RAIRO. INFORMATIQUE THÉORIQUE

ANDRÉ ARNOLD

Sémantique des processus communicants

RAIRO. Informatique théorique, tome 15, n° 2 (1981), p. 103-139

<http://www.numdam.org/item?id=ITA_1981__15_2_103_0>

R.A.I.R.O. Informatique théorique/Theoretical Informatics
(vol. 15, n° 2, 1981, p. 103 à 139)

SÉMANTIQUE DES PROCESSUS COMMUNICANTS (*)

par André ARNOLD (1)

Communiqué par M. NIVAT

Résumé. — *Pour étudier le fonctionnement d'un ensemble de processus échangeant constamment des informations, nous considérons un processus comme un transducteur de mots infinis, ce qui permet de définir une sémantique opérationnelle des réseaux de processus qui coïncide avec la sémantique dénotationnelle de G. Kahn.*

Abstract. — *In order to study the behaviour of a set of processes endlessly exchanging informations, we consider a process as a transducer of infinite words, which allows to define operational semantics of a net of processes equivalent to Kahn's denotational semantics.*

INTRODUCTION

Un préliminaire important à l'étude du parallélisme est de modéliser le fonctionnement d'un ensemble de machines qui échangent constamment des informations entre elles.

A la suite de G. Kahn [5] il est admis par les auteurs que ces échanges d'information peuvent être représentés par des mots finis ou infinis et que les machines transforment ces mots en d'autres. Un ensemble de machines peut alors être représenté par un système d'équations sur le cpo des mots finis ou infinis et Kahn a montré que sous des hypothèses parfaitement naturelles ces systèmes d'équations avaient un plus petit point fixe, ce qui lui permet de donner une sémantique dénotationnelle à ces ensembles de machines, ou de processus.

Par contre la sémantique opérationnelle de ces systèmes n'a pas encore été définie de manière satisfaisante. Si on se réfère à la discussion qui suit l'exposé de Keller [6] il semble que le problème essentiel que pose cette définition est que l'interprétation donnée à un mot n'est pas la même du point de vue dénotationnel ou opérationnel. En particulier le mot vide n'est dans le premier

(*) Reçu novembre 1979.
(1) Laboratoire d'Informatique, Université de Poitiers et Laboratoire d'Informatique théorique et Programmation, C.N.R.S.

R.A.I.R.O. Informatique théorique/Theoretical Informatics, 0399-0540/1981/103/$ 5.00

cas rien d'autre que « l'indéfini », le plus petit élément du cpo à partir duquel on construit une suite d'approximations, alors que du point de vue opérationnel il signifie très concrètement l'absence d'information sur une ligne de communication.

En donnant au mot vide cette deuxième signification nous définissons une sémantique opérationnelle où l'absence d'information (i. e. le mot vide) sur une ligne d'entrée d'un processus provoque la mise en attente de ce processus. Dans cette perspective, et du point de vue dénotationnel, les entrées d'un processus doivent être des mots infinis car sinon le processus finirait par rester perpétuellement en attente – situation analogue à celle d'une boucle dans un programme – et ne pourrait donc pas fournir de résultat. Aussi les équations associées à un « réseau » de processus communicants sont-elles des équations sur l'ensemble des mots infinis muni de la structure d'espace métrique [2, 3] et c'est pourquoi nous exigeons des processus qu'ils vérifient une condition analogue à la condition de Greibach pour les schémas de programme [1].

Nous montrons que ces deux sémantiques coïncident à la seule condition que dans la sémantique opérationnelle chaque processus puisse être « déclenché » une infinité de fois. Nous montrons que cette condition, la « condition de déclenchement », ne dépend que de la configuration initiale du réseau. Nous montrons aussi que n'importe quel réseau peut être remplacé par un seul processus.

Tout ceci s'applique aussi bien aux processus déterministes qu'aux non-déterministes. Mais, des caractérisations que nous donnons des fonctions (multivoques) calculables par des processus il apparaît que seul le non-déterminisme que nous appelons « finitaire (à chaque étape du calcul il n'y a qu'un nombre fini de possibilités) présente un intérêt, ce qui confirme les constatations similaires de Plotkin [9] ou Dijkstra [11, p. 181].

