$ définie par $\pi = \{\{a,b\}, \{c,\bar{c}\}\}$

Soient c_π la relation associée et f_π la fonction associée à π. D'après le lemme 9, on peut construire une fonction $g \in \{H, H^{-1}, CP\}$ telle que $g = f_1$ sur $(c\bar{c})^*$.

Donc (lemme 10), on peut construire :

$$g' \in \{H, H^{-1}, CP\}^*$$

telle que $f_\pi \circ g' \circ f_\pi = f'$, où f' est la fonction associée à la relation définie sur $\{a, b, c, \bar{c}\}$ par

$$\{xy \longleftrightarrow yx \ \forall (x,y) \in C_\pi\} \cup \{\bar{c}c \to c\bar{c}\}$$

Donc

$$f_\pi \circ g' \circ f_\pi \circ h(w) = \{w' \in \{a,b,c,\bar{c}\} / proj(w',\{a,b\}) = w,$$

$$proj(w', \{c,\bar{c}\}) \in f_1(c\bar{c})^i), \ i = |w|_a\} = L$$

Soit $R = (a\bar{c} + b + c)^*$.

A partir de $L \cap R$, il ne reste qu'à effacer les lettres a et $\bar{c}$ et transformer les $\hat{c}$ en $\hat{a}$ pour obtenir $f(w)$.
Au lieu d'utiliser un homomorphisme effaçant, on a le même résultat en appliquant à $L \cap R$ la fonction $f_{\pi'}$ avec $\pi' = \{\{a, \bar{c}\}, \{b, c\}\}$ suivi de h^{-1} (h défini plus haut).

Montrons que $f(w) = h^{-1} \circ f_{\pi'}(L \cap R)$

$\supseteq$ Soit $w' \in h^{-1} \circ f_{\pi'}(L \cap R)$

Posons $w = uv$, $w' = u'v'$ avec $|u| = |u'|$. Il faut montrer que $|u'|_a \geq |u|_a$. En appliquant le lemme 8, on aura le résultat :

$$w' \in h^{-1} \circ f_{\pi'}(L \cap R) \Rightarrow \text{ il existe } w'' \in L \cap R \text{ tel que}$$

$h^{-1} \circ f_{\pi'}(w'') = w' \Rightarrow w' = \alpha(\text{proj}(w'', \{b,c\}))$ avec $\alpha(b) = b$ et $\alpha(c) = a$

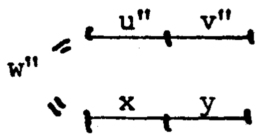
$w'' \in w \sqcup\!\sqcup\, w_1$, $w_1 \in f_1((c\bar{c})^i)$, $i = |w|_a$

Posons $w'' = u''v''$ avec $\alpha(\text{proj}(u'', \{b,c\})) = u'$.

Il faut alors montrer que $|u''|_c \geq |u|_a$

$\text{Proj}(w'', \{a,b\}) = w \Rightarrow w'' = xy$, avec $\text{proj}(x, \{a,b\}) = u$ et $|x|_a = |x|_{\bar{c}}$.

$\text{Proj}(w'', \{c,\bar{c}\}) \in f_1((c\bar{c})^i) \Rightarrow |x|_c \geq |x|_{\bar{c}} = |x|_a = |u|_a$



$|x|_a + |x|_b = |u| = |u'| = |u''|_c + |u''|_b$

et $|x|_c \geq |x|_a$

$\Rightarrow x = u''\alpha$

$\Rightarrow |x|_b \geq |u''|_b$

$\Rightarrow |x|_a = |u|_a \leq |u''|_c = |u'|_a$

$\subseteq$ Soit $w' \in f(w)$. Nous allons construire un mot w_1 de $D_1'^{*}$ sur $\{c,\bar{c}\}$ tel que $w_{1\,c} = w_a$ et

$$\begin{cases} |w_1|_c = |w|_a \text{ et} \\ w' \in h^{-1} \circ f_{\pi'} (w \sqcup\!\sqcup w_1 \cap (a\bar{c} + b + c)^*) \end{cases}$$

<u>1ère étape</u> : construction d'un mot $\alpha \in \{c, \bar{c}, a\}^*$.

α se construit en $|w|$ étapes, en regardant à la $n^{\text{ième}}$ étape la $n^{\text{ième}}$ lettre de w et la $n^{\text{ième}}$ lettre de w'.

Au départ, $\alpha = \varepsilon$.

Si la $n^{\text{ième}}$ lettre de w est 'a' on ajoute 'a' au mot déjà construit,
et si la $n^{\text{ième}}$ lettre de w' est 'a'

Si la $n^{\text{ième}}$ lettre de w est 'b' on n'ajoute rien au mot déjà construit,
et si la $n^{\text{ième}}$ lettre de w' est 'b'

Si la $n^{\text{ième}}$ lettre de w est 'a' on ajoute '$\bar{c}$' au mot déjà construit
et si la $n^{\text{ième}}$ lettre de w' est 'b'

Si la $n^{\text{ième}}$ lettre de w est 'b' on ajoute 'c' au mot déjà construit
et si la $n^{\text{ième}}$ lettre de w' est 'a'

Le mot α obtenu vérifie donc :

i) $|\text{proj}(\alpha, \{a, \bar{c}\})| = |w|_a$

ii) $|\text{proj}(\alpha, \{a, c\})| = |w'|_a$

iii) et pour tout découpage $\alpha = \alpha_1\alpha_2$, on peut associer les découpages $w = uv$, $w' = u'v'$ avec $|u| = |u'|$ tels que $|\alpha_1|_{a,c} = |u'|_a$ $|\alpha_1|_{a,\bar{c}} = |u|_a$.

Comme $w' \in f(w) \Rightarrow$ (lemme 8) $|u'|_a \geq |u|_a$

$\Rightarrow |\alpha_1|_{\bar{c}} \geq |\alpha_1|_c$

$\Rightarrow \text{proj}(\alpha, \{c, \bar{c}\}) \in D_1'^{*}$.

2ème étape : dans α_1 on remplace les occurrences de a par $c\bar{c}$ et on obtient le mot w_1 vérifiant

$$|w_1|_c = |w'|_a \text{ et } |w_1|_{\bar{c}} = |w|_a$$

Donc, les mots du langage $w \sqcup\!\sqcup\, w_1 \cap (a\bar{c} + b + c)^*$ sont générés par la suite d'opérations décrites dans la première partie.
Et par construction, il est clair que $w' \in h^{-1} \circ f_{\pi'}\, (w \sqcup\!\sqcup\, w_1 \cap (a\bar{c}+b+c)^*)$.

□

Nous allons maintenant étudier le cas général d'une fonction de semi-commutation définie sur un alphabet quelconque.

Dans le premier lemme qui suit, nous allons passer d'une relation (et donc d'une fonction) quelconque à une relation qui est encore de semi-commutation mais qui a des propriétés très particulières ; l'idée sera en fait de pouvoir associer à un mot w de X^*, l'ensemble de ses projections sur tout couple de deux lettres de X.

Le deuxième lemme nous montrera comment décomposer une relation du type de celle définie au lemme précédent en relation *"plus petites"* au sens du nombre d'éléments qu'elles contiennent.

Et enfin, par un raisonnement par récurrence, nous montrerons que de telles relations sont simulables.

<u>*Lemme 12.*</u>

Soient $X = \{x_1,\ldots,x_n\}$, $C \subseteq X \times X$ *une relation de commutation, et la fonction associée.*

Soient $Y = \{(xi, xj),\ i \neq j\}$ *et* h *l'homomorphisme défini sur* X^* *par* $h(x_i) = (x_i, x_1)(x_i, x_2)\ldots(x_i, x_{i-1})(x_i, x_{i+1})\ldots(x_i, x_n)$.

Soient $\pi = \{\{(x_i, x_j), (x_j, x_i)\}, \; i \neq j\}$ *la partition de* Y, *et* C_π *la relation de commutation partitionnée associée à* π.

Définissons alors sur Y *la relation de commutation* C' *par*

$$\{\alpha\beta \longleftrightarrow \beta\alpha, \; (\alpha, \beta) \in C_\pi\}$$

$$\cup$$

$$\{(x_i, x_j)(x_j, x_i) \to (x_j, x_i)(x_i, x_j), \; x_i x_j \to x_j x_i \in C\}$$

Notons f' *la fonction associée à* C'. *Alors*

$$f = h^{-1} \circ f' \circ h.$$

<u>Preuve</u>.

$* f \subseteq h^{-1} \circ f' \circ h.$

Par définition de h et de C', il est clair que

$$\forall x, y \in X, \; xy \underset{C}{\to} yx \Rightarrow h(x)\, h(y) \xrightarrow[C']{*} h(y)\, h(x)$$

Alors, par une induction facile sur le nombre de dérivations qui fait passer d'un mot w de X^* à $w' \in f(w)$, on conclut de manière évidente que $h(w') \in f'(h(w))$.

$* \; h^{-1} \circ f' \circ h \subseteq f$

$\forall w \in X^*$, $\forall w' \in X^*$ tel que $h(w) \xrightarrow[C']{*} h(w')$, on a (lemme 5)

$$\forall i, j \in \{1,\dots,n\}, \; \text{proj}\,(h(w), \{(x_i, x_j), (x_j, x_i)\}) \xrightarrow[C']{*} \text{proj}\,(h(w'), \{(x_i, x_j), (x_j, x_i)\})$$

alors, par définition de h et de c', on en déduit que

$$\forall i, j \in \{1,\dots,n\}, \; \text{proj}(w, \{x_i, x_j\}) \overset{*}{\underset{C}{\to}} \text{proj}\,(w', \{x_i, x_j\})$$

et en réutilisant le lemme 5, on conclut que $w \overset{*}{\underset{C}{\to}} w'$.

Lemme 13.

Soient $\pi = \{Z_1, Z_2\}$ *une partition de* Z, c_π *(resp.* f_π*) la relation (resp. la fonction) associée à* π *et* C *une relation de commutation définie sur* Z *vérifiant la propriété suivante :*

$$\forall (x,y) \in C_\pi,\ xy \leftrightarrow yx \in C.$$

Soient

$$C_1 = \{xy \leftrightarrow yx / (x,y) \in C_\pi\} \cup \{xy \to yx \ / \ xy \to yx \in C \text{ et } (x,y) \in Z_1 \times Z_1\}$$

et

$$C_2 = \{xy \leftrightarrow yx / (x,y) \in C_\pi\} \cup \{xy \to yx / xy \to yx \in C \text{ et } (x,y) \in Z_2 \times Z_2\}$$

Si f, f_1 *et* f_2 *désignent respectivement les fonctions associées à* c, c_1 *et* c_2*, alors :*

$$f = f_\pi \circ f_1 \circ f_2 \circ f_\pi.$$

Preuve.

$\supseteq$ $\quad C, C_1, C_2 \subseteq C$

$$\Rightarrow \forall w \in Z^*,\ f_\pi \circ f_1 \circ f_2 \circ f_\pi(w) \subseteq f(w)$$

$\subseteq$ $\quad$ Soient $w \in Z$, $w' \in f(w)$. Alors il existe une dérivation :

$$w \overset{*}{\underset{c}{\to}} w'.$$

Soient $w_1 = \text{proj}(w, Z_1)$ et $w_2 = \text{proj}(w, Z_2)$. Alors on peut aussi trouver une dérivation telle que :

$$w \underset{c}{\leftrightarrow} w_1 w_2 \overset{*}{\underset{c}{\leftrightarrow}} w.$$

De même si $w_1' = \text{proj}\,(w', Z_1)$ et $w_2' = \text{proj}(w', Z_2)$, on peut écrire

$$w' \overset{*}{\underset{c}{\leftrightarrow}} w_1'\, w_2' \overset{*}{\underset{c}{\leftrightarrow}} w'.$$

Donc

$$w \overset{*}{\underset{c}{\longleftrightarrow}} w_1w_2 \overset{*}{\underset{c}{\rightarrow}} w'_1\, w'_2 \overset{*}{\underset{c}{\longleftrightarrow}} w'$$

$$\Rightarrow \text{(lemme 1) } \mathrm{Proj}(w_1w_2,\, Z_1) \overset{*}{\underset{c}{\rightarrow}} \mathrm{Proj}(w'_1w'_2,\, Z_1)$$

$$\text{et } \mathrm{Proj}(w_1w_2,\, Z_2) \overset{*}{\underset{c}{\rightarrow}} \mathrm{Proj}(w'_1w'_2,\, Z_2)$$

$$\Rightarrow w_1 \overset{*}{\underset{c}{\rightarrow}} w'_1 \text{ et } w_2 \overset{*}{\underset{c}{\rightarrow}} w'_2$$

$$\Rightarrow w_1 \overset{*}{\underset{c_1}{\longrightarrow}} w'_1 \text{ et } w_2 \overset{*}{\underset{c_2}{\longrightarrow}} w'_2$$

de plus

$$w \overset{*}{\underset{c}{\rightarrow}} w_1w_2 \Rightarrow w \overset{*}{\underset{c_\pi}{\longleftrightarrow}} w_1w_2,$$

de même

$$w'_1w'_2 \overset{*}{\underset{c_\pi}{\longleftrightarrow}} w'.$$

Donc

$$w \overset{*}{\underset{c_\pi}{\longleftrightarrow}} w_1w_2 \overset{*}{\underset{c_2}{\longrightarrow}} w_1w'_2 \overset{*}{\underset{c_1}{\longrightarrow}} w'_1w'_2 \overset{*}{\underset{c_\pi}{\longleftrightarrow}} w'$$

Donc

$$w' \in f_\pi \circ f_1 \circ f_2 \circ f_\pi(w).$$

□

<u>*Lemme 14.*</u>

Soit $\pi = \{\{a_1,b_1\}, \{a_2,b_2\},\ldots,\{a_n, b_n\}\}$ *une partition de l'alphabet* $X = \{a_1, b_1,\ldots,a_n, b_n\}$, c_π *la relation associée à* π.

Soient c *une relation de commutation définie sur* X *vérifiant*

$$\forall (x,y) \in C_\pi,\ xy \leftrightarrow yx \in C,$$

et f *la fonction associée.*
Alors il existe $g \in \{H, H^{-1}, CP\}^*$ *telle que* $f = g$.

<u>Preuve</u>.

La démonstration utilise un raisonnement par induction sur le nombre de relations non symétriques de C. Soit r ce nombre.

Si $r = 0$, C est une relation de commutation symétrique donc, $f \in CP$. Supposons le résultat vrai pour toute valeur de r plus petite que p. Alors si $r = p \geq 1$:

Supposons qu'on ajoute la relation $b_1a_1 \to a_1b_1$.
Soient $\pi_1 = \{X \setminus \{a_1,b_1\}, \{a_1, b_1\}\}$ la partition de X, c_{π_1} (resp. f_{π_1}) la relation (resp. la fonction) associée à π_1 : $\forall (x,y) \in \bar{c}_{\pi_1}$, $(x,y) \in c_\pi$

Soient

$$C_1 = \{xy \leftrightarrow yx,\ (x,y) \in c_{\pi_1}\} \cup \{xy \to yx \ /\ xy \to yx \in C\}$$

et

$$C_2 = \{xy \leftrightarrow yx,\ (x,y) \in c_{\pi_1}\} \cup \{b_1a_1 \to a_1b_1\}$$

On a en fait

$$C_1 = C \setminus \{b_1a_1 \to a_1b_1\}.$$

En appliquant le lemme 13, on a $f = f_{\pi_1} \circ f_1 \circ f_2 \circ f_{\pi_1}$ où f_1 et f_2 désignent respectivement les fonctions associées à C_1 et C_2.

D'après l'hypothèse de récurrence, il existe $g_1 \in \{H, H^{-1}, CP\}^*$ telle que

$$g_1 = f_1 \text{ sur } X^*.$$

D'après le lemme 11, il existe $g_o \in \{H, H^{-1}, CP\}^*$ telle que

$$g_o = f_o \text{ sur } \{a_1, b_1\}^*,$$

où f_o désigne la fonction associée à la relation $\{b_1a_1 \to a_1b_1\}$.
Donc, avec le lemme 10, on en déduit qu'il existe $g_2 \in \{H, H^{-1}, CP\}^*$ telle que

$$f_2 = f_{\pi_1} \circ g_2 \circ f_{\pi_1} \text{ sur } (X \setminus \{a_1,b_1\})^* \sqcup \{a_1,b_1\}^* = X^*$$

En regroupant ces résultats, on en déduit que :

$$f = f_{\pi_1} \circ f_1 \circ f_2 \circ f_{\pi_1} = f_{\pi_1} \circ g_1 \circ f_{\pi_1} \circ g_2 \circ f_{\pi_1} \in \{H, H^{-1}, CP\}^*$$

□

Des lemmes 12 et 14, on déduit immédiatement le résultat suivant :

Proposition 4.

Toute fonction de semi-commutation appartient à $\{H, H^{-1}, CP\}^*$, c'est à dire, peut être obtenue par composition d'homomorphismes, d'homomorphismes inverses et de fonctions de commutation partielle.

CHAPITRE III

COMMUTATIONS ET CÔNES RATIONNELS FIDÈLES

A) MULTI-RESET est fermé pour l'opération TS

B) Propriétés des familles de langages fermées par commutation partielle

C) Le plus petit cône rationnel fidèle clos par commutation partielle

Après avoir vu que toute fonction de commutation pouvait s'exprimer à l'aide des transductions rationnelles fidèles et des fonctions de commutations partitionnées, nous allons étudier maintenant les propriétés des cônes rationnels (fidèles) fermés par commutation (partitionné).

Nous avons vu le lien très étroit qui relie les fonctions de commutation partitionnée et l'opération TS.
Aussi, dans la première section, nous allons montrer qu'une famille de langages bien connue : la famille des langages multireset, notée MULTI-RESET, est fermée pour l'opération TS.

Ce résultat nous permettra de caractériser les cônes rationnels fidèles clos par intersection qui ont la propriété d'être fermé par commutation partitionnée : un cône rationnel fidèle fermé par intersection est clos par commutation partitionnée si et seulement si il contient le langage.

$$\text{COPY} = \{w\bar{w},\ w \in \{a,b\}^*\}$$

ou encore si et seulement si il est fermé par l'opération TS. Nous en déduirons un résultat déjà connu [5] : si un cône rationnel fidèle clos par intersection contient le langage COPY, il est fermé par homomorphisme linéairement effaçant, et nous donnerons des exemples de familles de langages déjà connues vérifiant ces propriétés.

Dans la troisième section, nous spécifierons le plus petit cône rationnel (fidèle) fermé par commutation partitionnée et nous donnerons une nouvelle définition des familles de langages MULTI-RESET, BNP et Q qui sont les plus petits cônes rationnels fidèles fermés par intersection et contenant respectivement les langages

$$\text{Sym} = \{w\bar{w}^R,\ w \in \{a,b\}^*\}$$

et $D_2'^*$ le langage de Dyck restreint sur deux paires de parenthèses.

III.A - MULTI-RESET est fermé par l'opération TS.

Rappelons que la famille des langages MULTI-RESET est le plus petit cône rationnel fidèle clos par intersection et contenant le langage COPY (Notation : MULTI-RESET = C_n^f (COPY)).

Le langage Twin Shuffle est l'ensemble TS = $\{w \sqcup\!\sqcup \bar{w},\ w \in \{a,b\}^*\}$ Plus généralement, on peut définir le langage TSn sur n lettres ($n \geq 1$) par

$$TSn = \{w \sqcup\!\sqcup \bar{w} / w \in \{a_1, a_2, \ldots, a_n\}\}.$$

On peut facilement vérifier que :

Lemme 1.

$$C^f(TS_2) = C^f(TS_n),\ \forall n \geq 2.$$

Preuve.

Posons $X = \{a_1, a_2, \ldots, a_n\}$ et $\bar{X} = \{\bar{a}_1, \bar{a}_2, \ldots, \bar{a}_n\}$.