Cet article comprend cinq parties. La première comporte quelques brefs rappels sur les mots infinis et leur structure d'espace métrique. Dans la seconde nous définissons les processus et les fonctions qu'ils calculent. Les réseaux et leur sémantique opérationnelle ainsi que la condition de déclenchement sont définis dans la troisième. La quatrième partie est consacrée à la preuve de l'équivalence des sémantiques opérationnelle et dénotationnelle. Dans la cinquième partie nous montrons que tout réseau est équivalent à un seul processus puis nous caractérisons les fonctions calculables par un processus.

1. PRÉLIMINAIRES

Nous rappelons brièvement ici quelques notions sur les mots infinis et sur leur topologie qu'on trouvera plus développées dans [2, 3, 8].

Nous noterons $\mathbb{P}$ l'ensemble des entiers strictement positifs et pour tout $n \in \mathbb{P}$ nous noterons $[n]$ l'ensemble $\{1, \ldots, n\}$. L'ensemble $\mathbb{P}$ lui-même pourra être noté $[\infty]$.

Soit Σ un alphabet, fini ou infini. Nous noterons Σ^* l'ensemble des mots finis et pour tout $u \in \Sigma^*$, $\lg(u)$ sera la longueur du mot u c'est-à-dire le nombre de lettres qu'il contient. Si i est un entier appartenant à $[\lg(u)]$, $u/i/$ sera la i-ième lettre du mot u.

Un mot infini est une application de $\mathbb{P}$ dans Σ, c'est-à-dire une suite infinie de lettres. Nous noterons Σ^ω l'ensemble des mots infinis sur Σ. Par extension de la notation utilisée pour des mots finis nous noterons, pour tout $i \in \mathbb{P}$, $U/i/$ la i-ième lettre du mot infini U; de plus la longueur, $\lg(U)$ du mot U sera égale à ∞.

Soit $\Sigma^\infty = \Sigma^* \cup \Sigma^\omega$ l'ensemble des mots finis ou infinis sur Σ. Le produit de concaténation sur Σ^* est étendu en un produit sur Σ^∞ défini par :

si $U, V \in \Sigma^\infty$, $U.V$ est le mot de Σ^∞ de longueur $\lg(U)+\lg(V)$ tel que :

$$\forall i \in [\lg(U)+\lg(V)], \quad U.V/i/ = \begin{cases} U/i/ & \text{si } i \leqq \lg(U), \\ V/i-\lg(U)/ & \text{sinon.} \end{cases}$$

Il résulte de cette définition que si U est infini $U.V = U$.

Sur Σ^∞ nous définissons la relation « préfixe » notée $\leqq$, par

$$U \leqq V \text{ ssi } \exists\, W \text{ tel que } U.W = V,$$

ce qui est équivalent à :

$$\forall i \in [\lg(U)], \quad U/i/ = V/i/.$$

Cette relation est une relation d'ordre qui fait de Σ^∞ un ensemble ordonné inductif (cpo) dont les éléments maximaux sont les mots infinis.

Outre cette structure de cpo, Σ^∞ peut être également muni d'une structure d'espace métrique au moyen de la distance :

$$d(U, V) = \begin{cases} 0 & \text{si } U = V, \\ 2^{-i} & \text{avec } i = \inf\{p \,/\, U/p+1/ \neq V/p+1/\}. \end{cases}$$

Cette distance est une distance ultramétrique et fait de Σ^∞ un espace métrique complet. Si de plus Σ est fini alors Σ^∞ est compact. Enfin Σ^ω étant un sous-espace fermé de Σ^∞ il est aussi complet (ou compact) et la topologie induite sur Σ^∞ par cette métrique peut se relativiser à Σ^ω.

Le produit de concaténation sur Σ^∞ ainsi que l'ordre préfixe s'étendent aisément, composante par composante, à des produits cartésiens $\Sigma_1^\infty \times \ldots \times \Sigma_n^\infty$ dont on notera parfois vectoriellement les éléments. En particulier on dira que

$\vec{v}=\langle v_1, \ldots, v_n\rangle$ est un préfixe de longueur p de $\vec{V}=\langle V_1, \ldots, V_n\rangle$ si $\forall i\in[n], v_i$ est préfixe de longueur p de V_i, c'est-à-dire $v_i \leqq V_i$ et $\lg(v_i)=p$.