$\subseteq$ $TS_2 = g(TS_n)$ où g est l'homomorphisme défini sur $X \cup \bar{X}$ par $g(a_i) = a$ et $g(\bar{a}_i) = b$, $\forall i \in \{1, \ldots, n\}$. Donc

$$TS_2 \in C^f(TS_n) \Rightarrow C^f(TS_2) \subseteq C^f(TS_n)$$

$\supseteq$ Soit h l'homomorphisme défini sur $X \cup \bar{X}$ par $h(a_i) = ba^i$ et $h(\bar{a}_i) = \bar{b}\bar{a}^i$. Il est clair que

$$TS_n = h^{-1}(TS_2) \in C^f(TS_2).$$

Donc

$$C^f(TS_n) \subseteq C^f(TS_2)$$

$\square$

Rappelons que pour tout langage $L \subseteq X^*$, on note $TS(L)$ le langage $\{w \sqcup\!\sqcup \bar{w},\ w \in L\}$.
Si h est l'homomorphisme défini sur $X \cup \bar{X}$ par

$$h(x) = x,\ \forall x \in X \text{ et } h(\bar{x}) = \varepsilon,\ \forall \bar{x} \in \bar{X},$$

et si TS_n est le langage Twin Shuffle défini sur n lettres, alors il est clair que $TS(L) = h^{-1}(L) \cap TS_n$. On en déduit le :

<u>*Lemme 2.*</u>

Un cône rationnel fidèle clos par intersection et contenant TS_2 est fermé pour l'opération TS.

En utilisant le fait que MULTI-RESET est fermé par homomorphisme linéairement effaçant [5], il est facile de montrer que TS_2 appartient à MULTI-RESET. Nous allons cependant en donner une démonstration qui n'utilise pas cette propriété, ce qui nous permettra de donner une nouvelle preuve de la fermeture de MULTI-RESET par homomorphisme linéairement effaçant, au chapitre suivant.

<u>*Lemme 3.*</u>

MULTI-RESET contient le langage TS_2.

<u>Preuve.</u>

Nous allons montrer que l'ensemble des mots de longueur paire de TS_2 est dans MULTI-RESET, ; on en déduit facilement le même résultat sur les mots de longueur impaire.

Soit $w \in \{a,b\}^*$. Posons $w = w_1w_2$ avec $|w_1| = |w_2|$.
Soit $z \in w \sqcup\!\sqcup \bar{w} = w_1w_2 \sqcup\!\sqcup \bar{w}_1\bar{w}_2$, alors $z = z'z''$ avec $|z'| = |z''| = |w|$.
Deux cas se présentent :

1°) z' contient tout w_1 : $z' = w_1 w_2' \sqcup\!\sqcup \bar{w}_1'$

$z'' = w_2'' \sqcup\!\sqcup \bar{w}_1''\bar{w}_2$

avec $w_1 = w_1' w_1''$ et $w_2 = w_2' w_2''$.

2°) z' contient tout $\bar{w}_1$: $z' = w_1' \sqcup\!\sqcup \bar{w}_1 \bar{w}_2'$

$z'' = w_1'' w_2 \sqcup\!\sqcup \bar{w}_2''$

avec $w_1 = w_1' w_1''$ et $w_2 = w_2' w_2''$.

Donc $z \in L_1 \cup L_2$ avec

$$L_1 = \{(w_1 w_2' \sqcup\!\sqcup \bar{w}_1')(w_2'' \sqcup\!\sqcup \bar{w}_1'' \bar{w}_2) / \ |w_1| = |w_2|,\ w_2 = w_2' w_2'',\ w_1 = w_1' w_1''\}$$

et

$$L_2 = \{(w_1' \sqcup\!\sqcup \bar{w}_1 \bar{w}_2')(w_1'' w_2 \sqcup\!\sqcup \bar{w}_2'') / \ |w_1| = |w_2|,\ w_1 = w_1' w_1'',\ w_2 = w_2' w_2''\}$$

Il faut donc montrer que ces langages sont dans MULTI-RESET, par l'intermédiaire des deux lemmes suivants :

<u>*Lemme 4*</u>.

Le langage $\{(xy \sqcup\!\sqcup \bar{x})x \ / \ x \in \{a,b\}^*,\ y \in \{a,b\}^*\}$ *est un langage de MULTI-RESET.*

<u>Preuve</u>.

Si x est un mot de $\{a,b\}^*$, designons par $x(1)$ et $x(2)$ les recopies de x sur $\{a_1, b_1\}$ et $\{a_2, b_2\}$; $\bar{x}$ est sa recopie sur $\{\bar{a}, \bar{b}\}$. Soit $Y = \{\alpha,\beta\}$. Il est clair que les langages

$$L_1 = \{xy\ x(1) \ / \ x \in \{a,b\}^*,\ y \in Y\}$$

et

$$L_2 = \{\bar{x}\ x(2) \ / \ x \in \{a,b\}^*\}$$

sont dans $C_n^f(\text{COPY})$.

Alors $L_3 = L_1 \sqcup\!\sqcup L_2 \cap (X \cup Y \cup \bar{X})\ \{a_1a_2,\ b_1b_2\}^* \in C_n^f(\text{COPY})$ et il est clair que $\forall w \in L_3,\ w \in (xy \sqcup\!\sqcup \bar{x})((x(1) \sqcup\!\sqcup x(2)) \cap \{a_1a_2,\ b_1b_2\}^*)$. Il ne reste qu'à effacer le $x(2)$ (par un effacement 1-limité sur L_3) et à démarquer.
Soit

$$h : (X \cup Y \cup \bar{X} \cup X_1 \cup X_2)^* \to (X \cup \bar{X})^*$$

$$h(x) = x \ \forall x \in X \cup \bar{X}$$

$$h(\alpha) = h(a_1) = a$$

$$h(\beta) = h(b_1) = b$$

$$h(a_2) = h(b_2) = \varepsilon$$

On a bien $h(L_3) = \{(xy \sqcup\!\sqcup \bar{x})x / x \in \{a,b\}^*,\ y \in \{a,b\}^*\}$.

□

Remarque.

On démontre de même que les langages

$$\{(w \sqcup\!\sqcup \bar{x}\bar{w})w \ /\ w \in \{a,b\}^* \text{ et } x \in \{a,b\}^*\},$$

$$\{(w \sqcup\!\sqcup \bar{w}\bar{x})w \ /\ w,\ x \in \{a,b\}^*\} \text{ et } \{(xw \sqcup\!\sqcup \bar{w})w \ /\ x,\ w \in \{a,b\}^*\}$$

sont dans $C_n^f(\text{COPY})$.

Lemme 5.

Le langage L *défini par*

$$L = \{(xy_1\, z_2 \sqcup\!\sqcup \bar{x})(t_3 \sqcup\!\sqcup \bar{y}_1\bar{z}_2\bar{t}_3) \ /\ x \in \{a,b\}^*,\ y_1 \in \{a_1,b_1\}^*,\ z_2 \in \{a_2,b_2\}^* \text{ et } t_3 \in \{a_3,b_3\}^*,\ |xy_1| = |z_2t_3|\}$$

est dans $C_n^f(\text{COPY})$.

Preuve.

D'après le lemme 4, les langages

$$L_1 = \{(xy_1 \; z_2 ш \bar{x}) \; \bar{\bar{x}} / x \in X^*, \; y_1 \in X_1^*, \; z_2 \in X_2^*\}$$

et

$$L_2 = \{\bar{\bar{t}}_3(t_3 ш \bar{y}_1' \; \bar{z}_2' \; \bar{t}_3)/t_3 \in X_3^*, \; y_1' \in X_1^*, \; z_2' \in X_2^*\}$$

où $X = \{a,b\}$, $X_1 = \{a_1, b_1\}$, $X_2 = \{a_2, b_2\}$, $X_3 = \{a_3, b_3\}$, sont dans $C_n^f(\mathrm{COPY})$

Pour retrouver L, nous allons faire le shuffle de L_1 et L_2 pour intercaler le mot $\bar{\bar{x}}$ dans le mot $\bar{y}_1' \; \bar{z}_2' \; \bar{t}_3$ et $\bar{\bar{t}}_3$ dans $xy_1 \; z_2$, en contrôlant que $y_1 = y_1'$, $z_2' =$ et $|xy_1| = |z_2t_3|$.

Soit $C_1 = \{w_1\bar{w}_1 \; / \; w_1 \in \{a_1, b_1\}^*\}$. Notons Y l'alphabet sur lequel est défini $L_1 \cup L_2$.
Soit $h_1 \; Y^* \to (X_1 \cup \bar{X}_1)^*$ tel que $h_1(x) = x$ si $x \in X_1 \cup \bar{X}_1$ et $h_1(x) = \varepsilon$ sinon.

$C_1 \in C_n^f(\mathrm{COPY}) \Rightarrow h_1^{-1}(C_1) \in C_n^f(\mathrm{COPY})$. Le langage $h_1^{-1}(C_1)$ assurera la condition $y_1 = y_1'$.

De même, on définit $h_2^{-1}(C_2)$ qui contrôlera l'égalité $z_2 = z_2'$.
Soit $E = \{w\bar{w}' \; / \; w \in X^*X_1^*, \; \bar{w}' \in \bar{X}_2^*\bar{X}_3^*, \; |w| = |w'|\}$.
Il est clair que E est dans $C_n^f(\mathrm{COPY})$.
Soit $g : Y^* \to (X \cup X_1 \cup \bar{X}_2 \cup \bar{X}_3)^*$ défini par

$$g(x) = x \text{ si } x \in X \cup X_1 \cup \bar{X}_2 \cup \bar{X}_3,$$

$g(x) = \varepsilon$ sinon. Alors $g^{-1}(E)$ qui est un langage de $C_n^f(\mathrm{COPY})$ vérifiera l'égalité de la longueur des mots xy_1 et z_2t_3.

$$\text{Soit } L' = L_1 ш L_2 \cap h_1^{-1}(C_1) \cap h_2^{-1}(C_2) \cap g^{-1}(E) \cap R_1R_2$$

avec

$$R_1 = [((X\bar{\bar{X}}_3 \cup X_1\bar{\bar{X}}_3)^* \; (X \cup X_1 \cup X_2)^*) ш \bar{X}^*]$$

$$R_2 = [X_3^* ш ((X_1\bar{\bar{X}} \cup \bar{X}_2\bar{\bar{X}} \cup X_3\bar{\bar{X}})^* \; (X_1 \cup X_2 \cup X_3)^*)]$$

Il est clair que $L' \in C_n^f(\mathrm{COPY})$.
Il reste à effacer les lettres qui ont des doubles barres. Ce sera un effacement 1-limité sur L' grâce à l'intersection avec R_1R_2.

Soit f cet homomorphisme défini sur Y^* par $f(x) = \varepsilon$ si $x \in \bar{\bar{X}}_3 \cup \bar{\bar{X}}$, $f(x) = x$ sinon.
On a $f(L') = L$. Donc $L \in C_n^f(\mathrm{COPY})$.

$\square$

Symétriquement, on a

$$L'' = \{(x \sqcup\!\sqcup \bar{x}\, \bar{y}_1\, \bar{z}_2)(y_1\, z_2\, t_3 \sqcup\!\sqcup \bar{t}_3) \;/\; |xy_1| = |z_2t_3|\} \in C_n^f(\mathrm{COPY})$$

Revenons à la démonstration du lemme 3. D'après le lemme 5, on a :

$$L = \{(xy_1\, z_2 \sqcup\!\sqcup \bar{x})(t_3 \sqcup\!\sqcup \bar{y}_1\, \bar{z}_2\, \bar{t}_3) \;/\; |xy_1| = |z_2t_3|\} \in C_n^f(\mathrm{COPY})$$

et

$$L'' = \{(x \sqcup\!\sqcup \bar{x}\bar{y}_1\bar{z}_2)(y_1z_2t_3 \sqcup\!\sqcup \bar{t}_3) \;/\; |xy_1| = |z_2t_3|\} \in C_n^f(\mathrm{COPY})$$

Soit h l'homomorphisme défini sur $X \cup \bar{X} \cup X_1 \cup \bar{X}_1 \cup X_2 \cup \bar{X}_2 \cup X_3 \cup \bar{X}_3$ par

$$h(a) = h(a_1) = h(a_2) = h(a_3) = a$$

$$h(b) = h(b_1) = h(b_2) = h(b_3) = b$$

$$h(\bar{a}) = h(\bar{a}_1) = h(\bar{a}_2) = h(\bar{a}_3) = \bar{a}$$

$$h(\bar{b}) = h(\bar{b}_1) = h(\bar{b}_2) = h(\bar{b}_3) = \bar{b}$$

Alors, il est clair que $h(L) \cup h(L'') = \{w \sqcup\!\sqcup \bar{w},\ w \in \{a,b\}^*,\ w$ de lg paire$\}$

$\square$

Du lemme 3, on déduit aisément le résultat suivant :

Proposition 1.

MULTI-RESET est clos par l'opération TS.

Preuve.

On a vu que $\forall L \in$ MULTI-RESET, $TS(L) = h^{-1}(L) \cap TS\, n$. Or $TS_2 \in$ Multireset $\Rightarrow$ TSn $\in$ Multireset.
Comme $h^{-1}(L) \in$ MULTI-RESET, on a donc TS(L) est dans MULTI-RESET.

$\square$

III.B - Propriétés des familles de langages fermées par commutation partitionnée.

Soit L une famille de langages.
On note $Pi(L) = \{f(L),\ L \subseteq X^*$, f fonction de commutation partitionnée associée à une partition de i éléments de X} et $P(L) = \bigcup_{i \geq 1} P_i(L)$.

On dira qu'une famille L est fermée par commutation partitionnée si et seulement si on a $L = P(L)$.

D'après le résultat du lemme 6 (chapître II), nous pouvons écrire que si $L = P_2(L)$ (si L est fermée par commutation 2-partitionnée), alors $P_i(L) \subseteq L\ \forall i \geq 2$. Comme $L \subseteq P(L)$, nous pouvons énoncer :

Lemme 6.

Soit L une famille de langages.
Si $L = P_2(L)$, alors L est fermée par commutation partitionnée.

Nous allons maintenant donner une propriété des cônes rationnels fidèles clos par commutation partitionnée :

Proposition 2.

Tout cône rationnel fidèle $\mathcal{L}$ clos par commutation partitionnée est fermé par homomorphisme linéairement effaçant et pour l'opération TS.

Preuve.

1°) Soit $L \in \mathcal{L}$ et h un homomorphisme linéairement effaçant sur L : $\exists k \in \mathbb{N}^+$ tel que si $x \in L$, $|x| > k$, on a $|x| \leq k \ |h(x)|$.
Si $L \subseteq X^*$, soient $X_1 = \{x \in X \ / \ h(x) = \varepsilon\}$, $X_2 = X \setminus X_1$, $\pi = \{X_1, X_2\}$ la partition de X et f la fonction de commutation associée.

Nous allons montrer que $h(L) = h(f(L) \setminus X^* X_1^k X^*)$.

* il est clair que $h(f(L) \setminus X^*X_1^kX^*) \subseteq h(L)$

* soit $x \in L$

Si $|x| < k$ ou si $x \notin X^*X_1^kX^*$, il n'y a pas de problème.
Si $x \in X^*X_1^kX^*$, alors choisissons $y \in f(x) \setminus X^*X_1^kX^*$ (y existe car $|x| \leq k|h(x)| = k|h(\mathrm{proj}(X, X_1)(x))|$). De plus, $h(x) = h(y)$.
Donc $h(L) \subseteq h(f(L) \setminus X^*X_1^kX^*)$.
Comme h est un homomorphisme k limité sur $X^*X_1^kX^*$, on a $h(L) \in \mathcal{L}$.

2°) Soit $L \subseteq X^*$ un langage de $\mathcal{L}$. Posons $\bar{X} = \{\bar{x} \ / \ x \in X\}$ et définissons le morphisme h sur X par $h(x) = x\bar{x}$, $\forall x \in X$. Si f est la fonction de commutation partitionnée associée à la partition $\pi = \{X, \bar{X}\}$ de $X \cup \bar{X}$, il est clair que $TS(L) = f(h(L)) \in \mathcal{L}$.

□

Nous allons maintenant montrer qu'il est équivalent de dire qu'un cône rationnel fidèle fermé par intersection est clos par commutation partitionnée, ou est clos pour l'opération TS ou qu'il contient le langage $COPY = \{w\bar{w}, w \in \{a,b\}^*\}$.

Proposition 3.

Soit L un cône rationnel fidèle fermé par intersection. Les trois propriétés suivantes sont équivalentes :

1°) COPY appartient à L

2°) L est fermé par commutation partitionnée

3°) L est fermé pour l'opération TS.

Preuve.

1°) => 2°).

Nous allons montrer que $\mathcal{L} = P_2(\mathcal{L})$ (cela suffit d'après le lemme 6). Soit $L \in \mathcal{L}$, $L \subseteq X^*$. Soit $\pi = \{X_1, X_2\}$ une partition de X et f la fonction de commutation associée.

Posons $\bar{X} = \bar{X}_1 \cup \bar{X}_2$, $Z = X \cup \bar{X}$ et définissons les projections suivantes :

* sur $X_1 \cup \bar{X}_1$: $h_1 : Z^* \to (X_1 \cup \bar{X}_1)^*$

$$x \to x \text{ si } x \in X_1 \cup \bar{X}_1$$

$$x \to \varepsilon \text{ sinon.}$$

* sur $X_2 \cup \bar{X}_2$: $h_2 : Z^* \to (X_2 \cup \bar{X}_2)^*$

* sur X : $h : Z^* \to X^*$

$$x \to x \text{ si } x \in X$$

$$x \to \varepsilon \text{ si } x \in \bar{X}.$$

Soit $L' = \{u \in (X\bar{X})^* \,/\, h(u) \in L,\ h_1(u) \in TS(X_1^*),\ h_2(u) \in TS(X_2^*)\}$,

Il est clair que $L' = (X\bar{X})^* \cap h^{-1}(L) \cap h_1^{-1}(TS(X_1^*)) \cap h_2^{-1}(TS(X_2^*))$ est dans $\mathcal{L}(TS(X_1) \in C_n^f(\text{COPY}) \subseteq \mathcal{L}$ - lemme 3).

Soit $u \in L'$. On peut écrire $u = x_1\bar{y}_1\ x_2\bar{y}_2\ \dots\ x_n\bar{y}_n$, avec $\forall i \in \{1, 2,\dots,n\}$, $x_i \in X$, $\bar{y}_i \in \bar{X}$.
Comme $h(u) \in L$, $x_1x_2\ \dots\ x_n \in L$ et comme $h_1(u) \in TS(X_1^*)$, on en déduit que $\overline{h_1(x_1x_2,\dots,x_n)} = h_1(\bar{y}_1\bar{y}_2\ \dots\ \bar{y}_n)$. De même $\overline{h_2(x_1x_2\ \dots\ x_n)} = h_2(\bar{y}_1\ \bar{y}_2\ \dots\ \bar{y}_n)$.
Donc $y_1y_2\ \dots\ y_n$ est un mot de $h_1(x_1x_2\ \dots\ x_n) \sqcup\!\sqcup\, h_2(x_1x_2\ \dots\ x_n) = f(x_1x_2\dots x_n)$.
Dans le langage L', il reste à effacer les barres et les lettres non barrées.

Soit g l'homomorphisme défini sur $(X \cup \bar{X})^*$ par $g(\bar{x}) = x$ si $\bar{x} \in \bar{X}$, $g(x) = \varepsilon$ si $x \in X$. Alors $g(L') \subseteq \{h_1(w) \sqcup\!\sqcup\, h_2(w),\ w \in L\} = f(L)$.
Comme g est 1-limité sur L', $g(L')$ est dans $\mathcal{L}$.
Réciproquement, montrons que $f(L) \subseteq g(L')$:

$$\forall u \in f(L),\ \exists w \in L \text{ tel que } u \in h_1(w) \sqcup\!\sqcup\, h_2(w).$$

Soit $v = w \sqcup\!\sqcup\, \bar{u} \cap (X\bar{X})^*$.
Alors, il est clair que $u = g(v)$ et $v \in L'$ car $v \in (X\bar{X})^*$, $h(v) = w \in L$, $h_1(v) \in TS(X_1^*)$ et $h_2(v) \in TS(X_2^*)$.
Donc $f(L) \subseteq g(L')$. On en déduit que $f(L) = g(L') \in \mathcal{L}$ et ainsi $P_2(\mathcal{L}) = \mathcal{L}$.