De même la métrique d s'étend à $\Sigma_1^\omega \times \ldots \times \Sigma_n^\omega$ en posant :

$$d(\langle V_1, \ldots, V_n\rangle, \langle U_1, \ldots, U_n\rangle)=\max\{d(V_i, U_i)/i\in[n]\}.$$

Pour terminer ces préliminaires nous donnons d'abord la définition ces parties compacte que nous utiliserons :

une partie A d'un espace métrique E est compacte ssi toute suite $(x_i)_i$ d'éléments de A admet une valeur d'accumulation $x\in A$, c'est-à-dire $\forall\varepsilon>0$ il existe i tel que $d(x, x_i)<\varepsilon$ ou encore, ce qui est équivalent,

de toute suite $(x_i)_i$ d'éléments de A on peut extraire une sous-suite qui converge vers un élément de A.

Enfin nous utiliserons le lemme de König sous la forme suivante.

Soient $(A_i)_i$ une suite de parties finies non vides d'un ensemble E et R une relation binaire sur E telle que :

$$\forall i, \quad \forall x\in A_{i+1}, \quad \exists y\in A_i \quad \text{tel que } yRx;$$

dans ces conditions il existe une suite $(a_i)_i$ telle que $\forall i$, $a_i\in A_i$ et $a_i R a_{i+1}$.

2. LES PROCESSUS

Reprenant les spécifications de la notion de processus par Kahn [5] et Keller [6] ainsi que par Wiedmer [10] nous considérons qu'un processus transforme des suites infinies de données qui lui arrivent en entrée en une suite infinie de résultats. Cette transformation est réalisée par une machine vérifiant les conditions suivantes :

– elle a un nombre fini de « lignes » d'entrée et une ligne de sortie;

– chaque ligne d'entrée est munie d'une mémoire-tampon de dimension virtuellement infinie, fonctionnant en file FIFO, permettant d'enregistrer les données qui arrivent sur cette ligne;

– la machine dispose d'une mémoire interne infinie; elle est donc éventuellement capable de mémoriser toutes les informations reçues au cours de son activité;

– pour fonctionner, la machine doit acquérir des données sur certaines de ses lignes d'entrées; si ces données ne sont pas présentes elle se met en attente; son fonctionnement est donc asynchrone.

Il faut remarquer que les machines définies par Wiedmer n'obéissent pas exactement à cette dernière condition : elles n'acquièrent des données que sur

une seule ligne à la fois et si ces données ne sont pas présentes elles ne se mettent pas forcément en attente mais peuvent changer d'état, par exemple essayer de lire sur une autre ligne d'entrée.

En plus de ces conditions nous exigerons que la machine vérifie aussi la propriété suivante :

– dès que la machine a acquis une donnée sur chacune des lignes où elle en attendait une, ou bien elle se bloque et le processus est interrompu, ou bien elle fournit un ou plusieurs résultats sur la ligne de sortie.

C'est bien ce qui se passe pour la plupart des machines calculant des fonctions récursives partielles comme les machines de Turing : ou bien elles fournissent un résultat, ou bien elles calculent indéfiniment.

Une machine qui possède cette propriété sera appelée *machine à réponse non différée*. Pour elle l'absence de résultat sur la ligne de sortie alors qu'elle n'est pas en attente signifie qu'elle est bloquée. Cette condition est analogue à celle d'être de Greibach pour un schéma de programme récursif [1].

Nous donnons ci-dessous une définition formelle des processus, qui satisfait aux conditions données précédemment. Nous examinons d'abord le cas des processus déterministes, puis celui des processus non-déterministes et enfin nous adaptons les définitions au cas des machines ayant plusieurs lignes de sortie. Cette définition formelle des processus s'apparente à celle que donne R. Milner [7] en ce que toutes deux modélisent les processus par des transducteurs de mots infinis.

DÉFINITION 2.1 : Un *processus* à n entrées et 1 sortie est défini par le multiplet $P = \langle \Sigma_0, \Sigma_1, \ldots, \Sigma_n, Q, q_0, \delta, R \rangle$ où :

$\Sigma_0, \Sigma_1, \ldots, \Sigma_n$ sont des alphabets, éventuellement infinis; Q est un ensemble, éventuellement infini, *d'états*; q_0 est un élément de Q, *l'état initial*; δ est une application de Q dans $\mathscr{P}([n])$; R est un ensemble de règles.

Σ_0 est l'ensemble des résultats qui peuvent apparaître sur la ligne de sortie alors que pour $i \in [n]$, Σ_i est l'ensemble des données pouvant apparaître sur la i-ième ligne d'entrée.

L'application δ associe à chaque état les numéros des lignes sur lesquelles la machine doit acquérir une donnée quand elle est dans cet état.