2) $\Rightarrow$ 3) voir proposition 2.

3) $\Rightarrow$ 1) $COPY = TS(\{a,b\}^*) \cap \{a,b\}^*\{\bar{a}, \bar{b}\}^*$.

□

De cette proposition, on déduit deux corollaires.

Corollaire 1 [5]

Soit $\mathcal{L}$ une famille de langages.
Si COPY appartient à $C_n^f(\mathcal{L})$, alors $C_n^f(\mathcal{L})$ est fermé par homomorphisme linéairement effaçant.

Corollaire 2.

Les familles BNP = C_n^f(SYM) ([6]) et Q = C_n^f(Alg) ([4]) sont fermées par commutation partitionnée.

Preuve.

Il suffit de montrer que COPY est dans $C_n^f(\mathrm{SYM})$ (donc dans $C_n^f(\mathrm{Alg})$).
Soient $S_o = \mathrm{Sym} = \{w\bar{w}^R,\ w \in \{a,b\}^*\}$, $S_1 = \{w^R\bar{w}/w \in \{a_1,b_1\}^*\}$ et
$\mathrm{Sym}' = \{ww^R,\ w \in \{a,b\}^*\}$.
Notons $X = \{a,b\}$, et $X_1 = \{a_1, b_1\}$.
Soit $L = S_o \sqcup\!\sqcup\, S_1 \cap [(XX_1)^* \cap h^{-1}(\mathrm{Sym}')].(\bar{X}\bar{X}_1)^*$, où h est l'homomorphisme défini sur $(X \cup X_1)^*$ par $h(x) = x$ si $x \in X$, $h(x_1) = x$ si $x \in X_1$.

Si $u \in L$, $u \in w\bar{w}^R \sqcup\!\sqcup\, w_1'^R\bar{w}_1'$ avec $w \in \{a,b\}^*$, $w_1' \in \{a_1, b_1\}^*$ et $u = v\bar{v}$, $v \in (XX_1)^* \cap h^{-1}(\mathrm{Sym}')$ et $\bar{v} \in (\bar{X}\bar{X}_1)^*$.

Posons $w = x_1x_2 \ldots x_n$, $x_i \in X$, $\forall i$ et $w_1'^R = y_1'y_2' \ldots y_p'$, $y_i' \in X_1$, $\forall i$.
Alors $v \in (XX_1)^* \Rightarrow n = p$.
Donc $v = x_1y_1'x_2y_2' \ldots x_ny_n'$, et $v \in h^{-1}(\mathrm{Sym}')$

$$\Rightarrow h(v) \in \mathrm{Sym}$$

$$\Rightarrow x_1y_1x_2y_2 \ldots x_ny_n \in \mathrm{Sym}, \text{ avec } x_i, y_i \in \{a,b\}\ \forall i$$

$$\Rightarrow x_1 = y_n,\ x_2 = y_{n-1}, \ldots, x_n = y_1$$

$$\Rightarrow x_1x_2 \ldots x_n = (y_1y_2 \ldots y_n)^R$$

Donc $w = h(w_1')$.

$$\Rightarrow u \in (w\bar{w}^R \sqcup\!\sqcup\, w_1^R\bar{w}_1)^* \cap (XX_1)^* (\bar{X}\bar{X}_1)^* \text{ ou } h(w_1) = w.$$

Soit $g : (X \cup X_1 \cup \bar{X} \cup \bar{X}_1)^* \to X^*$ tel que $g(x) = x\ \forall x \in X$, $g(y) = \varepsilon\ \forall y \in \bar{X} \cup X_1$, $g(\bar{a}_1) = a$ et $g(\bar{b}_1) = b$.

g est un homomorphisme 1-limité sur L et il est clair que $g(L) = \mathrm{COPY}$.
Donc $\mathrm{COPY} \in C_n^f(\mathrm{Sym})$.

□

III.C - LE PLUS PETIT CONE RATIONNEL FERMEE PAR COMMUTATION.

Après avoir établi des critères permettant de reconnaître une famille de langages fermée par commutation partitionnée, il semble naturel de s'intéresser maintenant à la plus petite de ces familles, plus précisèment, au plus petit cône rationnel fidèle fermé pour les opérations de commutation partitionnée (donc de commutation quelconque). Dans un premier temps nous allons montrer qu'il est égal à MULTI-RESET, puis nous en donnerons une spécification plus précise.

Notation.

Pour une famille de langages ,

$$\mathcal{L}.\mathcal{L} = \{L_1L_2 \ / \ L_1, L_2 \in \mathcal{L}\}$$

$C_p(\mathcal{L})$ (resp. $C_p^f(L)$) représente le plus petit cône rationnel (resp. fidèle) fermé par commutation, contenant $\mathcal{L}$.

Nous allons donner une condition nécessaire et suffisante pour qu'un cône rationnel fidèle fermé par commutation, soit clos par intersection.

Proposition 4.

Soit $\mathcal{L}$ une famille de langages fermée par recopie. Les deux propriétés suivantes sont équivalentes :

1°) $C_p^f(\mathcal{L})$ est fermé par intersection
2°) $\mathcal{L}.\mathcal{L} \subseteq C_p^f(\mathcal{L})$.

Preuve.

1) $\Rightarrow$ 2) Si $C_p^f(\mathcal{L})$ est fermé par intersection, alors $C_p^f(\mathcal{L})$ est fermé par Shuffle ([9]).

Alors, $\forall L_1, L_2 \in C_p^f(\mathcal{L})$, $L_1 \subseteq X_1^*$, $L_2 \subseteq X_2^*$, $L_1 \sqcup\!\sqcup \overline{L_2} \in C_p^f(\mathcal{L})$, ce qui entraîne que

$$L_1L_2 = h(L_1 \sqcup\!\sqcup \overline{L_2} \cap X_1^* \bar{X}_2^*)$$

où h est l'homomorphisme qui enlève les barres.

2) $\Rightarrow$ 1) Il suffit de montrer que $C_p^f(\mathcal{L})$ est fermé par produit (si $L_1 \subseteq X_1^*$, $L_2 \subseteq X_2^*$ sont deux langages de $C_p^f(\mathcal{L})$, avec $X_1 \cap X_2 = \emptyset$, $\pi = \{X_1, X_2\}$ est une partition de $X_1 \cup X_2$ et f la fonction associée, alors $f(L_1 L_2) = L_1 \sqcup\!\sqcup L_2 \in C_p^f(\mathcal{L})$, donc $C_p^f(\mathcal{L})$ est fermé par shuffle et intersection).

* Soient $L_1 \subseteq X_1^*$ et $L_2 \subseteq X_2^*$ deux langages de $P(\mathcal{L})$, avec $X_1 \cap X_2 = \emptyset$. On peut écrire $L_1 = f_1(L_1')$, $L_1' \in \mathcal{L}$, f_1 fonction de commutation associée à une partition π_1 de X_1 à n éléments, de même $L_2 = f_2(L_2')$, $L_2' \in \mathcal{L}$, f_2 fonction de commutation associée à une partition π_2 de X_2 à n éléments.

Si $\pi_1 = \{X_1^1, X_1^2, \dots, X_1^n\}$ et $\pi_2 = \{X_2^1, X_2^2, \dots, X_2^n\}$, posons $Y^k = X_1^k \cup X_2^k$, $Y = X_1 \cup X_2$, $\pi = \{Y^1, Y^2, \dots, Y^n\}$ et f la fonction de commutation associée à π.

Alors $L_1L_2 = f(L_1'L_2') \cap X_1^*X_2^*$;
Comme $L_1'L_2' \in C_p^f(\mathcal{L})$, on a donc $L_1L_2 \in C_p^f(\mathcal{L})$.

** Si L_1 et L_2 sont dans $C^f(P(\mathcal{L}))$, avec $L_1 \subseteq X_1^*$, $L_2 \subseteq X_2^*$ et $X_1 \cap X_2 = \emptyset$, on peut écrire :

$$L_1 = g_1(h_1^{-1}(L_1') \cap R_1),\ L_1' \in P(\mathcal{L}),\ L_1' \subseteq Y_1^*,\ R_1 \in \text{Rat}$$

et

$$L_2 = g_2(h_2^{-1}(L_2') \cap R_2),\ L_2' \in P(\mathcal{L}),\ L_2' \subseteq Y_2^*,\ R_2 \in \text{Rat}.$$

On peut supposer que $Y_1 \cap Y_2 = \emptyset$, et que si $R_1 \subseteq Z_1^*$, $R_2 \subseteq Z_2^*$, on a $Z_1 \cap Z_2 = \emptyset$.

Soient h l'homomorphisme défini sur $Z_1 \cup Z_2$ par $h/Z_1 = h_1$ et $h/Z_2 = h_2$, et g défini sur $Z_1 \cup Z_2$ par $g/Z_1 = g_1$ et $g/Z_2 = g_2$; soit $R = R_1R_2$.

Alors $L_1L_2 = g(h^{-1}(L_1'L_2') \cup R_1R_2)$. Comme L_1' et L_2' sont dans $P(\mathcal{L})$, $L_1'L_2' \in C_p^f(\mathcal{L})$ donc L_1L_2 est dans $C_p^f(\mathcal{L})$.

En raisonnant par induction, supposons que $\mathcal{L}_1 = C^f(P(\dots(C^f(P(\mathcal{L})\dots)$ est fermé par produit, alors, $\forall L_1, L_2 \in P(\mathcal{L}_1)$, $L_1.L_2 \in C^f(P(\mathcal{L}_1)) \subseteq C_p^f(\mathcal{L})$: la démonstration est la même que (*) et

$$\forall L_1, L_2 \in C^f(P(\mathcal{L}_1)),\ L_1.L_2 \in C^f(P(\mathcal{L}_1)) \subseteq C_p^f(\mathcal{L}) :$$

la démonstration est la même que (**), on en déduit donc que $C_p^f(\mathcal{L})$ est fermé par produit, et donc par intersection.

$\square$

On en déduit alors facilement la

Proposition 5.

La famille des langages MULTI-RESET est le plus petit cône rationnel fidèle fermé par commutation (partitionnée), et la famille des langages récursivement énumérables est le plus petit cône rationnel fermé par commutation.

Preuve.

1°) RAT $\subseteq C_n^f(\text{COPY})$ qui est fermé par commutation partitionnée. (Proposition 3). Donc $C_p^f(\text{RAT}) \subseteq C_n^f(\text{COPY})$.

D'autre part, toujours d'après la proposition 3, COPY $\in C_p^f(\text{RAT})$ donc $C^f(\text{COPY}) \subseteq C_p^f(\text{RAT})$ qui est fermé par intersection (proposition 4), ce qui implique que $C_n^f(\text{COPY}) \subseteq C_p^f(\text{RAT})$.
Donc $C_p^f(\text{RAT})$ = MULTI-RESET est le plus petit cône rationnel fidèle clos par commutation (partitionnée).

2°) La démonstration est analogue pour montrer que

$$\text{RE} = C_n(\text{COPY}) = C_p(\text{RAT}).$$

□

Nous allons maintenant spécifier la famille $C_p^f(\text{RAT})$ et plus précisément, montrer que tout langage de cette famille est de la forme $h(L \cap K)$ où h est homomorphisme strictement alphabétique, L un langage de P(RAT) et K un langage rationnel. Nous obtiendrons en même temps une nouvelle caractérisation des familles BNP = $C_n^f(\text{SYM})$ et Q = $C_n^f(\text{Alg})$.
Pour cela, nous avons besoin de quelques résultats intermédiaires.

Définition.

Une famille de langages L est fermée par shuffle disjoint si pour tous langages L_1 et L_2 de L définis sur des alphabets disjoints, $L_1 \sqcup\!\sqcup L_2$ est dans L.

Lemme 7.

Soit L une famille de langages fermée par shuffle disjoint et homomorphisme alphabétique inverse. Alors, $C^f(L)$ est fermé par intersection.

Preuve.

Soient $L_1, L_2 \in C^f(L)$.

Alors

$$L_1 = h_1(M_1 \cap R_1) \quad , \quad M_1 \in L, \quad R_1 \in \text{Rat}$$

et

$$L_2 = h_2(M_2 \cap R_2) \quad , \quad M_2 \in L, \quad R_2 \in \text{RAT},$$

h_1 (resp. h_2) étant un homomorphisme alphabétique ε-limité sur R_1 (resp. R_2).

Supposons L_1 et L_2 définis sur l'alphabet X.
Soient $\bar{L}_2 = \bar{h}_2(\bar{M}_2 \cap \bar{R}_2)$ (on barre les lettres), et h l'homomorphisme défini sur $(X \cup \bar{X})$ par $h(x) = h_1(x)$ et $h(\bar{x}) = h_2(x)$.

Alors $L_1 = h(M_1 \cap R_1)$ et $\bar{L}_2 = h(\bar{M}_2 \cap \bar{R}_2)$. Comme h est alphabétique $L_1 ⧢ \bar{L}_2 = h(M_1 ⧢ \bar{M}_2 \cap R_1 ⧢ \bar{R}_2)$. Et comme $M_1 ⧢ \bar{M}_2 \in L$, on a donc $L_1 ⧢ \bar{L}_2 \in C^f(L)$ et $L_1 ⧢ L_2 \in C^f(L)$. On en déduit que $C^f(L)$ est fermé par shuffle et donc par intersection.

□

Lemme 8.

Soit $k \in \mathbb{N}^+$ et $g : Y^* \to X^*$ un homomorphisme alphabétique. Alors, pour toute fonction de commutation k-partitionnée f, définie sur X, il existe $f' \in P_k$ tel que $g^{-1} \circ f = f' \circ g^{-1}$.

Preuve.

Soit $\pi = \{X_1, X_2,\dots,X_k\}$ la partition de X correspondant à la fonction f.

Pour $i \in \{1,\dots,k\}$, posons $h_i = \text{proj}(X, X_i)$ et $Y_i = \{y \in Y, g(y) \in X_i\}$. Soit f' la fonction de commutation partitionnée associée à la partition $\pi' = \{Y_1 \cup Z, Y_2,\dots,Y_k\}$ où $Z = \{y \in Y, g(y) = \varepsilon\}$. Soit $h_1' = \text{proj}(Y, Y_1 \cup Z)$

et $\forall i \in \{2,\dots,k\}$, $h'_i = \text{proj}(Y, Y_i)$.

Alors, $\forall w \in X^*$, $g^{-1} \circ f(w) = g^{-1}(h_1(w) ⧢ \dots ⧢ h_k(w))$ et puisque g est alphabétique, $g^{-1} \circ f(w) = g^{-1} \circ h_1(w) ⧢ \dots ⧢ g^{-1} \circ h_k(w)$.

Pour $i \in \{1,\dots,k\}$, $g^{-1} \circ h_i(w) = h'_i \circ g^{-1}(w) ⧢ Z^*$, donc $g^{-1} \circ f(w) = (h'_1 \circ g^{-1}(w) ⧢ Z^*) ⧢ \dots ⧢ (h'_k \circ g^{-1}(w) ⧢ Z^*)$.
Mais $Z^* ⧢ Z^* = Z^*$ et $h'_1 \circ g^{-1}(w) = h'_1 \circ g^{-1}(w) ⧢ Z^*$.
Nous pouvons alors écrire $g^{-1} \circ f(w) = h'_1 \circ g^{-1}(w) ⧢ \dots ⧢ h'_k \circ g^{-1}(w) = f' \circ g^{-1}(w)$.

$\square$

Une conséquence immédiate de ce lemme est :

Corollaire 3.

Soit $k \in \mathbb{N}^+$ *et* L *une famille de langages fermée par homomorphisme alphabétique inverse. Alors* $P_k(L)$ *et* $P(L)$ *sont fermés par homomorphisme alphabétique inverse.*

Lemme 9.

Soit L *une famille de langages fermée par produit. Alors* $P(L)$ *est fermée par shuffle disjoint.*

Preuve.

Soient $L_1 \subseteq X_1^*$, $L_2 \subseteq X_2^*$ deux langages de $P(L)$ avec $X_1 \cap X_2 = \emptyset$.
Alors, pour $i \in \{1,2\}$, $L_i = f_i(M_i)$ où $M_i \in L$ et f_i est une fonction de commutation partitionnée associée à une partition π_i de X_i. Il est alors facile de vérifier que $L_1 ⧢ L_2 = f(M_1M_2)$ où f est la fonction de commutation partitionnée associée à la partition $\pi = \pi_1 \cup \pi_2$ de $X_1 \cup X_2$.
Donc $L_1 ⧢ L_2 \in P(L)$.

$\square$

Proposition 6.

Soit L un cône rationnel fermé par produit. Il vérifie alors :

$$C^f(P(L)) = C^f_p(L) = C^f_n(P(L)).$$

Preuve.

On déduit des lemmes 7, 9 et du corollaire 3 que $C^f(P(L))$ est fermé par intersection et $C^f_n(P(L)) = C^f(P(L)) \subseteq C^f_p(L)$. Or, il est clair que COPY $\in C^f_n(P(L))$. Alors, d'après la proposition 3, $C^f_n(P(L))$ est fermé par commutation partitionnée. Donc, $C^f_p(L) \subseteq C^f_n(P(L))$ et nous obtenons le résultat désiré.

$\square$

Pour toutes familles de langages L_1 et L_2, nous noterons $L_1 \wedge L_2$ la famille $\{L_1 \cap L_2 \,/\, L_1 \in L_1,\ L_2 \in L_2\}$ et $H^{\varepsilon}(L_1)$ la famille $\{h(L_1)/L_1 \in L_1$, h est un homomorphisme ε-limité sur $L_1\}$. En utilisant les résultats précédents, on obtient une première caractérisation des familles MULTI-RESET, BNP et Q.

Corollaire 4.

$$\text{MULTI-RESET} = C^f_p(\text{RAT}) = H^{\varepsilon}(P(\text{RAT}) \wedge \text{RAT}) = C^f(P(\text{Rat}))$$

$$\text{BNP} = C^f_n(\text{SYM}) = H^{\varepsilon}(P(\text{Lin}) \wedge \text{RAT}) = C^f(P(\text{Lin}))$$

$$Q = C^f_n(\text{Alg}) = H^{\varepsilon}(P(\text{Alg}) \wedge \text{RAT}) = C^f(P(\text{ALg})).$$

Preuve.

Pour montrer les égalités concernant MULTI-RESET et Q, il suffit de remarquer que RAT et Alg sont des cônes rationnels fermés par produit et d'appliquer les résultats de la proposition 6, du corollaire 2.

La famille Lin n'est pas fermée par produit, mais en utilisant une autre définition des langages linéaires, on montre que P(Lin) est fermé par shuffle disjoint et on en déduira de la même manière le résultat.

On peut toujours écrire un langage linéaire $L \subseteq X^*$ sous la forme :

$$L = \{uv^R \ / \ (u,v) \in R, \ R \text{ rationnel de } X^* \times X^* \ .$$

Si

$$L_1 = \{uv^R/(u,v) \in R \subseteq X^* \times X^*\}$$

et

$$L_2 = \{xy^R/(x,y) \in R' \subseteq X'^* \times X'^*\},$$

avec $X \cap X' = \emptyset$, alors

$$L_3 = \{wz^R/(w,z) \in RR'\} \in \text{Lin}.$$

Soient $\pi = \{X_1,\ldots,X_k\}$ une partition de X et $\pi' = \{X'_1,\ldots,X'_r\}$ une partition de X'. Notons f et f' les fonctions associées.
Alors $f(L_1) ⧢ f'(L_2) = f''(L_3)$ où f" est la fonction correspondant à la partition $\pi'' = \{X_1, X_2,\ldots,X_k, X'_1,\ldots,X'_r\}$ de $X \cup X'$. Donc P(Lin) est fermée par shuffle disjoint.