DÉFINITION 2.2 : Chaque *règle* est de la forme $\langle q; u_1, \ldots, u_n \rangle \to \langle u; q' \rangle$ avec $q, q' \in Q$ et :

$$u \in \Sigma_0^+ = \Sigma_0 . \Sigma_0^*,$$

$$u_i \in \Sigma_i \quad \text{si} \quad i \in \delta(q),$$

$$u_i = \Lambda \quad \text{si} \quad i \notin \delta(q).$$

DÉFINITION 2.3 : Une *configuration* est un multiplet $\langle v_0; q; V_1, \ldots, V_n \rangle$, où v_0 est un mot fini de Σ_0^*, $q \in Q$, V_i est un mot fini ou infini de Σ_i^∞.

DÉFINITION 2.4 : Une configuration $c = \langle v_0; q; V_1, \ldots, V_n \rangle$ *se dérive en* une configuration $c' = \langle v_0'; q'; V_1', \ldots, V_n' \rangle$ ce qui notera $c \underset{P}{\vdash} c'$ si il existe une règle $\langle q; u_1, \ldots, u_n \rangle \to \langle u; q' \rangle$ telle que :

$$v_0' = v_0 . u \qquad \text{et} \qquad \forall i \in [n], \quad V_i = u_i . V_i'.$$

DÉFINITION 2.5 : Une *dérivation de longueur p* est une suite de configurations $c_0, c_1, \ldots, c_p$ telle que $\forall i \in [p]$, $c_{i-1} \underset{P}{\vdash} c_i$. De même une *dérivation infinie* est une suite infinie de configurations $c_0, c_1, \ldots$ telle que $\forall i \geqq 1$, $c_{i-1} \underset{P}{\vdash} c_i$.

Les résultats qui suivent découlent immédiatement des définitions.

PROPOSITION 2.1 : *Soit $c_0, c_1, \ldots, c_p$ une dérivation de longueur p avec $c_i = \langle v_0(i); q_i; V_1(i), \ldots, V_n(i) \rangle$. Alors :*

(i) *la suite $v_0(0), v_0(1), \ldots, v_0(p)$ est croissante pour l'ordre préfixe et $\forall i \in [p]$ $\lg(v_0(i-1)) < \lg(v_0(i))$;*

(ii) *pour tout $i \in [n]$, soit W_i un élément de Σ_i^∞. Pour tout $j \in [p]$ on pose $c_j' = \langle v_0(j); q_j; V_1(j).W_1, \ldots, V_n(j).W_n \rangle$.*
Alors $c_0', \ldots, c_p'$ est une dérivation de longueur p;

(iii) *pour tout $i \in [n]$ il existe $W_i \in \Sigma_i^\infty$ et une séquence $v_i(0), \ldots, v_i(p)$ de telle sorte que :*

- $\forall i \in [n]$, $\lg(v_i(0)) \leqq p$;
- $\forall i \in [n]$, $\forall_j \in [p+1]$, $V_i(j-1) = v_i(j-1).W_i$;
- *la suite $(\langle v_0(i); q_i, v_1(i), \ldots, v_n(i) \rangle)_{i=1,\ldots,p}$ est une dérivation.*

DÉFINITION 2.6 : Du point (i) de cette proposition il résulte que si $c_0, c_1, \ldots, c_p, \ldots$ est une dérivation infinie avec :

$$c_i = \langle v_0(i); q_i; V_1(i), \ldots, V_n(i) \rangle,$$

alors la suite $(v_0(i))_i$ est croissante et sa borne supérieure est un mot infini qui sera appelé *le résultat de la dérivation.*

DÉFINITION 2.7 : La fonction $\hat{P} : \Sigma_1^\omega \times \ldots \times \Sigma_n^\omega \to \mathscr{P}(\Sigma_0^\omega)$ associée au processus P est alors définie par : $\hat{P}(V_1, \ldots, V_n)$ est l'ensemble des résultats des dérivations infinies à partir de la configuration initiale $\langle \Lambda; q_0; V_1, \ldots, V_n \rangle$.

2.1. Les processus déterministes

DÉFINITION 2.8 : On dira qu'un processus P est *déterministe* si pour tout multiplet $\langle q, u_1, \ldots, u_n \rangle$ avec $q \in Q$:

$$u_i \in \Sigma_i \quad \text{si} \quad i \in \delta(q),$$
$$u_i = \Lambda \quad \text{si} \quad i \notin \delta(q),$$

il existe au plus une règle du processus ayant ce multiplet en partie gauche.