$\boxdot$

Il nous faut maintenant transformer l'homomorphisme ε-limité en homomorphisme strictement alphabétique. Pour cela, nous allons introduire deux types de transduction rationnelle : les transductions rationnelles croissantes et les fonctions sous séquentielles.

Définition.

Une transduction rationnelle $\tau : X^* \to Y^*$ *est croissante si* $\forall u \in X^*, \forall v \in \tau(u), |v| \geq |u|$.

La caractérisation de ces transductions est due à J. Leguy.

Proposition 7 ([15]).

Soit τ *une transduction rationnelle.* τ *est croissante si et seulement si il existe un alphabet* Z, *un langage rationnel* $R \subseteq Z^*$, *un morphisme alphabétique* h *et un morphisme strictement alphabétique* g *tels que* $\tau = g \circ (\cap R) \circ h^{-1}$

Définition (voir [1]).

Un transducteur sous-séquentiel est un 5-uplet $S = \langle X, Y, Q, q_o, \rho \rangle$ *où :*

- X *est l'alphabet d'entrée*
- Y *est l'alphabet de sortie*
- Q *est un ensemble fini d'états*
- $q_o \in Q$ *est l'état initial*
- ρ *est une fonction partielle :* $Q \to Y^*$

sur lequel sont définies 2 fonctions partielles ayant le même domaine de définition :

$\bullet : Q \times X \to Q$ *appelée fonction de transition*

$* : Q \times X \to Y^*$ *appelée fonction de sortie.*

Ces fonctions sont étendues à $Q \times X^*$ *en posant, pour* $u \in X^*$, $x \in X$,

$$q\bullet\varepsilon = q,\ q\bullet(ux) = (q\bullet u)\bullet x$$

$$q*\varepsilon = \varepsilon,\ q*(ux) = (q*u)((q\bullet u)*x).$$

Alors, la fonction sous-séquentielle θ *associée à* S *est définie par :*

$$\forall u \in X^*,\ \theta(u) = (q_o*u)\rho(q_o\bullet u)$$

et θ *est fidèle si il existe* $k \in \mathbb{N}^+$ *tel que* $\forall q \in Q$, $\forall u \in X^*$ $q*u = \varepsilon$ *implique* $|u| \le k$.

Lemme 10.

Soient $k,t \in \mathbb{N}^+$, *et* f *une fonction de commutation* k*-partitionnée définie sur* X. *Alors il existe une fonction sous-séquentielle fidèle* θ *et une fonction* f' *de commutation* k*-partitionnée tels que* $f = \theta^{-1} \circ f' \circ \theta$ *et* $\forall u \in X^*$, $|u| \geq n \Rightarrow |u| \geq t|\theta(u)|$.

Preuve.

Soit $\pi = \{X_1,\ldots,X_k\}$ la partition de X correspondant à f, et pour $i \in \{1,\ldots,k\}$, posons $h_i = \mathrm{proj}(X, X_i)$, et $\overline{h_i} = \mathrm{proj}(X, X \backslash X_i)$.

Soit S le transducteur sous-séquentiel défini de la manière suivante :

$$S = \langle X, Y, Q, q_o, \rho\rangle \text{ avec}$$

avec

* $Y = \bigcup_{i=1}^{k} Y_i$, et $\forall i \in \{1,\ldots,k\}$, $Y_i = \{(u)/u \in X_i^*,\ 1 \leq |u| \leq t+1\}$

* $Q = \{(u)\ /\ u \in X^*$, et $\forall i \in \{1,\ldots,k\}$, $|h_i(u)| \leq t\}$

* $q_o = (\varepsilon)$

* ρ est définie sur Q par $\rho((u)) = u_1u_2\ldots u_k$ où $\forall i \in \{1,\ldots,k\}$, $u_i = \varepsilon$ si $h_i(u) = \varepsilon$, $u_i = (h_i(u))$ sinon.

Les fonctions de transition et de sortie sont définis par :

$$\forall (u) \in Q,\ \forall i \in \{1,\ldots,k\},\ \forall x \in X_i,$$

$$(u)\cdot x = \begin{cases} (ux) \text{ si } |h_i(u)| < t, \\ (\overline{h_i}(u)) \text{ sinon} \end{cases}$$

$$(u) * x = \begin{cases} \varepsilon \text{ si } |h_i(u)| < t \\ (h_i(u)x) \text{ sinon.} \end{cases}$$

Il est clair que la fonction sous-séquentielle associée à S est fidèle. Notons cette fonction θ.

Posons maintenant $\pi' = \{Y_1,\ldots,Y_k\}$ et f' la fonction de commutation partitionnée définie sur Y associée à π'.
Pour $i \in \{1,\ldots,k\}$, posons $h_i' = \mathrm{proj}(Y, Y_i)$. Il est facile de vérifier que $\theta^{-1} \circ \theta$ est la fonction identité sur chaque X_i^* et que $\forall i \in \{1,\ldots,k\}$, $h_i' \circ \theta = \theta \circ h_i'$ et $h_i = \theta^{-1} \circ h_i' \circ \theta$.

Soit $w \in X^*$. Alors,

$$\forall i \in \{1,\ldots,k\},\ h_i' \circ \theta \circ f(w) = \theta \circ h_i \circ f(w) = \theta \circ h_i(w) = h_i'\circ\theta(w),$$

donc,

$$\theta \circ f(w) \subseteq h_1' \circ \theta(w) ш \ldots ш h_k' \circ \theta(w) = f' \circ \theta(w)$$

et

$$f(w) \subseteq \theta^{-1} \circ f' \circ \theta(w).$$

Réciproquement, si $w' \in \theta^{-1} \circ f' \circ \theta(w)$, alors $\theta(w') \in f' \circ \theta(w)$ et $\forall i \in \{1,\ldots,k\}$, $h_i' \circ \theta(w') = h_i' \circ \theta(w)$, ce qui entraîne que :

$$h_i(w') = \theta^{-1} \circ h_i' \circ \theta(w') = \theta^{-1} \circ h_i' \circ \theta(w) = h_i(w).$$

Donc $w' \in h_1(w) ш \ldots ш h_k(w) = f(w)$. On en conclut donc que $f = \theta^{-1} \circ f' \circ \theta$.

Posons maintenant $n = k(t+1)^2$ et considérons $w \in X^*$ avec $|w| \geq n$.
Alors $\theta(w) = (u_1) \ldots (u_p)(v_1) \ldots (v_q)$ avec $0 \leq q \leq k$,
$|u_1| = |u_2| = \ldots = |u_p| = t+1$, $|w| = |u_1| + \ldots + |u_p| + |v_1| + \ldots + |v_q|$
et $\forall i \in \{1,\ldots,q\}$, $|v_i| \leq t$.

Alors, $|\theta(w)| = p+q \leq p+k$ et $p(t+1) \leq |w| \leq p(t+1) + kt$.
Puisque $n = k(t+1)^2 \leq |w| \leq p(t+1) + kt$, nous avons alors $p \geq kt$ et
$|w| \geq p(t+1) \geq pt + kt = t(p+k) \geq t|\theta(w)|$.

$\square$

Lemme 11.

Soit $\mathcal{L}$ un cône rationnel fidèle et $L \subseteq X^*$ *un langage de* $P_k(\mathcal{L})$. *Alors pour tout homomorphisme* $h : Z^* \to X^*$, *il existe un langage* $L' \in P_k(\mathcal{L})$ *et une transduction rationnelle croissante* τ *tels que* $\tau(L') = h^{-1}(L)$.

Preuve.

D'après le corollaire 3, $P_k(\mathcal{L})$ est fermée par homomorphisme alphabétique inverse. Aussi nous pouvons supposer que h est non effaçant.
Posons $L = f(M)$ où $M \in \mathcal{L}$ et f est une fonction de commutation k-partitionnée.
Soit $t = \sup\{|h(y)| \ / \ y \in Z\}$, et θ, f', et n vérifiant le lemme précédent.

Considérons alors le langage $L' := f' \circ \theta(M) \cup \{\varepsilon\} \in P_k(\mathcal{L})$, $A_n = X^n X^*$ et la transduction rationnelle $\tau_1 = h^{-1} \circ \theta^{-1} \circ (\cap\, \theta(A_n))$.
Il est clair que $\theta^{-1} \circ \theta(A_n) = A_1$ et τ_1 est croissante.
En effet, si $v \in \tau_1(u)$, il existe $u' \in \theta^{-1}(u \cap \theta(A_n))$ tel que $v \in h^{-1}(u')$; donc $u \in \theta^{-1} \circ \theta(A_n) = A_n$ et $h(v) = u'$, $\theta(u') = u$. Avec le lemme 10 et le choix de t, nous avons alors $t|u| = t|\theta(u')| \leq |u'| \leq t|v|$ et donc $|u| \leq |v|$.
Nous avons aussi

$$\tau_1(L') = h^{-1}(\theta^{-1}(L' \cap \theta(A_n))) = h^{-1}(\theta^{-1}(L') \cap A_n).$$

De plus, puisque $\varepsilon \in L'$ et $h^{-1}(L \backslash A_n)$ est un ensemble fini, il est facile de trouver une transduction rationnelle croissante τ_2 telle que $\tau_2(L') = h^{-1}(L \backslash A_n)$.
Donc, nous pouvons construire une transduction rationnelle croissante τ telle que

$$\tau(L') = \tau_1(L') \cup \tau_2(L') = h^{-1}(L \cap A_n) \cup h^{-1}(L \backslash A_n) = h^{-1}(L).$$

$\square$

Nous en arrivons maintenant au résultat cherché :

Proposition 8.

Soit $\mathcal{L}$ un cône rationnel fidèle. Alors, pour tout $k \in \mathbb{N}^+$, $C^f(P_k(\mathcal{L})) = \mathcal{H}^{s\alpha}(P_k(\mathcal{L}) \wedge \mathrm{Rat})$ *et* $C^f(P(\mathcal{L})) = \mathcal{H}^{s\alpha}(P(\mathcal{L}) \wedge \mathrm{Rat})$.

<u>Preuve</u>.

L'inclusion $\mathcal{H}^{s\alpha}(P_k(L) \wedge \mathrm{Rat}) \subseteq C^f(P_k(L))$ est évidente.
Pour l'inclusion inverse, posons $L \in C^f(P_k(L))$. Alors, $L = g(h^{-1}(L') \cap R)$, où $R \in \mathrm{Rat}$, $L' \in P_k(L)$ et g est un homomorphisme strictement alphabétique. Mais, d'après le lemme précédent, $h^{-1}(L') = \tau(L'')$ où $L'' \in P_k(L)$ et τ est une transduction rationnelle croissante. Alors, $L = \tau'(L'')$ avec $\tau' = g \circ \cap R \circ \tau$ qui est une transduction rationnelle croissante.
Nous déduisons alors de la proposition 7 qu'il existe un langage rationnel K, un homomorphisme strictement alphabétique g' et un homomorphisme alphabétique h' tels que $\tau' = g' \circ (\cap K) \circ h'^{-1}$ et $L = g'(h'^{-1}(L'') \cap K)$. Puisque $P_k(L)$ est fermée par homomorphisme alphabétique inverse, $h^{-1}(L'') \in P_k(L)$ et donc $L \in \mathcal{H}^{s\alpha}(P_k(L) \wedge \mathrm{Rat})$.

Nous obtenons alors $C^f(P_k(L)) = \mathcal{H}^{s\alpha}(P_k(L) \wedge \mathrm{Rat})$ et $C^f(P(L)) = \bigcup_{k\geq 1} C^f(P_k(L)) = \mathcal{H}^{s\alpha}(P(L) \wedge \mathrm{Rat})$.

□

Nous allons montrer maintenant qu'en fait, on peut remplacer l'intersection avec un langage rationnel par un homomorphisme marqué inverse. Nous allons prouver pour cela que $\forall k \in \mathbb{N}^+$, $P_k(L) \wedge \mathrm{Rat} \subseteq \mathcal{H}^{-1}(P_k(L))$.

<u>*Proposition 9.*</u>

Soit L un cône rationnel fidèle. Pour tout $k \in \mathbb{N}^+$

$$C^f(P_k(L)) = \mathcal{H}^{s\alpha} \circ \mathcal{H}^{-1}(P_k(L))$$

et

$$C^f(P(L)) = \mathcal{H}^{s\alpha} \circ \mathcal{H}^{-1}(P(L)).$$

<u>Preuve</u>.

Soient $L \in P_k(L)$, $L \subseteq X^*$ et $R \subseteq X^*$ un langage rationnel. Alors $L = f(M)$ où $M \in L$ et f est une fonction de commutation k-partitionnée associée à une partition $\pi = \{X_1,\ldots,X_k\}$ de X.

Posons $\bar{X} = \{\bar{x} \ / \ x \in X\}$ et h et g deux homomorphismes définis sur X^* par $\forall x \in X$, $h(x) = x\bar{x}$ et $g(x) = \bar{x}$. Alors $L \cap R = h^{-1}(L \sqcup\!\sqcup g(R))$. Il reste donc à montrer que $L \sqcup\!\sqcup g(R) = f(M) \sqcup\!\sqcup g(R) \in P_k(\mathcal{L})$.
Puisque $\mathcal{L}$ est un cône rationnel fidèle, $M \sqcup\!\sqcup g(R) \in \mathcal{L}$ et on peut facilement vérifier que $f(M) \sqcup\!\sqcup g(R) = f'(M \sqcup\!\sqcup g(R))$ où f' est la fonction de commutation associée à la partition $\pi' = \{X_1 \cup \bar{X}, X_2, \ldots, X_k\}$.
Donc $P_k(\mathcal{L}) \wedge \mathrm{Rat} \subseteq \mathcal{H}^{-1}(P_k(\mathcal{L}))$ ce qui implique que
$C^f(P_k(\mathcal{L})) = \mathcal{H}^{s\alpha}(P_k(\mathcal{L}) \wedge \mathrm{Rat}) \subseteq \mathcal{H}^{s\alpha} \circ \mathcal{H}^{-1}(P_k(\mathcal{L}))$.
Comme il est évident que $\mathcal{H}^{s\alpha} \circ \mathcal{H}^{-1}(P_K(\mathcal{L})) \subseteq C^f(P_k(\mathcal{L}))$, nous en déduisons que $C^f(P_k(\mathcal{L})) = \mathcal{H}^{s\alpha} \circ \mathcal{H}^{-1}(P_k(\mathcal{L}))$ et donc que $C^f(P(\mathcal{L})) = \mathcal{H}^{s\alpha} \circ \mathcal{H}^{-1}(P(\mathcal{L}))$.

□

En regroupant ce résultat et ceux déjà obtenus au corollaire 4, nous obtenons une nouvelle caractérisation des trois familles de langages MULTI-RESET, BNP et Q :

Proposition 10.

$$\text{MULTI-RESET} = C^f_n(\text{COPY}) = \mathcal{H}^{s\alpha}(P(\mathrm{Rat}) \wedge \mathrm{Rat}) = \mathcal{H}^{s\alpha} \circ \mathcal{H}^{-1}(P(\mathrm{Rat}))$$

$$\text{BNP} = C^f_n(\text{Sym}) = \mathcal{H}^{s\alpha}(P(\mathrm{Lin}) \wedge \mathrm{Rat}) = \mathcal{H}^{s\alpha} \circ \mathcal{H}^{-1}(P(\mathrm{Lin}))$$

$$Q = C^f_n(\mathrm{Alg}) = \mathcal{H}^{s\alpha}(P(\mathrm{Alg}) \wedge \mathrm{Rat}) = \mathcal{H}^{s\alpha} \circ \mathcal{H}^{-1}(P(\mathrm{Alg})).$$

CHAPITRE IV

HIÉRARCHIES ET PROBLÈMES DE DÉCIDABILITÉ

A) La hiérarchie des familles de langages $CP^k(RAT)$, $k \geq 0$

B) La hiérarchie des familles de langages $SC^k(RAT)$, $k \geq 0$

C) La hiérarchie des familles de langages $P^k(RAT)$, $k \geq 0$

D) Comparaisons entre familles

E) Problèmes de décidabilité.

Ce dernier chapitre aborde des questions classiques, que soulève la définition de nouvelles familles de langages.

Ainsi, dans les trois premières sections, nous allons montrer que les hiérarchies des familles $SC^k(RAT)$, $CP^k(RAT)$ et $P^k(RAT)$ $(k \geq 0)$ sont infinies : c'est à dire qu'il existe des langages "*fabriqués*" à partir d'un langage rationnel en lui appliquant k fonctions de semi-commutation (resp. commutation partielle et commutation partitionnée) qu'on ne pourra jamais obtenir à partir d'un langage rationnel en lui appliquant moins de k fonctions.

Pour la hiérarchie des familles $SC^k(RAT)$, $k \geq 0$, et $CP^k(RAT)$, $k \geq 0$ l'exemple du langage construit est le même : il est défini sur un alphabet de trois letrres, ce qui permet de voir un peu mieux le résultat de l'application de chaque fonction.

Mais le cas des langages de $P^k(RAT)$ est différent car on montrera qu'on est obligé d'augmenter la taille de l'alphabet avec le nombre de fonctions qu'on applique. Nous obtiendrons, ainsi, des résultats sur la composition des fonctions de commutation.

Le résultat de la 4ème section consiste en une série d'inclusions strictes : $P^k(RAT) \subsetneq CP^k(RAT) \subsetneq SC^k(RAT)$, $\forall k > 0$.

Enfin, dans la 5ème section, nous nous intéressons aux problèmes de décidabilité. La liste des propriétés énoncées n'est sûrement pas exhaustive. Citons seulement un résultat : il est indécidable de déterminer si un langage rationnel et un langage de $P(RAT)$ sont égaux.

A - LA HIERARCHIE DES FAMILLES $CP^n(RAT)$, $n \geq 0$.

Nous allons montrer que les familles $CP^n(RAT)$, $n \geq 0$ forment une hiérarchie infinie : $CP^n(RAT) \subsetneq CP^{n+1}(RAT)$ $\forall n \geq 0$. Le cas $n = 0$ est trivial. Nous allons prouver ce résultat pour $n > 1$ en utilisant le résultat du lemme suivant :

Lemme 1.

Soit k *un entier positif, et* k *mots :* $u_1, u_2, \ldots, u_k$ *de* $\{a,b\}^*$. *Alors* $Com(u_1)\ Com(u_2) \ldots Com(u_k) \subseteq (ab+ba)^k$ *implique* $\forall i \in \{1,\ldots,k\}$, $u_i \in \{ab, ba\}$.

Preuve.

Nous allons d'abord montrer que l'on a toujours $|u_i| = 2\ \forall i \in \{1,\ldots,k\}$

* Si $|u_i| > 2$, u_i contient au moins 2 occurrences de a ou de b. Supposons $Com(u_i) = Com(aabu_i')$, avec $u_i' \in \{a,b\}^*$. Si $\alpha aabu_i'\gamma = \alpha aab\beta \in (ab+ba)^k$

$$\Rightarrow \alpha a \in (ab + ba)^+ \text{ et } ab\beta \in (ab + ba)^+$$

$$\Rightarrow \alpha a \in (ab + ba)^+ \text{ et } \beta \in (ab + ba)^+$$

Or $\alpha aabu_i'\gamma \in Com(u_1)\ldots Com(u_k) \Rightarrow \alpha baa\beta \in (ab + ba)^+$

$$\Rightarrow \alpha ba \in (ab + ba)^{\pm} \text{ et } a\beta \in (ab + ba)^+$$

$$\Rightarrow \alpha \in (ab + ba)^+ \text{ et } a\beta \in (ab + ba)^+$$

et on a une contradiction.
Donc $|u_i| < 2$

* On écarte alors immédiatement les cas où $|u_i| = 0$ ou $|u_i| = 1$, qui impliqueraient qu'il existe un indice j tel que $|u_j| > 2$.

Montrons maintenant le résultat énoncé, par récurrence sur k.

* Si k = 1, $com(u) \subseteq \{ab, ba\}$ implique de manière évidente que u = ab ou ba.

* Supposons le résultat vrai $\forall n \leq k$, et montrons qu'il l'est encore à l'ordre k+1 :

$$Com(u_1) \ldots Com(u_k)\ Com(u_{k+1}) \subseteq (ab + ba)^{k+1}$$

On sait alors que $\forall i \in \{1,\dots,k+1\}$, $|u_i| = 2$. Les cas u_1 = aa ou u_1 = bb sont à écarter immédiatement. Il reste alors u_1 = ab ou u_1 = ba. Ce qui implique que

$$Com(u_2) \dots Com(u_{k+1}) \subseteq (ab + ba)^k,$$

et donc, par hypothèse de récurrence, u_i = ab ou ba $\forall i \in \{2,\dots,k+1\}$.

□

Il est clair que $\forall i \geq 0$, $CP^i(RAT) \subseteq CP^{i+1}(RAT)$.
Nous allons maintenant montrer l'inclusion stricte en exhibant un exemple de langage appartenant à $CP^{i+1}(RAT)$ et n'appartenant pas à $CP^i(RAT)$.

Proposition 1.

Soient $R = ((ab)^+c)^{k+1}(ab)^+$ *avec* $k \geq 0$, f_o *la fonction de commutation partielle associée à la relation* $C_o = \{(a,b)\}$, f_1 *la fonction associée à* $C_1 = \{(b,c)\}$ *et* f_2 *la fonction associée à* $C_2 = \{(a,c)\}$.
Alors

$$L = (f_2 \circ f_1)^k \in f_o(R) \quad CP^{2k+1}(RAT) - CP^{2k}(RAT)$$

et

$$L' = f_1(L) \in CP^{2k+2}(RAT) - CP^{2k+1}(RAT).$$

Preuve.

Supposons qu'il existe un langage rationnel K et 2k fonctions de commutation $g_1, g_2,\dots,g_{2K}$ tels que $L = g_{2k} \circ \dots \circ g_1(K)$.

Nous pouvons faire alors plusieurs remarques.

* On ne peut pas trouver de fonctions g_i associées à la relation $\{(a,c), (b,c)\}$ car $L \cap c^+(a+b)^+ = \emptyset$

* De même, on ne peut avoir une fonction g_i associée à la relation $\{(a,c), (a,b)\}$ car $L \cap a^+(b+c)^+ = \emptyset$.

* Même chose avec la relation $\{(b,c), (a,b)\}$.
On en conclut donc que $\forall i \in \{1,\dots,2k\}$, $g_i \in \{f_o, f_1, f_2\}$.

* On a forcément au moins une des fonctions g_i qui est f_o. En effet $\mathrm{Proj}(\{a,b,c\}, \{a,b\}))(L) = (D_1^+)^{k+2}$ (où D_1^+ est le langage de Dyck sur $\{a,b\}^+$). Et ce langage n'est pas rationnel.

On en déduit aussi que $\forall i \in \{1,\dots,2k\}$,

$$\mathrm{Proj}(X, \{a,b\})\ (g_i \circ g_{i-1} \circ \dots \circ g_1(K)) \subseteq (D_1^+)^{k+2},$$

où X désigne l'alphabet $\{a, b, c\}$.

Posons $L_1 = \{u \in L, |u|_{a,b} = 2(k+2)\}$.
L_1 contient les mots de la forme $(ab + ba)\ c^{k+1}\ (ab + ba)^{k+1}$.
Comme $L = g_{2k} \circ \dots \circ g_1(K)$, on peut trouver un mot de K :
$u = u_1\ c\ u_2\ c \dots c\ u_{k+2}$ tel que $g_{2k} \circ \dots \circ g_1(u) \in L_1$. Et comme il existe i tel que $g_i = f_o$, on a alors

$$\mathrm{Com}(u_1)\ \mathrm{Com}(u_2) \dots \mathrm{Com}(u_{k+2}) \subseteq (ab + ba)^{k+2} = \mathrm{proj}(X, \{a,b\})(L)$$

On déduit alors du lemme précédent que $u_j = ab$ ou ba, $\forall j \in \{1,\dots,k+2\}$.
Nous pouvons alors affirmer que g_1 est l'unique opérateur égal à f_o ; en effet supposons que $g_1 = f_1$ (ou f_2).

Alors $u \xrightarrow[c_1]{*} u'_1\ c\ u'_2\ c \dots c\ u'_{k+1}$ avec un mot u'_i de longueur supérieure ou égale à 3.
Mais on doit encore avoir $f_o(u') \subseteq L_1$ donc

$$\mathrm{Com}(u'_1) \dots \mathrm{Com}(u'_{k+1}) \subseteq (ab + ba)^{k+2}$$

et on a une contradiction avec le lemme .
Donc

$$g_1 = f_o.$$

Montrons qu'il est l'unique opérateur égal à f_o : supposons que $g_i = f_o$, avec $i > 2$ (on peut supposer que $g_j \neq g_{j+1}$ $\forall j \in \{1,\dots,2k-1\}$).
Alors $g_{i-1} \circ \dots \circ g_1(u) \ni v_1\ c\ v_2\ c \dots c\ v_{k+2} = v$, où un des v_j est de longueur supérieure strictement à 2.

En remarquant qu'on doit alors avoir $f_o(v) \in L_1$, donc $Com(v_1) \ldots Com(v_{k+2}) \subseteq (ab + ba)^{k+2}$, on a encore la même contradiction.
Donc $g_1 = f_o$ et $g_i = f_1$ ou f_2 $\forall i > 1$.

Nous allons maintenant montrer qu'on ne pourra alors jamais obtenir le mot $w = ab\ c^{k+1}\ (ab)^{k+1}$ $(w \in L_1)$.
Pour obtenir ce mot, on doit avoir $g_1(u) = (abc)^{k+1}\ ab$. Alors, à partir de $g_1(u)$, pour atteindre w, il faut au moins appliquer 2k fois les fonctions f_1 ou f_2 : raisonnons par récurrence sur k :

- pour $k = 1$: $g_1(u) = abcabcab$.

Les seules fonctions possibles étant f_1 et f_2, il est clair qu'il faudra appliquer f_1 puis f_2 pour obtenir $ab\ c^2(ab)^2$.
Donc

$$g_2 \circ g_1(u) \neq f_2 \circ f_1 \circ f_2(u).$$

Supposons donc que si $g_1(u) = (abc)^{k+1}ab$, il faut 2k permutations pour obtenir $ab\ c^{k+1}(ab)^{k+1}$ (donc pour amener la dernière occurrence de la lettre c devant la première occurrence du sous-mot ab), alors :

si $g_1(u) = (abc)^{k+2}\ ab = abc\ (abc)^{k+1}\ ab$, il faut 2k permutations pour obtenir le mot $abc^{k+1}abcab$ et donc il faut encore appliquer successivement les fonctions f_1 et f_2 pour obtenir $ab\ c^{k+2}\ (ab)^{k+3}$, donc en tout 2k+2 opérations f_1 et f_2.
Donc

$$ab\ c^{k+1}\ (ab)^{k+1} \notin g_{2k} \ \ldots \ g_1(u).$$

Et enfin, pour montrer que $L' \notin CP^{2k+1}(RAT)$, le raisonnement sera tout à fait identique : on montrera qu'on ne peut pas atteindre le mot $ac^{k+1}b(ab)^{k+1}$ en moins de 2k+1 opérations à partir du mot $g_1(u) = (abc)^{k+1}ab$.

□

__Proposition.__

$$\forall n \geq 1,\ CP^n(RAT) \subsetneq CP^{n+1}(RAT).$$

B - LA HIERARCHIE INFINIE DES FAMILLES $SC^k(RAT)$, $k \geq 0$.

Pour montrer ce résultat, nous allons utiliser le même exemple que précédemment, mais le cheminement de la démonstration sera un peu différent car le nombre de fonctions de semi-commutation qu'on peut définir sur un alphabet de 3 lettres est plus important que celui des fonctions de commutation partielle et on ne peut pas à priori exclure une majorité de ces fonctions. Mais la contradiction à laquelle on aboutira reste la même.

Proposition 2.

$\forall k \geq 0$, $SC^k(RAT) \subsetneq SC^{k+1}(RAT)$, *ou la hiérarchie des familles* $SC^k(RAT)$, $k \geq 0$ *est infinie.*

Preuve.

Pour $k \geq 0$, nous allons considérer le langage rationnel $R=((ab)^+c)^{k+1}(ab)^+$ et les 3 fonctions de semi-commutation suivantes :

f_o sera associée à la relation $C_o = \{ab \leftrightarrow ba\}$;

f_1 sera associée à la relation $C_1 = \{bc \rightarrow cb\}$;

f_2 sera associé à la relation $C_2 = \{ac \rightarrow ca\}$.

Posons $L = (f_2 \circ f_1)^k \circ f_o(R)$. Alors $L \in SC^{2k+1}(RAT)$. Nous allons montrer que $L \notin SC^{2k}(RAT)$.
Supposons pour cela qu'il existe un langage rationnel K et 2k fonctions de semi-commutations $g_i (i \in \{1,\dots,2k\})$ associées à des relations S_i $(i \in \{1,\dots,2k\})$ vérifiant $L = g_{2k} \circ \dots \circ g_1(K)$.
Nous pouvons alors faire les constatations suivantes :

1°) $\forall i \in \{1,\dots,2k\}$, S_i ne peut pas contenir les relations $ac \rightarrow ca$ et $bc \rightarrow cb$ car $L \cap c^+(a+b)^+ = \emptyset$.
De même, S_i ne peut pas contenir $\{ba \rightarrow ab, bc \rightarrow cb\}$ car $L \cap (a+c)^+b^+ = \emptyset$, ni $\{ab \rightarrow ba, ac \rightarrow ca\}$ car $L \cap (b+c)^+a^+ = \emptyset$.

2°) Il est facile de voir que $Proj(L, \{a,b\}) = (D_1^+)^{k+2} \notin RAT$.
Donc, au moins une des relations S_i doit contenir une relation qui fait commuter a et b : $ab \to ba$ ou $ba \to ab$ ou $ab \leftrightarrow ba$ (car $Proj(K, \{a,b\}) \in RAT$).
Montrons qu'en fait, il existe i et $j \in \{1,\dots,2k\}$ tels que S_i contient la relation $ab \to ba$ et S_j la relation $ba \to ab$ (on peut avoir $i = j$).

Supposons le contraire : $\exists i \in \{1,\dots,2k\}$ tel que S_i contient la relation $ba \to ab$, et $\forall j \in \{1,\dots,2k\}$, S_j ne contient pas la relation $ab \to ba$.

$Proj(L, \{a,b\}) = (D_1^+)^{k+2} \Rightarrow g_{2k/\{a,b\}} \circ \dots \circ g_{1/\{a,b\}} \dots (proj(K, \{a,b\})) = L'$

où L' vérifie $(D_1^+)^{k+2} \subseteq L' \subseteq D_1^+$.
Et d'après l'hypothèse, $g_{2k/\{a,b\}} \circ \dots \circ g_{1/\{a,b\}} = f$ où f est la fonction associée à la relation $ba \to ab$.
Alors l'égalité $f(Proj(K, \{a,b\})) = L'$ est impossible : le mot $((b^n a^n)c)^{k+1} b^n a^n$ est dans L, $\forall n \in 1$. Donc $(b^n a^n)^{k+2} \in L' \Rightarrow \exists w \in proj(K, \{a,b\})$ tel que $(b^n a^n)^{k+2} \in f(w) \Rightarrow w \in f^{-1}((b^n a^n)^{k+2})$ où f^{-1} est associée à la relation $ab \to ba$. On en conclut que $w = b^n w_1$. En choisissant n suffisamment grand, on aura alors un facteur itérant dans w qui est dans b^+. Ce qui est impossible puisque $Proj(K, \{a,b\}) \subseteq \{u \in \{a,b\}^+ / |u|_a = |u|_b\}$.
On déduira aisément la même chose en supposant que S_i contient la relation $ba \to ab$.

Nous allons maintenant nous intéresser au sous-ensemble de L défini par $L_1 = \{u \in L \ / \ |u|_a = |u|_b = k+2\}$, et plus particulièrement au mot $w' = ab\, c^{k+1}(ab)^{k+1}$ qui appartient à L_1, pour montrer qu'on ne pourra jamais l'obtenir à partir d'un mot de K en lui appliquant la suite $g_1,\dots,g_{2K}$.
Il est clair que $\forall u \in L_1$, $proj(u, \{a,b\}) \in (ab + ba)^{k+2}$. D'autre part, il existe $u_1 \in K$ tel que $g_{2k} \circ \dots \circ g_1(u_1) \ni w'$. Comme $K \subseteq L$, on en déduit que $u_1 \in L_1$ et donc $proj(u_1, \{a,b\}) \in (ab+ba)^{k+2}$.
Soit $(u_1, u_2,\dots,u_{2k}, u_{2k+1} = w')$ la suite de mots qui représentent le passage de u_1 à w' : $\forall i \geq 1$, $u_i \xrightarrow[S_i]{*} u_{i+1}$: nous pouvons constater que :

- $\forall i \in \{1,\dots,2k\}$, $u_i \in L_1$, donc $proj(u_i, \{a,b\}) \in (ab+ba)^{k+2}$

•• $\forall i \geq 1$, $g_i(u_{i+1}) \subseteq L_1$, donc $proj(g_i(u_{i+1}), \{a,b\}) \subseteq (ab+ba)^{k+2}$

Soit $i = \sup\{j \in \{1,\dots,2k\} / S_j$ contient la relation $ba \to ab\}$.
(D'après la remarque 2, i existe). Nous allons alors montrer que :

1°) $\forall j \geq i+1$, $\mathrm{proj}(u_j, \{a,b\}) = (ab)^{k+2}$
2°) $u_{i+1} = (abc)^{k+1}\, ab$.

1°) Supposons qu'il existe $j \geq i+1$ tel que $\mathrm{proj}(u_j, \{a,b\}) \notin (ab)^{k+2}$.

• Soit $\forall k > j$, S_k ne contient pas la relation $ab \to ba$; on devrait alors avoir $\mathrm{proj}(u_j, \{a,b\}) = \mathrm{proj}(w', \{a,b\}) = (ab)^{k+2}$, ce qui est faux.

• Soit $\exists k > j$ tel que S_k contient la relation $ab \to ab$. On doit alors avoir $g_{k/\{a,b\}}(\mathrm{proj}(u_j, \{a,b\})) \ni (ab)^{k+2} = \mathrm{proj}(w', \{a, b\})$ et on peut écrire que $\mathrm{proj}(u_j, \{a,b\}) = ab^{\alpha}ba\ x$, où $\alpha \geq o$ et $x \in (ab + ba)^{k+2-(\alpha+}$
Le résultat du lemme II.8 nous dit que :

$\forall v \in g_{k/\{a,b\}}(\mathrm{proj}(u_j, \{a,b\}))$, si $v = v_1 v_2$ avec $v_1 = 2\alpha+1$, on a $|v_1|_b \geq |(ab)^{\alpha} b|_b = \alpha+1$. Or le facteur gauche de longueur $2\alpha+1$ de $\mathrm{proj}(w', \{a,b\})$ contient α occurrences de la lettre b. On en déduit que $\mathrm{proj}(w', \{a,b\}) \notin g_{k/\{a,b\}}(\mathrm{proj}(u_j, \{a,b\}))$.

2°) Nous venons de voir que $\mathrm{proj}(u_{i+1}, \{a,b\}) = (ab)^{k+2}$;
d'autre part, $g_i(u_{i+1}) \subseteq L_1$.
Posons $u_{i+1} = x_1\, c\, x_2\, c \dots c\, x_{k+2}$, avec $x_1 x_2 \dots x_{k+2} = (ab)^{k+2}$.
Si il existe j tel que $|xj| \neq 2$, on peut supposer que $|x_j| > 2$.

x_j contient donc un facteur de la forme aba ou bab : il est alors clair qu'en appliquant une seule fois la fonction g_i (en dérivant ba en ab) au mot u_j, on sort de L_1, car on obtient un mot dont la projection sur $\{a,b\}$ contient le sous mot aabb ou bbaa, ce qui est impossible.

Nous pouvons maintenant conclure : $w' \in g_{2k} \circ \dots \circ g_{i+1}((abc)^{k+1}\, ab)$, et seules les relations faisant intervenir la lettre c peuvent être utilisées. Nous avons vu dans la démonstration du paragraphe A), qu'il fallait au moins 2k fonctions pour réaliser ce passage, et nous en avons ici au maximum 2k-1.

$\square$

C - LA HIERARCHIE DES FAMILLES $P^k(RAT)$, $k \geq 0$.

De même que pour les familles $CP^k(RAT)$, et $SC^k(RAT)$ $k \geq 0$, nous allons montrer qu'on a une hiérarchie infinie. Mais la démonstration sera ici un peu plus compliquée : pour les commutations partielles ou les semi-commutations, les lettres qui ne "*bougent*" pas, (qui ne commutent avec rien pour une relation donnée), jouent le rôle de barrière dans le mouvement des autres lettres. Mais dans le cas des commutations partitionnées, toute relation permet à chaque lettre de commuter avec au moins une autre lettre.
De plus, pour un alphabet X donné, le nombre de partitions (et donc de fonctions de commutation que l'on peut définir) est fini. Et nous allons montrer que la composition de k fonctions de commutation partitionnée ($k \geq 3$) dont 2 sont identiques se réduit à la composition de (k-1) fonctions. La taille de l'alphabet sur lequel nous donnerons les exemples grandira donc avec le nombre de fonctions qu'on compose.

Nous allons donc énoncer quelques résultats concernant la composition de fonctions de commutation partitionnée ; puis nous montrerons que $P(RAT) \subsetneq P^2(RAT)$ en exhibant un langage rationnel R et deux fonctions de commutation f_1 et f_2 tels que $f_2 \circ f_1(R)$ n'appartient pas à P(RAT) et de plus nous montrerons que $f_2 \circ f_1(R)$ n'appartient pas à $P^k(RAT)$, $\forall k \geq 3$ et que $f_2 \circ f_1(R) = g_2 \circ g_1(K)$, pour 2 fonctions g_1 et g_2 et un rationnel K, implique $f_2 = g_2$ et $f_1 = g_1$. Nous montrerons alors que $\forall k \geq 0$, $P^k(RAT) \subsetneq P^{k+1}(RAT)$.

Notation.

Nous noterons P^* l'ensemble des fonctions obtenues en composant des fonctions de commutation partitionnée.

Lemme 2.

Soient $f \in P$, *et* $u,v \in P^*$.
Alors

$$f \circ u \circ f \circ v \circ f = f \circ u \circ v \circ f.$$

Preuve.

Nous nous placerons dans le cas où f est une fonction de commutation 2-partitionnée, c'est à dire f est associée à une partition $\pi = \{Y, \bar{Y}\}$ de X. La démonstration sera identique pour une partition quelconque.
$\forall w \in X^*$, $\forall w' \in f \circ u \circ f \circ v \circ f(w)$, il existe des mots w_1, w_2, w_3, w_4 vérifiant.

$$w \overset{*}{\underset{c_\pi}{\longleftrightarrow}} w_1 \overset{*}{\underset{v}{\longleftrightarrow}} w_2 \overset{*}{\underset{c_\pi}{\longleftrightarrow}} w_3 \overset{*}{\underset{u}{\longleftrightarrow}} w_4 \overset{*}{\underset{c_\pi}{\longleftrightarrow}} w'$$

$$w \overset{*}{\underset{c_\pi}{\longleftrightarrow}} w_1 \iff w \overset{*}{\underset{c_\pi}{\longleftrightarrow}} y_1y_2 \overset{*}{\underset{c_\pi}{\longleftrightarrow}} w_1 \text{ avec } y_1 \in Y^*,\ y_2 \in \bar{Y}^*.$$

de même $w_4 \overset{*}{\underset{c_\pi}{\longleftrightarrow}} w' \iff w_4 \overset{*}{\underset{c_\pi}{\longleftrightarrow}} z_1\, z_2 \overset{*}{\underset{c_\pi}{\longleftrightarrow}} w'$, avec $z_1 \in Y^*$, $z_2 \in \bar{Y}^*$.