Il en résulte qu'à partir d'une configuration donnée c_0 il existe au plus une dérivation infinie et donc que la fonction $\hat{P}$ associée à ce processus peut être considérée comme une application partielle de $\Sigma_1^\omega \times \ldots \times \Sigma_n^\omega$ dans Σ_0^ω. On peut alors montrer :

PROPOSITION 2.2 : *Si P est un processus déterministe, la fonction $\hat{P}$ est continue et son domaine de définition est fermé.*

Démonstration : Soient $\vec{V} = \langle V_1, \ldots, V_n \rangle$ et $\vec{V}' = \langle V_1', \ldots, V_n' \rangle$ deux éléments de $\Sigma_1^\omega \times \ldots \times \Sigma_n^\omega$, où $\hat{P}$ est défini. Il existe donc deux dérivations infinies $c_0, c_1, \ldots$ et $c_0', c_1', \ldots$ de résultats respectifs W et W'. Posons :

$c_i = \langle v_i; q_i; V_1(i), \ldots, V_n(i) \rangle$ et $c_i' = \langle v_i'; q_i'; V_1'(i), \ldots, V_n(i) \rangle$;

on a donc $q_0 = q_0'$, $v_0 = v_0' = \Lambda$, $V_j(0) = V_j$ et $V_j'(0) = V_j'$. Supposons que, pour tout $i \in [n]$, V_i et V_i' aient un préfixe commun $\bar{v}_i$ de longueur supérieure ou égale à p, autrement dit $d(\vec{V}, \vec{V}') \leqq 2^{-p}$. Alors d'après les points (ii) et (iii) de la proposition 2.1 il existe deux dérivations de longueur p à partir de $\langle \Lambda; q_0; \bar{v}_1, \ldots, \bar{v}_n \rangle$ dont la première se termine par une configuration $\langle v_p; q_p; u_1, \ldots, u_n \rangle$ et l'autre par $\langle v_p'; q_p'; u_1', \ldots, u_n' \rangle$. Comme le processus est déterministe ces deux dérivations sont identiques donc $v_p = v_p'$. De plus v_p est préfixe commun à W et à W'. Comme d'après le point (i) de la proposition 2.1 la longueur de v_p est supérieure ou égale à p, $d(W, W') \leqq 2^{-p}$. Il en résulte que :

$$d(\vec{V}, \vec{V}') \leqq 2^{-p} \quad \Rightarrow \quad d(\hat{P}(\vec{V}), \hat{P}(\vec{V}')) \leqq 2^{-p}$$

et donc que $\hat{P}$ est continue sur son domaine de définition.

Soit maintenant $\vec{V}(1), \ldots, \vec{V}(p), \ldots$ une suite convergente d'éléments de $\Sigma_1^\omega \times \ldots \times \Sigma_n^\omega$ telle que $\hat{P}$ est définie en chaque point $\vec{V}(i) = \langle V_1(i), \ldots, V_n(i) \rangle$ et soit $\vec{V} = \langle V_1, \ldots, V_n \rangle$ sa limite. Pour chaque $i \in [n]$ posons $u_i(p)$ le préfixe de longueur p de V_i. Comme la suite $\vec{V}(i)$ converge vers $\vec{V}$, pour tout p il existe q tel que chaque $u_i(p)$ est préfixe de $V_i(q)$. Comme $\hat{P}$ est définie en $\vec{V}(q)$ il existe une dérivation infinie à partir de $\langle \Lambda, q_0, V_1(q), \ldots, V_n(q) \rangle$ et donc, en appliquant le point (iii) puis le point (ii) de la proposition 2.1, il existe une dérivation de longueur p à partir de $\langle \Lambda, q_0, V_1, \ldots, V_n \rangle$. Ceci étant vrai pour tout p et

compte tenu du fait que le processus est déterministe, il existe donc bien une dérivation infinie à partir de $\langle \Lambda, q_0, V_1, \ldots, V_n \rangle$ et donc $\hat{P}$ est défini en $\vec{V}$. □

On remarquera que la condition « réponse non différée » assure que le résultat d'une dérivation infinie est un mot infini et qu'en cela elle joue le même rôle que la condition de Greibach pour les grammaires ou les schémas de programme [1, 3, 8]. Si l'on retirait cette condition le résultat d'une dérivation infinie pourrait être un mot fini; on pourrait alors encore montrer que $\hat{P}$ est continue pour la topologie associée à l'ordre préfixe sur l'ensemble des mots finis ou infinis, comme le demandent Kahn et Keller, mais elle ne serait plus continue pour la topologie associée à la métrique comme le montre l'exemple ci-dessous.