Donc $z_1\, z_2 \in f \circ u \circ f \circ v \circ f(y_1y_2)$. D'après le lemme II.1, on peut écrire

$$\mathrm{proj}(z_1\, z_2, Y) \in f \circ u \circ f \circ v \circ f(\mathrm{proj}(y_1\, y_2, Y))$$

et

$$\mathrm{proj}(z_1\, z_2, \bar{Y}) \in f \circ u \circ f \circ v \circ f(\mathrm{proj}(y_1\, y_2, Y)\ ),$$

c'est à dire $z_1 \in f \circ u \circ f \circ v \circ f(y_1)$ et $z_2 \in f \circ u \circ f \circ v \circ f(y_2)$.
Mais pour obtenir z_1 (resp. z_2) à partir de y_1 (resp. y_2), aucune commutation relative à c ne sera utilisée puisque $y_1 \in Y^*$ et $y_2 \in \bar{Y}^*$. Donc en fait, $z_1 \in u \circ v(y_1)$ et $z_2 \in u \circ v(y_2)$. Donc $z_1z_2 \in u \circ v(y_1y_2)$, et on conclut ainsi que $w' \in f \circ u \circ v \circ f(w)$, et $f \circ u \circ f \circ v \circ f \subseteq f \circ u \circ v \circ f$.
Comme l'inclusion inverse est évidente, l'égalité énoncée est vérifiée.

$\square$

Lemme 3.

Soit X *un alphabet et* $f, u_1, u_2, \ldots, u_k$ *des fonctions de commutation partitionnée définies sur* X. *On peut alors trouver* k *fonctions de commutation partitionnée* $v_1, v_2, \ldots, v_k$ *vérifiant* $f \circ u_1 \circ u_2 \ldots \circ u_k \circ f = v_1 \circ v_2 \circ \ldots \circ v_k$.

Preuve.

Nous démontrerons le résultat pour des fonctions de commutation 2 partitionnées, l'extension au cas des partitions quelconques se fait facilement.

Le raisonnement est basé sur une récurrence sur k.
Posons $\pi = \{Y, \bar{Y}\}$ la partition qui définit f et $\pi_1 = \{X_1, \bar{X}_1\}$ la partition qui définit u_1.

Nous allons montrer que $f \circ u_1 \circ f = v$, ou v est la fonction associée à la partition $\pi' = \{Y \cap X_1, Y \cap \bar{X}_1, \bar{Y} \cap X_1, \bar{Y} \cap \bar{X}_1\}$.

- $\forall w \in X^*$, l'inclusion $f \circ u_1 \circ f(w) \subseteq v(w)$ est triviale.
- $\forall w \in X^*$, montrons que $v(w) \subseteq f \circ u_1 \circ f(w)$.

$\forall w' \in v(w)$, il existe une dérivation $w \xrightarrow[c_{\pi'}]{*} w'$.

Si cette dérivation est de longueur nulle, $w' = w \in f \circ u_1 \circ f(w)$.
Supposons le résultat vrai pour toute dérivation de longueur $\leq \ell$.

Alors si $w \xrightarrow[c_{\pi'}]{\ell+1} w'$, on peut écrire $w \xrightarrow[c_{\pi'}]{1} w_1 \xrightarrow[c_{\pi'}]{\ell} w'$.

$w \xrightarrow[c_{\pi'}]{1} w_1 \Rightarrow w_1 \in f \circ u_1 \circ f(w)$ car $w_1 \in f(w)$ ou $w_1 \in u_1(w)$

$w_1 \xrightarrow[c_{\pi'}]{\ell} w' \Rightarrow w' \in f \circ u_1 \circ f(w_1)$ par hypothèse de récurrence.

Donc $w' \in f \circ u_1 \circ f \circ f \circ u_1 \circ f(w) = f \circ u_1 \circ f \circ u_1 \circ f(w) = f \circ u_1 \circ f(w)$ d'après le lemme 2. Ainsi $v(w) \subseteq f \circ u_1 \circ f(w)$, $\forall w \in X^*$.
Supposons maintenant que $\forall \alpha \leq k$, $f \circ u_1 \circ u_2 \circ \ldots \circ u_\alpha \circ f = v_1 \circ \ldots \circ v_\alpha$.
Alors $f \circ u_1 \circ u_2 \circ \ldots \circ u_{k+1} \circ f = f \circ u_1 \circ \ldots \circ u_k \circ f \circ f \circ u_{k+1} \circ f$ (lemme 2) $= v_1 \circ \ldots \circ v_k \circ f \circ u_{k+1} \circ f$ (hyp. de rec) $= v_1 \circ v_2 \circ \ldots \circ v_k \circ v_{k+1}$ (hyp rec). On peut aussi ajouter qui si u_i est associée à la partition $\{X_i, \bar{X}_i\}$ de X, v_i sera associée à la partition de X $\{X_i \cap X, \bar{X}_i \cap X, X_i \cap \bar{X}, \bar{X}_i \cap \bar{X}\}$.

$\square$

Lemme 4.

Soient $f_1, f_2, \ldots, f_k$, k *fonctions de commutation partitionnée définies sur* X. *Alors* :

$$f_1 \circ f_2 \circ \ldots \circ f_k = f_k \circ \ldots \circ f_1 \iff f_1 \circ f_2 \ldots \circ f_k = f \in P.$$

Preuve.

$$\Leftarrow) \quad f_1 \circ f_2 \circ \ldots \circ f_k = f \Rightarrow f_k \circ \ldots \circ f_1 = f \quad ?$$

• Montrons que $f_k \circ \ldots \circ f_1(w) \subseteq f(w),\ \forall w \in X^*$

$f_1 \circ f_2 \circ \ldots \circ f_k(w) = f(w)\ \forall w \in X^* \Rightarrow \forall i \in \{1,\ldots,k\},\ f_i(w) \subseteq f(w)$
$\Rightarrow f_k \circ \ldots \circ f_1(w) \subseteq f(w)$.

Réciproquement, montrons que $\forall w \in X^*$, $f(w) \subseteq f_k \circ \ldots \circ f_1(w)$.
Notons C_π la relation associée à f, et C_{π_i} la relation associée à f_i, $\forall i \in \{1,\ldots,k\}$.
$\forall w \in X^*$, $\forall w' \in f(w)$, on peut écrire que $w \in f(w')$.
Donc $w' \overset{*}{\underset{C_{\pi_k}}{\longleftrightarrow}} w_1 \overset{*}{\underset{C_{\pi_{k-1}}}{\longleftrightarrow}} \ldots \overset{*}{\underset{C_{\pi_1}}{\longleftrightarrow}} w$ puisque $f = f_1 \circ f_2 \circ \ldots \circ f_k$.

$$\Rightarrow \quad w \overset{*}{\underset{C_{\pi_1}}{\longleftrightarrow}} w_{k-1} \overset{*}{\longleftrightarrow} \ldots \overset{*}{\underset{C_{\pi_k}}{\longleftrightarrow}} w' \Rightarrow w' \in f_k \circ \ldots \circ f_1(w).$$

$$\Rightarrow) \quad f_1 \circ f_2 \circ \ldots \circ f_k = f_k \circ \ldots \circ f_1 \Rightarrow f_1 \circ f_2 \circ \ldots \circ f_k = f \in P \quad ?$$

Nous utiliserons ici, un raisonnement par récurrence sur k. Mais d'abord, précisons la notation qui sera utilisée :

si f est associée à une partition $\{X_1, X_2, \ldots, X_n\}$ et si g est associée à une partition $\{Y_1, Y_2, \ldots, Y_p\}$, nous noterons f+g la fonction associée à la partition $\{X_i \cap X_j,\ i \neq j,\ i \in \{1,\ldots,n\},\ j \in \{1,\ldots,p\}\}$.

Montrons maintenant le résultat pour k = 2 :

si $f_1 \circ f_2 = f_2 \circ f_1$, nous pouvons écrire :

$f_1 \circ f_2 = f_1 \circ f_1 \circ f_2 \circ f_2 = f_1 \circ f_2 \circ f_1 \circ f_2 = (f_1 + f_2) \circ f_2$ (lemme 3)
$= f_1 + f_2$ car $f_2 \subseteq f_1 + f_2$.

Supposons le résultat vrai pour (k-1) fonctions :

si $f_1 \circ \ldots \circ f_{k-1} = f_{k-1} \circ \ldots \circ f_1$ alors $f_1 \circ \ldots \circ f_{k-1} = f_1 + f_2 + \ldots + f_{k-1} \in P$.

Alors, pour k fonctions :

$$
\begin{aligned}
f_1 \circ f_2 \circ \ldots \circ f_k &= f_1 \circ f_1 \circ f_2 \circ \ldots \circ f_k \circ f_k \\
&= f_1 \circ f_k \circ f_{k-1} \circ \ldots \circ f_1 \circ f_k \\
&= (f_1 + f_k) \circ (f_1 + f_{k-1}) \circ \ldots \circ (f_1 + f_2) \circ f_k \quad \text{(lemme 3)} \\
&= f_k \circ (f_1 + f_k) \circ \ldots \circ (f_1 + f_2) \circ f_k \text{ car } f_k \subseteq f_1 + f_k \\
&= (f_1 + f_k) \circ (f_1 + f_k + f_{k-1}) \circ \ldots \circ (f_1 + f_k + f_2) \quad \text{(lemme 3)}
\end{aligned}
$$

On a réduit à (k-1) fonctions.

$$
\begin{aligned}
f_k \circ \ldots \circ f_1 &= f_k \circ f_k \circ f_{k-1} \circ \ldots \circ f_1 \circ f_1 \\
&= f_k \circ f_1 \circ f_2 \circ \ldots \circ f_k \circ f_1 \\
&= f_k \circ (f_1 + f_2) \circ \ldots \circ (f_1 + f_k) \quad \text{(lemme 3)} \\
&= f_k \circ (f_1 + f_2) \circ \ldots \circ (f_1 + f_k) \circ f_k \text{ car } f_k \subseteq f_1 + f_k \\
&= (f_1 + f_k + f_2) \circ \ldots \circ (f_1 + f_k + f_{k-1}) \circ (f_1 + f_k) \quad \text{(lemme 3)}
\end{aligned}
$$

Comme $f_1 \circ f_2 \circ \ldots \circ f_k = f_k \circ \ldots \circ f_1$, on a donc

$$
(f_1 + f_k) \circ (f_1 + f_k + f_{k-1}) \circ \ldots \circ (f_1 + f_{k+2}) = (f_1 + f_k + f_2) \quad \ldots
$$
$$
\ldots \circ (f_1 + f_k + f_{k-1}) \circ (f_1 + f_k)
$$

Nous pouvons alors appliquer l'hypothèse de récurrence à ces (k-1) fonctions, et conclure que $f_1 \circ f_2 \circ \ldots \circ f_k = f_1 + f_2 + \ldots + f_k$.

□

Lemme 5.

Soient X *un alphabet,* f_1 *la fonction de commutation associée à la partition* $\pi_1 = \{X_1, X_2, X_3 \cup X_4, X_5 \ldots, X_p\}$ *et* f_2 *la fonction de commutation associée à la partition* $\pi_2 = \{X_1 \cup X_2, X_3, X_4, X_5, \ldots X_p\}$.
Alors $f_1 \circ f_2 = f_2 \circ f_1 = f$, *où* f *est la fonction de commutation associée à la partition* $\pi = \{X_1, X_2, X_3, X_4, \ldots X_p\}$.

Preuve.

- $\forall w \in X^*$, $f_1(w) \subseteq f(w)$ et $f_2(w) \subseteq f(w)$, donc $f_1 \circ f_2(w) \subseteq f(w)$, et $f_2 \circ f_1(w) \subseteq f(w)$.

- $\forall w \in X^*$, $\forall w' \in f(w)$, $\forall i \in \{1,\ldots,p\}$, $\text{proj}(w', X_i) = \text{proj}(w, X_i)$.

$$w \xrightarrow[c_{\pi_1}]{*} \text{proj}(w, X_1)\, \text{proj}(w, X_2)\, \text{proj}(w, X_3 \cup X_4)\, \text{proj}(w, X_5) \ldots \text{proj}(w, X_p)$$

$$\xrightarrow[c_{\pi_1}]{*} \text{proj}(w', X_1 \cup X_2)\, \text{proj}(w, X_3 \cup X_4)\, \text{proj}(w', X_5) \ldots \text{proj}(w', X_p)$$

car $\text{proj}(w', X_1 \cup X_2) \in \text{proj}(w, X_1) \sqcup\!\sqcup\, \text{proj}(w, X_2)$ et $\forall i \geq 5$, $\text{proj}(w, X_i) = \text{proj}(w', X_i)$.

$$\xrightarrow[c_{\pi_2}]{*} \text{proj}(w', X_1 \cup X_2)\, \text{proj}(w, X_3)\, \text{proj}(w, X_4)\, \text{proj}(w', X_5) \ldots \text{proj}(w', X_p)$$

$$= \text{proj}(w', X_1 \cup X_2)\, \text{proj}(w', X_3)\, \text{proj}(w', X_4) \ldots \text{proj}(w', X_p).$$

Soit w_1 ce mot. $w_1 \in f_2 \circ f_1(w)$

$$f_2(w_1) = \text{proj}(w_1, X_1 \cup X_2) \sqcup\!\sqcup\, \text{proj}(w_1, X_3) \sqcup\!\sqcup \ldots \sqcup\!\sqcup\, \text{proj}(w_1, X_p)$$

$$= \text{proj}(w', X_1 \cup X_2) \sqcup\!\sqcup\, \text{proj}(w', X_3) \sqcup\!\sqcup \ldots \sqcup\!\sqcup\, \text{proj}(w', X_p)$$

Donc $f_2(w_1)$ contient w', et $w' \in f_2 \circ f_1(w)$.
Alors $\forall w \in X^*$, $f_2 \circ f_1(w) = f(w) \Rightarrow$ (lemme 5) $f_1 \circ f_2 = f_2 \circ f_1 = f$.

$\square$

<u>*Lemme 6.*</u>

Soient X *un alphabet,* f_1 *la fonction de commutation associée à la partition* $\pi_1 = \{X_1, X_2 \cup X_3, X_4, \ldots, X_p\}$ *et* f_2 *la fonction associée à la partition* $\pi_2 = \{X_1 \cup X_2, X_3, X_4, \ldots, X_p\}$. *Alors* $f_1 \circ f_2 = f_2 \circ f_1 = f$, *où* f *est la fonction associée à la partition* $\pi = \{X_1, X_2, \ldots, X_p\}$.

<u>Preuve.</u>

- $\forall w \in X^*$, $f_1(w) \subseteq f(w)$ et $f_2(w) \subseteq f(w)$ donc $f_1 \circ f_2(w) \subseteq f(w)$ et $f_2 \circ f_1(w) \subseteq f(w)$.

- $\forall w \in X^*$, $\forall w' \in f(w)$, $\forall i \in \{1,\ldots,p\}$, $\text{proj}(w, X_i) = \text{proj}(w', X_i)$

$$w \xrightarrow[c_{\pi_1}]{*} \text{proj}(w, X_1)\ \text{proj}(w, X_2 \cup X_3)\ \text{proj}(w, X_4) \ldots \text{proj}(w, X_p)$$

$$\xrightarrow[c_{\pi_1}]{*} u_1\ \text{proj}(w', X_4) \ldots \text{proj}(w', X_p)$$

où u_1 est défini par : $\text{proj}(u_1, X_1 \cup X_2) = \text{proj}(w', X_1 \cup X_2)$ qui est inclus dans $\text{proj}(w, X_1) \sqcup\!\sqcup\ \text{proj}(w, X_2)$, $\text{proj}(u_1, X_3) = \text{proj}(w, X_3)$.

$$\xrightarrow[c_{\pi_2}]{*} \text{proj}(u_1, X_1 \cup X_2)\ \text{proj}(u_1, X_3)\ \text{proj}(w', X_4) \ \ldots\ \text{proj}(w', X_p)$$

$$= \text{proj}(w', X_1 \cup X_2)\ \text{proj}(w', X_3) \ \ldots\ \text{proj}(w', X_p)$$

$$= w_1$$

$w_1 \in f_2 \circ f_1(w)$ et $f_2(w_1) = \text{proj}(w', X_1 \cup X_2) \sqcup\!\sqcup\ \text{proj}(w', X_3) \ldots \sqcup\!\sqcup\ \text{proj}(w', X_p)$

contient donc w'.
Donc $f(w) \subseteq f_2 \circ f_1(w)$ et ainsi $f(w) = f_2 \circ f_1(w) = f_1 \circ f_2(w)$.

$\square$

Nous allons maintenant exhiber un exemple de langage de $P^2(\text{RAT})$ qui n'est pas dans $P(\text{RAT})$. Le résultat obtenu sera plus fort, puisque pour le langage choisi : $L = f_2 \circ f_1(R)$, avec $R \in \text{RAT}$ et f_1 et f_2 des fonctions de commutation, nous allons montrer que f_1 et f_2 sont les seules fonctions possibles,

et composées dans cet ordre, qui peuvent être utilisées pour obtenir L à partir d'un langage rationnel.
Dans les deux lemmes suivants on désignera par :

* X l'alphabet $\{a_1, a_2, b_1, b_2\}$

* f_1 la fonction de commutation partitionnée associée à la partition de X $\pi_1 = \{\{a_1, a_2\}, \{b_1, b_2\}\}$

* f_2 la fonction de commutation associée à la partition $\pi_2 = \{\{a_1, b_2\}, \{a_2, b_1\}\}$

* R le langage rationnel $(a_1b_1a_2b_2)^+$

Lemme 7.

Etant donné X, f_1, f_2, R *définis comme ci-dessus, il n'existe pas de langage rationnel* $K \subseteq X^*$ *et de fonction de commutation partitionnée* g *définie sur* X *tels que* $g(K) = f_2 \circ f_1(R)$.

Preuve.

Nous allons d'abord montrer que $f_2 \circ f_1(R) \neq \mathrm{Com}(R)$, où Com(R) désigne la commutation totale de R. Pour cela, nous allons prouver que le mot $b_2b_2a_1a_2a_1a_2b_1b_1 = w'$ n'est pas dans $f_2 \circ f_1(R)$.

Pour obtenir le mot w', on ne peut partir que du mot $w = a_1b_1a_2b_2a_1b_1a_2b_2$

Supposons qu'il existe une dérivation $w \xrightarrow[c_{\pi_1}]{*} w_1 \xrightarrow[c_{\pi_2}]{*} w'$. Puisque par f_2, (a_1,b_2) ne commutent pas, il faut que $\mathrm{proj}(w_1, \{a_1,b_2\}) = \mathrm{proj}(w', \{a_1,b_2\}) = b_2b_2a_1a_1$. Mais par f_1, (b_1, b_2) ne commutent pas. Donc $\mathrm{proj}(w_1, \{b_1, b_2\}) = \mathrm{proj}(w, \{b_1,b_2\}) = b_1b_2b_1b_2$. On en déduit donc que $\mathrm{proj}(w_1, \{a_1,b_1,b_2\}) = b_1b_2b_1b_2a_1a_1$. Et comme (a_1, a_2) ne commutent pas par f_1, $\mathrm{proj}(w_1, \{a_1,a_2\}) = a_1a_2a_1a_2$ et ainsi $w_1 = b_1b_2b_1b_2a_1a_2a_1a_2$. Donc $\mathrm{proj}(w_1, \{a_2,b_1\}) = b_1b_1a_2a_2$ et comme (a_2,b_1) ne commutent pas par f_2, on doit avoir $\mathrm{proj}(w_1, \{a_2,b_1\}) = \mathrm{proj}(w', \{a_2, b_1\})$. Or $\mathrm{proj}(w', \{a_2,b_1\}) = a_2a_2b_1b_1$.
On a donc une contradiction.

Supposons maintenant qu'il existe un langage rationnel K et une fonction de commutation partitionnée g définie sur X vérifiant $g(K) = f_2 \circ f_1(R)$. Alors :

$$f_1(R) = \{(a_1a_2)^m \sqcup\!\sqcup (b_1b_2)^m,\ m > o\} \subseteq g(K)$$

donc $proj(f_1(R), \{a_1,b_1\}) = D_1^+ \subseteq proj(g(K), \{a_1,b_1\}) \subseteq D_1^+$. Si (a_1,b_1) ne commutent pas par g, on doit alors avoir $proj(K, \{a_1, b_1\}) = D_1^+$: on a l'égalité d'un langage rationnel et d'un langage non rationnel, ce qui est évidemment impossible. Donc (a_1, b_1) commutent par g. De la même manière on montre alors que (a_1,b_1), (b_1,a_2), (b_2,a_2) commutent par g.

On tient le même raisonnement pour montrer que g fait commuter (a_1,a_2) et (b_1,b_2) en remarquant que

$$f_2(R) = \{(a_1,b_2)^n \sqcup\!\sqcup (a_2b_1)^n,\ n > o\} \subseteq g(K).$$

Donc si g existe, g est associée à la commutation totale, ce qui est en contradiction avec la première partie de la démonstration.

□

<u>*Lemme 8.*</u>

Etant donné X, f_1,f_2, R *définis plus haut, si* $f_2 \circ f_1(R) = g_p \circ \ldots \circ g_1(K)$ *où* K *est un langage rationnel et où les* g_i $(i \in \{1,\ldots,p\})$ *sont des fonctions de commutation partitionnée vérifiant* $g_i \neq$ Identité *et* $g_i \neq g_{i+1}$ *pour* $i \in \{1,\ldots,p-1\}$, *alors* $p=2$, $g_1 = f_1$ *et* $g_2 = f_2$.

<u>Preuve</u>.

On élimine immédiatement le cas p=1 (voir lemme 7). De même on peut écarterle cas où g_i = com (la commutation totale), puisque $f_2 \circ f_1(R) \neq Com(R)$ (lemme 7).

Sur l'alphabet {x, y, z, t}, on peut avoir des partitions de la forme {{x}, {y}, {z,t}} ou {{x,y}, {z,t}} ou{{x}, {y,z,t}}

1°) Dans une première partie, on a écarter le cas où l'une des g_i est associ à une partition de la forme $\{\{x\}, \{y\}, \{z,t\}\}$.
Supposons qu'il existe $i \in \{1,\dots,p\}$ tel que g_i soit associée à $\pi_i = \{\{x\}, \{y\}, \{z,t\}\}$ avec $\{x, y, z, t\} = \{a_1, a_2, b_1, b_2\}$.

Si aucune des autres fonctions ne fait commuter (z,t), alors $\forall j \neq i$, $g_j \subseteq g_i$, donc $g_p \circ \dots \circ g_1 = g_i$, ce qui est impossible (lemme 7).

Si une fonction fait commuter (z,t) : soit g_j une de ces fonctions, choisissons la plus proche au sens des indices, et regardons tous les cas possibles :

- g_j est associée à $\pi_j = \{\{x,z\}, \{y,t\}\}$ avec $\{x,y,z,t\} = \{a_1,b_1,a_2$

Alors $g_p \circ \dots \circ g_i \circ \dots \circ g_j \circ g_1 = g_p \circ \dots \circ g_i \circ g_j \circ \dots \circ g_1$ car $\forall\ i \leq \ell < j,\ g_\ell \subseteq g_i$

$= g_p \circ \dots \circ g_i \circ k \circ g_j \circ \dots \circ g_1$ où k est une fonction associée à la partition $\{\{x\}, \{y,z,t\}\}$; $(k \subseteq g_i)$

$= g_p \circ \dots \circ g_i \circ \ell \circ \dots \circ g_1$ où ℓ est associée à la partition $\{\{x\}, \{z\}, \{y,t\}\}$ ($k \circ g_j = \ell$: lemme 6)

$=$ com ($g_i \circ \ell =$ com, où com est la commutation totale : lemme 2).

Ce qui est impossible.
Si $j < i$, on écrit la même chose.

- Si g_j est associée à une partition du type $\{\{x,y\}, \{z\}, \{t\}\}$

Alors $g_i \circ g_j = g_j \circ g_i =$ Com d'après le lemme 5.

- Si g_j est associée à une partition du type $\{\{x,y,z\}, \{t\}\}$

Alors $g_i \circ g_j = g_i \circ k \circ g_j = g_i \circ \ell =$ com $= g_j \circ g_i$ en appliquant le lemme 6.
On a le même résultat si g_j est associée à la partition $\{\{x,y,t\}, \{z\}\}$.
Dansla suite de la démonstration, on pourra donc conclure à une impossibilité dès qu'on pourra réduire la composée de plusieurs fonctions à une fonction du type de g_i.

2°) Dans la deuxième partie, nous allons regarder la nature de g_1.

* Si g_1 est associée à une partition du type $\{\{x\}, \{y,z,t\}\}$, alors :

- g_2 peut être du même type : g_2 est associée à $\{\{y\}, \{x,z,t\}\}$

par exemple. Mais alors $g_2 \circ g_1$ est associée à $\{\{x\}, \{y\}, \{z,t\}\}$ (lemme 6), ce qui est en contradiction avec la première partie.

- g_2 est alors associée à une partition du type $\{\{x,y\}, \{z,t\}\}$. Alors $g_2 \circ g_1$ est associée encore à $\{\{x\}, \{y\}, \{z,t\}\}$ (lemme 6) et on a la même contradiction.

* Si g_1 est associée à une partition du type $\{\{x,y\}, \{z,t\}\}$

- si g_2 est associée à une partition du type $\{\{x\}, \{y,z,t\}\}$ alors $g_2 \circ g_1$ correspond à la partition $\{\{x\}, \{y\}, \{z,t\}\}$ (lemme 6) ce-qui est impossible.

On peut remarquer que la succession de ces deux types de commutation (g_1 et g_2) se traduit par un cas impossible. On peut donc conclure que tous les g_i sont associées à des partitions du type $\{\{x,y\}, \{z,t\}\}$. Il y a trois possibilités :

- f_1 associée à $\{\{a_1,a_2\}, \{b_1,b_2\}\} = \pi_1$

- f_2 associée à $\{\{a_1,b_2\}, \{a_2,b_1\}\} = \pi_2$

- f_3 associée à $\{\{a_1,b_1\}, \{a_2 b_2\}\} = \pi_3$

Dans la troisième partie de la démonstration, nous allons envisager tous les cas possibles de composition de ces trois fonctions.

<u>1ère remarque</u>.

$f_1 + f_2 = f_1 + f_3 = f_2 + f_3 =$ Com (la commutation totale). Rappelons que si f_i est associée à $\{X_i, \bar{X}_i\}$ et f_j à $\{X_j, \bar{X}_j\}$, $f_i + f_j$ est la fonction associée à $\{X_i \cap X_j, X_i \cap \bar{X}_j, \bar{X}_i \cap X_j, \bar{X}_i \cap \bar{X}_j\}$.

<u>2ème remarque</u>.

si $p \geq 4$

$f_2 \circ f_1(R) = g_p \circ \ldots \circ g_1(R) = g_p \circ \ldots \circ f_\alpha \circ f_k \circ f_j \circ f_i(K)$ avec $\alpha, i, j, k \in \{1,2,3\}$ et $\alpha \neq k$. Donc $\alpha=i$ ou $\alpha=j$; alors

$$f_i \circ f_k \circ f_j \circ f_i = (f_k + f_i) \circ (f_j + f_i) = \text{Com si } \alpha=i \text{ (lemme 3)}$$

et

$$f_j \circ f_k \circ f_j \circ f_i = (f_k + f_j) \circ f_i = \text{Com si } \alpha=j \text{ (lemme 3).}$$

On peut donc éliminer le cas où $p \geq 4$.

Etudions le cas où $p = 2$:

* $f_2 \circ f_1(R) = f_1 \circ f_2(K)$

$\Rightarrow \forall w \in f_2 \circ f_1(R) = f_1 \circ f_2(K),\ f_2(w) \in f_2 \circ f_1(R) = f_1 \circ f_2(K)$

$\Rightarrow f_2 \circ f_1 \circ f_2(K) = f_1 \circ f_2(K) = (f_1 + f_2)(K) = \mathrm{Com}(K)$,

ce qui est impossible (lemme 3, lemme 7).

* $f_2 \circ f_1(R) = f_3 \circ f_1(K)$

$= f_2 \circ f_3 \circ f_1(K)$ (même raisonnement)

$= f_3 \circ f_2 \circ f_3 \circ f_1(K)$ (même raisonnement)

$= (f_2 + f_3) \circ f_1(K) = \mathrm{Com}(K)$ (lemme 3).

Donc même impossibilité.

* $f_2 \circ f_1(R) = f_1 \circ f_3(K)$

$= f_2 \circ f_1 \circ f_3(K)$

$= f_1 \circ f_2 \circ f_1 \circ f_3(K) = (f_1 + f_2) \circ f_3(K) = \mathrm{Com}(K)$: impossible

* $f_2 \circ f_1(R) = f_3 \circ f_2(K)$

$= f_2 \circ f_3 \circ f_2(K) = (f_2 + f_3)(K) = \mathrm{Com}(K)$: impossible.

* $f_2 \circ f_1(R) = f_2 \circ f_3(K)$.

Le principe ici doit être différent. On sait que le mot $w' = b_2b_2a_1a_2a_1a_2b_1b_1$ n'appartient pas à $f_2 \circ f_1(R)$ (lemme 7). Montrons que ce mot appartient à $f_2 \circ f_3(K)$. Soit $w \in K$ tel que $|w|_{a_1} = |w|_{a_1} = |w|_{b_2} = 2$.

Alors $w \xrightarrow[c_{\pi_3}]{*} \text{proj}(w, \{a_2, b_2\})\ \text{proj}(w, \{a_1, b_1\})$ et

$\text{proj}(w, \{a_2, b_2\})\ \text{proj}(w, \{a_1, b_1\}) \xrightarrow[c_{\pi_2}]{*} b_2b_2a_2a_2a_1a_1b_1b_1$ car (a_2,b_2) et (a_1,b_1) commutent par f_2 puis $b_2b_2a_2a_2a_1a_1b_1b_1 \xrightarrow[c_{\pi_2}]{*} b_2b_2a_1a_2a_1a_2b_1b_1$ car (a_1, a_2) commutent dans f_2.

Donc $w' \in f_2 \circ f_3(K)$.

Il nous reste à regarder le cas où $p = 3$:

* $f_2 \circ f_1(R) = f_i \circ f_j \circ f_i(K)$ avec $i, j \in \{1,2,3\}$, $i \neq j$ alors $f_2 \circ f_1(R) = (f_i + f_j)(K) = \text{Com}(K)$: impossible.

* $f_2 \circ f_1(R) = f_k \circ f_j \circ f_i(K)$ avec $i, j, k \in \{1,2,3\}$ $i \neq j \neq k$.

Alors, on aura forcément une inclusion $f_\alpha \circ f_\beta(K) \subseteq f_2 \circ f_1(R)$ qui est impossible puisque dans tous les cas étudiés précédemment, il y avait des incompatibilités.

On peut énoncer ce lemme sous une autre forme : □

Lemme 8 bis.

f_1, f_2, R étant définis comme précédemment, si $f_2 \circ f_1(R) = g_p \circ \ldots \circ g_1(K)$, avec $p > 1$, $g_i \in P$ $\forall i \in \{1,\ldots,p\}$ et $K \in \text{RAT}$, alors il existe k vérifiant $1 \leq k < p$ tel que :

- $\forall i < k$, $g_i = f_1$ *ou Identité, et* $g_i = f_1$ *pour au moins* 1 i
- $\forall j \geq k$, $g_j = f_2$ *ou Identité et* $g_j = f_2$ *pour au moins* 1 j.

Proposition 3.

La hiérarchie des familles $P^n(\text{RAT})$ pour $n \geq 0$ est infinie. C'est à dire $\forall n \geq 1$, $P^{n-1}(\text{RAT}) \subsetneq P^n(\text{RAT})$.

Preuve.

Le cas n=2 a été démontré dans le cadre du lemme 7. Le cas n=1 est trivial ($Com((ab)^*) \in P(RAT)$ et $\notin RAT$). On supposera donc $n > 2$. Posons $X = \{a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_m\}$ et définissons

n partitions de X : $\pi_1, \pi_2, \ldots, \pi_n$

par

$$\pi_1 = \{\{a_1, a_2, \ldots, a_n\}, \{b_1, b_2, \ldots, b_n\}\}$$

et

$$\pi_i = \{\{a_1, b_2, b_3, \ldots, b_i, a_{i+1}, \ldots, a_n\}, \{b_1, a_2, \ldots, a_i, b_{i+1}, \ldots, b_n\}\}$$

$\forall i > 1$.

A chaque partition π_i, associons la fonction f_i ($i \in \{1, \ldots, n\}$)
Soit $R = (a_1b_1a_2b_2\ldots a_nb_n)^+$ et posons $L = f_n \circ \ldots \circ f_1(R)$. $L \in P^n(RAT)$.
Montrons que $L \notin P^{n-1}(RAT)$.
Mais d'abord, nous pouvons faire 2 remarques importantes :

Remarque 1.

$\forall i \in \{2, \ldots, n\}$

$$proj(L, \{a_1, b_1, a_i, b_i\}) = f_i\Big|_{\substack{a_1,b_1\\a_i,b_i}} \circ f_1\Big|_{\substack{a_1,b_1\\a_i,b_i}} (proj(R, \{a_1, b_1, a_1, b_1\}))$$

$$= t_i \circ p_i((a_1b_1a_ib_i)^+)$$

en notant p_i la fonction associée à la partition $\{\{a_1, a_i\}, \{b_1, b_i\}\}$

et t_i la fonction associée à la partition $\{\{a_1, b_i\}, \{b_1, a_i\}\}$.

Nous savons (lemme 8) que ce sont les seules fonctions possibles pour obtenir le langage $proj(L, \{a_1, b_1, a_i, b_i\})$ à partir d'un rationnel.

Remarque 2.

$\forall i,j \in \{2,\dots,n\},\ i \neq j,\ i < j,$

$$\mathrm{proj}(L, \{a_i,b_i,a_j,b_j\}) = f_j \Big|_{\substack{a_i,b_i \\ a_j,b_j}} \circ f_i \Big|_{\substack{a_i,b_i \\ a_j,b_j}} \circ f_1 \Big|_{\substack{a_i,b_i \\ a_j,b_j}} ((a_i b_i a_j b_j)^+)$$

$$= p_{ij} \circ t_{ij} \circ p_{ij} ((a_i b_i\ a_j b_j)^+)$$

où p_{ij} est la fonction associée à la partition $\{\{a_i,a_j\}, \{b_i,b_j\}\}$

et t_{ij} est la fonction associée à la partition $\{\{a_i,b_j\}, \{a_j,b_i\}\}$.

$$= \mathrm{Com}[(a_i b_i a_j b_j)^+].$$

Supposons maintenant qu'il existe n-1 fonctions de commutation $g_1, g_2, \dots, g_{n-1}$ et un langage rationnel K tels que $L = g_{n-1} \circ \dots \circ g_1(K)$. On peut supposer $g_i \neq$ Identité, $\forall i \in \{1,\dots,n-1\}$. On doit alors avoir :

$\forall i \in \{2,\dots,n\},$

$$\mathrm{proj}(L, \{a_1,b_1,a_i,b_i\}) = g_{n-1} \Big|_{\substack{a_1,b_1 \\ a_i,b_i}} \cdots\ g_1 \Big|_{\substack{a_1,b_1 \\ a_i,b_i}} (\mathrm{proj}(K, \{a_1,b_1,a_i,b_i\})$$

$$= t_i \circ p_i((a_1 b_1 a_i b_i)^+) \qquad \text{(\underline{rem. 1})}$$

Donc, d'après le lemme 8 bis, $\forall i \in \{2,\dots,n\}$, on peut trouver $k_i \in \{2,\dots,n-1\}$ tel que :

$$\text{et} \quad \begin{cases} g_j \Big|_{\substack{a_1,b_1 \\ a_i,b_i}} = t_i \text{ ou Identité}, \quad \forall j < k_i \\[2ex] g_j \Big|_{\substack{a_1,b_1 \\ a_i,b_i}} = p_i \text{ ou Identité}, \quad \forall j \geq k_i \end{cases}$$

Si, pour un i fixé, il existe j tel que $g_j \Big|_{\substack{a_1,b_1\\ a_i,a_i}}$ = Identité alors, g_j sera associée à une partition de la forme :

$$\{\{a_1, b_1, a_i, b_i, \ldots\}, \{\ldots\}, \ldots\}$$

donc a_1 et b_1 sont dans le même élément de la partition.

Comme

$$\forall k \in \{2,\ldots n\},\quad g_j \Big|_{\substack{a_1,b_1\\ a_k,b_k}} = p_k \text{ ou } t_k \text{ ou Identité},$$

on aura forcément $g_j \Big|_{\substack{a_1,b_1\\ a_k,b_k}}$ = Identité car p_k et t_k imposent que a_1 et b_1 soient dans des éléments différents de la partition.

Donc g_j = Identité. Or nous avons supposé que toutes les fonctions g_j étaient différentes de l'identité.

Nous pouvons donc affirmer que :

$\forall i \in \{2,\ldots,n\}$, il existe $k_i \in \{2,\ldots,n-1\}$ tel que :

$$\begin{cases} g_1 \Big|_{\substack{a_1,b_1\\ a_i,b_i}} = \ldots = g_{k_i-1} \Big|_{\substack{a_1,b_1\\ a_i,b_i}} = t_i = f_1 \Big|_{\substack{a_1,b_1\\ a_i,b_i}} \\[2em] g_{k_i} \Big|_{\substack{a_1,b_1\\ a_i,b_i}} = \ldots = g_{n-1} \Big|_{\substack{a_1,b_1\\ a_i,b_i}} = p_i = f_i \Big|_{\substack{a_1,b_1\\ a_i,b_i}} \end{cases}$$

Donc, il existe 2 indices $i, j \in \{2,\ldots,n\}$, $i \neq j$ tels que $k_i = k_j$.

On a donc la configuration suivante :

g_1 est associée à la partition $\{\{a_1, a_i, a_j, \ldots\}, \{b_1, b_i, b_j, \ldots\}\}$

g_{k_i-1} est associée à la partition $\{\{a_1, a_i, a_j, \ldots\}, \{b_1, b_i, b_j, \ldots\}\}$

g_{k_i} est associée à la partition $\{\{a_1, b_i, b_j, \ldots\}, \{b_1, a_i, a_j, \ldots\}\}$

$\vdots$

g_{n-1} est associée à la partition $\{\{a_1, b_i, b_j, \ldots\}, \{b_1, a_i, a_j, \ldots\}\}$

Alors proj(L, $\{a_i,b_i,a_j,b_j\}$) = Com($(a_ib_ia_jb_j)^+$) d'après la remarque 2

$$= g_{1\,\Big|\substack{a_i,b_i\\ b_j,b_j}} \cdots\ g_{k_i\,\Big|\substack{a_i,b_i\\ a_j,b_j}} \cdots\ g_{n-1\,\Big|\substack{a_i,b_i\\ a_j,b_j}} (\mathrm{proj}(K, \{a_i,b_i,a_j,b_j\}))$$

$$= g_{1\,\Big|\substack{a_i,b_i\\ a_j,b_j}} (\mathrm{proj}\ (K, \{a_i,b_i,a_j,b_j\}))$$

où $g_{1\,\Big|\substack{a_i,b_i\\ a_j,b_j}}$ est associée à la partition $\{\{a_i,a_j\}, \{b_i,b_j\}\}$.

Donc Proj(K, $\{a_i,a_j\}$) = proj(com($(a_ib_ia_jb_j)^+$), $\{a_i,a_j\}$) = Com ($(a_ia_j)^+$) : on a ici l'égalité d'un langage rationnel et d'un langage non rationnel, ce qui nous donne la contradiction.

□

D - COMPARAISONS DES FAMILLES P^k(RAT), CP^k(RAT), SC^k(RAT).

Comme les fonctions de commutation partitionnée forment un sous-ensemble des fonctions de commutation partielle qui sont elles même des cas particuliers de fonctions de semi-commutation, nous avons clairement les inclusions $P^k(RAT) \subseteq CP^k(RAT) \subseteq SC^k(RAT)$, pour $k \geq 0$.
Nous allons montrer que ces inclusions sont strictes en énonçant un résultat un peu plus fort : une fonction de commutation partielle (resp. semi-commutation) appliquée à un rationnel ne pourra pas toujours se simuler par l'application d'une suite de fonctions de commutation partitionnée (resp. commutation partielle) à un autre langage rationnel : en d'autres termes, $CP(RAT) \not\subseteq P^*(RAT)$ et $SC(RAT) \not\subseteq CP^*(RAT)$.

Proposition 4.

$$CP(RAT) \not\subseteq P^*(RAT)$$

Preuve.

Soient le langage rationnel $R = (ab)^* \#$ et la fonction de commutation partielle f associée à la relation $C = \{ab \leftrightarrow ba\}$. Alors $f(R) = D_1^* \#$ où $D_1^* \#$ est le langage de Dyck sur $\{a,b\}$ ($f(R) \notin RAT$).

Supposons qu'il existe $k \geq 1$ tel que $f(R) \in P^k(RAT)$. Alors il existe un rationnel K et k fonctions de commutation partitionnée f_i ($i \in \{1,\ldots k\}$) vérifiant $f(R) = f_k \circ \ldots \circ f_1(K)$. On peut supposer que $f_k \neq$ Identité.

Alors f_k est associée à une partition π_k de $\{a,b,\#\}$, donc # permute au moins avec les lettres a ou b.

De plus, $\forall w \in f(R) = f_k \circ \ldots \circ f_1(K)$, $f_k(w) \in f_k \circ \ldots \circ f_1(K) = f(R)$.

Si $(a,\#)$ commutent par f_k, le mot $ba\#$ est dans $f(R)$ donc $f_k(ba\#) \subseteq f(R)$ donc $b\#a \subseteq f(R)$, ce qui est impossible.

De même, si $(b,\#)$ commutent par f_k, le mot $ab\#$ appartenant à $f(R) = f_k \circ f(R)$, on en déduit que $a\#b$ doit appartenir à $f(R)$, ce qui est faux. Donc $f(R) \notin P^k(RAT)$, $\forall k \geq 0$.

□

Corollaire.

$$P^k(RAT) \subsetneq CP^k(RAT), \ \forall k \geq 1$$

Preuve.

On sait que

$$CP(RAT) \neq P^k(RAT) \quad \forall k \geq 0$$

$$CP(RAT) \subsetneq CP^k(RAT) \quad \forall k \geq 1$$

$$P^k(RAT) \subseteq CP^k(RAT), \ \forall k \geq 1$$

La conclusion est immédiate.

□

Pour comparer les familles $CP^k(RAT)$ et $SC^k(RAT)$, le raisonnement est tout à fait analogue et l'exemple est le même.

Proposition 5.

$$SC(RAT) \not\subseteq CP^*(RAT)$$

Preuve.

Posons $R = (ab)^* \#$ et f la fonction de semi-commutation associée à la relation $C = \{ba \to ab\}$. Alors $f(R) = D_1'^* \#$ où $D_1'^*$ est le langage de semi-Dyck sur $\{a,b\}$. (Rem : $f(R) \notin RAT$).

Supposons qu'il existe $k \geq 1$ et k fonctions de commutation partielle f_i, $i \in \{1,\dots,k\}$ et un langage rationnel K vérifiant $f_k \circ \dots \circ f_1(K) = f(R)$
Comme $proj(f(R), \{a,b\}^*) = D_1'^* \notin RAT$, on a forcément une des fonctions f_i qui fait commuter a et b.
Soit $w \in K$ tel que $|w|_a = |w|_b = 1$. Comme $K \subseteq f(R)$, on a $w = ab \#$; alors $f_i(w) \subseteq f(R)$ et $f_i(w)$ contient $ba \#$ ce qui est contradictoire.

Le corollaire suivant est immédiat :

Corollaire.

$$\forall k \geq 1,\ CP^k(RAT) \subsetneq SC^k(RAT).$$

E - PROBLEMES DE DECIDABILITE.

Nous allons donner quelques propriétés de décidabilité ou d'indécidabilité concernant pour la plupart les langages rationnels et les langages de P(RAT).

Proposition 6.

Il est indécidable de déterminer si un langage rationnel et un langage de P(RAT) sont égaux.

Preuve.

Nous allons exhiber une famille de langages rationnels $\mathcal{L}$ et une fonction de commutation partitionnée f tels que l'égalité $f(R) = X^*$, pour $R \in \mathcal{L}$, est indécidable, en nous ramenant au théorème de Post.

1ère partie.

Soient X un alphabet et $\pi = \{X_1, X_2\}$ une partition de X. Notons π_1 la projection de X sur X_1 et π_2 la projection de X sur X_2.

Soient h et g deux homomorphismes non effaçants définis sur X_1^* tels que $\forall x \in X_1$, $h(x) \in X_2^+$ et $g(x) \in X_2^+$. Posons :

$$L = \{\varepsilon\} \cup \{w \in X^* \ / \ h(\pi_1(w)) \neq \pi_2(w)\}$$

$$\cup \ \{w \in X^* \ / \ g(\pi_1(w)) \neq \pi_2(w)\}$$

Nous allons montrer que $L \neq X^* \iff \exists u \in X_1^+ \ / \ h(u) = g(u)$.

$\Longrightarrow$ Si $L \neq X^*$, il existe $u \in X^+$ tel que $u \notin L$.

$$u \notin L \Longrightarrow h(\pi_1(u)) = \pi_2(u) \text{ et } g(\pi_1(u)) = \pi_2(u).$$

Soit $w = \pi_1(u)$ ($w \neq \varepsilon$ car $u \neq \varepsilon$). Il est clair que $h(w) = g(w)$.

$\Longleftarrow$ Soit $w = uh(u)$ $w \neq \varepsilon$ car $u \neq \varepsilon$. Alors

$$h(\pi_1(w)) = h(u) = \pi_2(w)$$

et

$$g(\pi_1(w)) = g(u) = \pi_2(w)$$

donc

$$w \notin L \text{ et } L \neq X^*$$

2ème partie.

Montrons que $L \in P_2(RAT)$.

Si $X = \{a_1, a_2, \ldots, a_n\}$, posons

$$R_1 = \left(\sum_{i=1}^{n} a_i\, h(a_i)\right)^* \left(\sum_{i=1}^{n} a_i (X_2^* \setminus h(ai))\right) \left(\sum_{i=1}^{n} a_i\, X_2^*\right)^*,$$

$$R_2 = \left(\sum_{i=1}^{n} a_i\, g(a_i)\right)^* \left(\sum_{i=1}^{n} a_i (X_2^* \setminus g(a_i))\right) \left(\sum_{i=1}^{n} a_i\, X_2^*\right)^*$$

et

$$R = \{\varepsilon\} \cup R_1 \cup R_2.$$

Si f désigne la fonction de commutation associée à la partition de X $\{X_1, X_2\}$, il est clair que :

$$f(R_1) = \{w \in X^* \ / \ h(\pi_1(w)) \neq \pi_2(w)\}$$

et

$$f(R_2) = \{w \in X^* \ / \ g(\pi_1(w)) \neq \pi_2(w)\}$$

Donc

$$L = f(R).$$

D'après la première partie, nous pouvons alors écrire :

$$f(R) \neq X^* \iff \exists u \in X_1^+ \ / \ h(u) = g(u),$$

et cette propriété est indécidable d'après le théorème de Post.

□

<u>*Corollaire.*</u>

La propriété $K \subseteq f(R)$, *où* R *et* K *sont des langages rationnels et* f *une fonction de commutation partitionnée, est indécidable.*

<u>Preuve.</u>

Nous venons de voir qu'il est indécidable de déterminer si $X^* \subseteq f(R)$ (puisque $f(R) \subseteq X^*$ est évident).

□

Proposition 7.

Etand donné un langage rationnel $R \subseteq X^*$ *et une fonction de commutation partitionnée* f *définie sur* X, *on peut décider si* $f(R) = R$.

Preuve.

La relation de commutation C associée à f peut être considérée comme un ensemble de règles de réécriture que nous noterons P

$$P = \{xy \to yx,\ yx \to xy \ /\ (x,y) \in C\}$$

D'autre part, il existe un automate déterministe minimal

$$M = (X, Q, *, q_o, F)$$

tel que $T(M) = R$.
Nous allons définir une relation sur l'ensemble des états Q de l'automate comme suit :

$$\forall q, q' \in Q,\ q \leq q' \iff \forall w \in X\ ,\ q*w \in F \implies q'*w \in F$$

Montrons maintenant que :

$$f(R) = R \iff \forall q \in Q,\ \forall u \to v \in P,\ q*u \leq q*v$$

$\implies$ $\forall q \in Q,\ \exists w_1 \in X^*\ /\ q_o*w_1 = q$

Soit $q_1 = q*u$.
Alors $\forall w_2 \in X^*$ tel que $q_1*w_2 \in F$, $w_1uw_2 \in R$ donc $w_1vw_2 \in R$ et $q*vw_2 \in F$ c'est à dire $(q*v)\ w_2 \in F$; ainsi $q_1 = q*u \leq q*v$.

$\Longleftarrow$ Montrons que $\forall w' \in f(R)$, $w' \in R$.
$\forall w' \in f(R)$, $\exists w \in R$ tel que $w' \in f(w)$; il existe donc une dérivation qui fait passer de w à w'. Nous allons raisonner par récurrence sur la longueur de cette dérivation.
Si $\ell = 1$: $w = w_1uw_2$, $w' = w_1vw_2$ et $u \to v \in P$ ($w_1,w_2 \in X^*$).

$$w_1uw_2 \in R \implies q_o*w_1uw_2 \in F$$

Posons $q_1 = q_o * w_1$. Alors, $(q_1 * u) * w_2 \in F$. Puisque $q_1 * u \leq q_1 * v$, on a $q_1 * v * w_2 \in F$, donc $w_1 v w_2 \in R$. Si la longueur de la dérivation est $\ell+1$:

$$w \underset{c}{\to} w_1 \overset{\ell}{\underset{c}{\to}} w'.$$

Par hypothèse de récurrence, $w \underset{c}{\to} w_1$ implique $w_1 \in R$ et $w_1 \overset{\ell}{\underset{c}{\to}} w'$ entraîne $w' \in R$. Donc $f(R) = R$. Il nous reste à montrer maintenant que $\forall q, q' \in Q$, la propriété $q \leq q'$ est décidable : posons $Lq = \{w \in X^* / q * w \in F\}$.

Lq est un langage rationnel, et $q \leq q'$ est équivalent à $Lq \subseteq Lq'$, et cette propriété est décidable.

□

Nous allons montrer dans la proposition suivante que ce résultat est faux pour les langages algébriques, par l'intermédiaire de deux lemmes :

Lemme 9.

Soit $X = \{x_1, x_2, \ldots, x_k\}$. *On peut déterminer si un langage algébrique* $L \subseteq X^*$ *contient* $x_1^* x_2^* \ldots x_k^*$.

Preuve.

On construit $L' = L \cap x_1^* x_2^* \ldots x_k^*$ et $\psi(L')$ où ψ désigne la fonction de Parikh. Puis il suffit de regarder si $\psi(L') = \mathbb{N}^k$, ce qui est décidable.

□

Lemme 10.

Il est indécidable de déterminer si un langage algébrique (ou linéaire) est commutatif.

Preuve.

Soit $X = \{x_1, x_2, \ldots, x_k\}$ et $L \subseteq X^*$ un langage algébrique. Alors il est clair que

$$L = X^* \iff \left| \begin{array}{l} 1 \quad x_1^* x_2^* \ldots x_k^* \subseteq L \\ \\ 2 \quad L \text{ est commutatif.} \end{array} \right.$$

Comme la condition 1 est décidable (lemme précédent), $L = X^*$ étant une propriété indécidable, la condition 2 ne peut être décidable.

□

Nous pouvons alors énoncer le résultat suivant :

<u>*Proposition 8*</u>.

La propriété $L = f(L)$ *où* L *est un langage algébrique et* f *une fonction de commutation partitionnée est indécidable.*

<u>Preuve</u>.

Il suffit de choisir pour f la fonction qui réalise la commutation totale.

□

<u>*Proposition 9*</u>.

Il est indécidable de déterminer si l'intersection d'un langage rationnel et d'un langage de P(RAT) *est vide.*

<u>Preuve</u>.

Soient $X = X_1 \cup X_2$ un alphabet, et h et g deux homomorphismes non effaçants de X_1^* dans X_2^*.
Posons $K = (\sum_{i=1}^{n} x_i\, h(x_i))^+$ et $R = (\sum_{i=1}^{n} x_i\, g(x_i))^+$ avec $X = \{x_1, x_2, \ldots, x_n\}$.

Si f désigne la fonction associée à la partition $\{X_1, X_2\}$, on peut facilement montrer que :

$$f(R) \cap K \neq \emptyset \iff \exists u \in X_1^+ \;/\; h(u) = g(u)$$

et cette propriété est indécidable (problème de Post).

□

En utilisant le même raisonnement et le même exemple, on a le résultat suivant :

Proposition 10.

Il est indécidable de déterminer si l'intersection de deux langages de P(RAT) est vide.

Ceci ne représente certainement qu'une petite partie des questions qu'on pourrait se poser.

Citons quelques exemples de problème qui pourront peut-être être résolus.

Si $\{g_k, 1 \leq k \leq n\}$ est un ensemble de n fonctions de commutation peut-on décider si $g_n \circ \ldots \circ g_1 = Com$ (la commutation totale) ? Plus généralement peut on décider de l'égalité de deux fonctions de P^*, ou de l'égalité de deux fonctions de SC^* ?

Quelle est la plus courte suite de fonctions de commutation (2) partitionnée qui nous donne la commutation totale quand on les compose ?

Une méthode de construction d'une telle suite est la suivante. On divise systématiquement l'alphabet en 2 sous-ensembles de même taille (à 1 élément près, si la taille de l'alphabet est impaire).

Exemple :

Soit $X = \{a_1, a_2, a_3, a_4, a_5, a_6\}$.

Alors définissons les partitions suivantes :

$\pi_1 = \{a_1, a_2, a_3\}, \{a_4, a_5, a_6\}$ sera associée à la fonction f_1

$\pi_2 = \{a_1, a_5, a_6\}, \{a_2, a_3, a_4\}$ sera associée à la fonction f_2

$\pi_3 = \{a_1, a_2, a_5\}, \{a_3, a_4, a_6\}$ sera associée à la fonction f_3.

Alors $\forall w \in X^*$

$$w \xrightarrow[c_{\pi_1}]{*} proj(w, \{a_1,a_2,a_3\})\ proj(w, \{a_4,a_5,a_6\})$$

$\xrightarrow[c_{\pi_2}]{*}$ proj(w,{a_1}) proj(w,{a_2,a_3}) proj(w,{a_4}) proj(w,{a_5,a_6})

$\xrightarrow[c_{\pi_3}]{*}$ w_1 où w_1 $\in$ proj(w,{a_1})com[proj(w,{a_2,a_3})] proj(w,{a_4}) com[proj(w,{a_5,a_6

$\xrightarrow[c_{\pi_2}]{*}$ w_2 où w_2 $\in$ Com[proj(w, {a_1,a_2,a_3})] Com [proj(w, {a_4,a_5,a_6})]

$\xrightarrow[c_{\pi_1}]{*}$ w' où w' $\in$ Com(w). Donc Com = $f_1 \circ f_2 \circ f_3 \circ f_2 \circ f_1$.

Si la taille de l'alphabet est n, le nombre de fonctions composées est de l'ordre de $2\log_2 n$. Peut-on faire mieux ?

[1] J. BERSTEL, "*Transductions and context-free languages*", Teubner Verlag, Stuttgart, 1979.

[2] A. BERTONI, G. MAURI et N. SABADINI, "*A hierarchy of regular trace languages and some combinatorial applications*", 2nd World Conference on Mathematics at the Service of Men, Las Palmas, 1982.

[3] A. BERTONI, G. MAURI et N. SABADINI, "*Equivalence and Membership problems for regular trace languages*", 9ème ICALP, Aarhus, Lecture Notes in Computer Science 140 (1982), 61-71.

[4] R. BOOK et S. GREIBACH, "*Quasi-realtime languages*", Math. Systems theory 4 (1970), 97-111.

[5] R. BOOK, S. GREIBACH et C. WRATHALL, "*Reset-Machines*", J. of Computer and System Sciences 19 (1979), 256-276.

[6] R. BOOK, M. NIVAT et M. PATERSON, "*Reversal-bounded acceptors and intersections of linear languages*", SIAM J. Comp. 3 (1974), 283-295.

[7] P. CARTIER et D. FOATA, "*Problèmes combinatoires de commutation et réarrangements*", Lecture Notes in mathematics 85, Springer-Verlag, Berlin 1969.

[8] R. CORI et D. PERRIN, "*Sur la reconnaissabilité dans les monoïdes partiellement commutatifs libres*", Publication Interne du LITP, 1983.

[9] S. GINSBURG, "*Algebraic and Automata theoretic properties of formal languages*", North Holland, Amsterdam, 1975.

[10] J. GRABOWSKI, "*On partial languages*", Fundamenta Informaticae 4 (1981), 427-498.

[11] E. KNUTH, "*Cycles of partial orders*", MFCS, Zakopane, Lecture Notes in Computer Science 64 (1978), 315-325.

[12] G. LALLEMENT, "*Semigroups and Combinatorial applicatons*", J. Wiley and Sons, New-York, 1979.

[13] M. LATTEUX et J. LEGUY, *"On the composition of morphisms and inverse morphisms"*, 10th ICALP, Barcelone, Lecture Notes in Computer Science 154, 420-432.

[14] M. LATTEUX, *"Langages à un compteur"*, J. of Comput. System Sc.", 26 (1983), 14-33.

[15] J. LEGUY, *"Transductions rationnelles décroissantes"*, RAIRO Info. théorique 5 (1981), 141-148.

[16] A. MAZURKIEWICZ, *"Concurrent program schemes and their interpretations"*, DAIMI, PB.78, Aarhus University, 1977.

[17] F. RODRIGUEZ et J.L. SCHMITT, *"Autoshuffle et langages FIFO"*, Publication de l'ENSEEIH, Toulouse, 1982.

[18] M. SZIJARTO, *"Trace languages and closure operations"*, Technical report of dept. of numerical and computer Math., L. Eötvos University, Budapest, 1979.

[19] M. SZIJARTO, *"A classification and closure properties of languages for describing concurrent system behaviours"*, Fundamenta Informaticae 4 (1981), 531-549.

RESUME

Nous définissons des fonctions de commutation sur les mots par la donnée d'une relation binaire quelconque sur un alphabet ; les fonctions de semi-commutation correspondent aux relations non reflexives, les fonctions de commutation partielle sont associées aux relations non réflexives et symétriques, et dans le cas où la relation est le complémentaire d'une relation d'équivalence, nous obtenons les fonctions de commutation partitionnée.

Ce dernier type de fonction a l'avantage de s'exprimer facilement au moyen d'opérations bien connues : les homomorphismes et l'opération de shuffle.

Nous montrons alors qu'on peut factoriser les fonctions de commutation partielle et les fonctions de semi-commutation en une succession d'homomorphismes, d'homomorphismes inverses et de fonctions de commutation partitionnée.

Puis nous étudions les propriétés des cônes rationnels fidèles clos par commutation partielle, ce qui nous permet d'obtenir une nouvelle caractérisation de la famille des langages Multireset qui est le plus petit cône rationnel fidèle clos par commutation partielle.

Nous montrons ensuite que la hiérarchie des familles de langages construite à partir des langages rationnels au moyen des fonctions de semi-commutation (resp. commutation partielle, commutation partitionnée) est une hiérarchie infinie.

Enfin, nous étudions les problèmes de décidabilité concernant principalement P(RAT), la famille des langages images par une fonction de commutation partitionnée d'un langage rationnel.