EXEMPLE 2.1 : Soit $\Sigma = \{a, b\}$ et soit P le processus à 2 entrées et 1 sortie défini par les règles :

$$\langle q; a, a \rangle \to \langle \Lambda; q \rangle,$$
$$\langle q; a, b \rangle \to \langle b; q \rangle.$$

On a alors, pour tout n, $\hat{P}(a^\omega, a^n b^\omega) = b^\omega$ et pour $\langle a^\omega, a^\omega \rangle$, limite de la suite $(\langle a^\omega, a^n b^\omega \rangle)_n$ on obtient la dérivation infinie :

$$\langle \Lambda; q; a^\omega, a^\omega \rangle \underset{P}{\vdash} \langle \Lambda; q; a^\omega, a^\omega \rangle \ldots$$

Soit on considère que le résultat de cette dérivation est Λ, et $\hat{P}$ n'est pas continue, soit le résultat n'est pas défini, et le domaine de définition de $\hat{P}$ n'est pas fermé. □

2.2. Les processus finitaires

DÉFINITION 2.9 : Un processus P est *finitaire* si pour tout multiplet $\langle q; u_1, \ldots, u_n \rangle$ avec $q \in Q$ et :

$$u_i \in \Sigma_i \quad \text{si} \quad i \in \delta(q),$$
$$u_i = \Lambda \quad \text{si} \quad i \notin \delta(q),$$

il existe un nombre fini de règles de P dont ce multiplet est la partie gauche.

Il en résulte que le nombre de dérivations de longueur n à partir d'une configuration donnée est fini.

Il est bien évident que tout processus réalisé par un transducteur fini habituel est finitaire. C'est aussi le cas pour les transducteurs à pile et de manière plus générale pour tous les automates à nombre fini d'états et à mémoire interne pouvant contenir un mot sur un alphabet fini tels que leurs mouvements ne

dépendent que d'une portion bornée de la mémoire (par exemple le sommet de pile pour les automates à pile ou le symbole lu pour les machines de Turing), ne modifient qu'une portion bornée de cette mémoire et ne donnent en sortie qu'un nombre fini de mots différents.

Cette notion de non-déterminisme finitaire se trouve très explicitement énoncée dans Plotkin [9] qui ne considère que cette sorte de non-déterminisme, et Dijkstra a fait remarquer au cours d'une discussion (p. 181 de [11]) qu'on ne perdait rien à se limiter au non-déterminisme finitaire car le non-déterminisme non finitaire provoquerait des exécutions demandant une « quantité infinie d'énergie ».

Contrairement à ce qui se passe pour les processus déterministes, la fonction $\hat{P}$ associée à un processus finitaire (et *a fortiori* à un processus non-déterministe quelconque) n'est pas continue – avec comme topologie sur l'espace d'arrivée $\mathscr{P}(\Sigma_0^\omega)$ la topologie induite par la métrique de Hausdorff – ainsi que le montre l'exemple suivant.

EXEMPLE 2.2 : Soit P le processus à 1 entrée et 1 sortie défini par :

$$\langle \Sigma_0, \Sigma_1, Q, q_0, \delta, R \rangle,$$

avec $\Sigma_0 = \Sigma_1 = \{a, b\}$, $Q = \{q_0, q_1, q_2\}$, $\delta(q) = \{1\}$, pour tout $q \in Q$, et dont l'ensemble des règles est :

$$\begin{aligned}
&\langle q_0; a \rangle \to \langle a; q_1 \rangle,\\
&\langle q_0; a \rangle \to \langle b; q_2 \rangle,\\
&\langle q_0; b \rangle \to \langle b; q_2 \rangle,\\
&\langle q_1; a \rangle \to \langle a; q_1 \rangle,\\
&\langle q_2; a \rangle \to \langle b; q_2 \rangle,\\
&\langle q_2; b \rangle \to \langle b; q_2 \rangle.
\end{aligned}$$

Ce processus est bien finitaire.

On voit que pour tout mot infini U de $\{a, b\}$ il existe une dérivation infinie c_0, c_1, ... à partir de $\langle \Lambda; q_0; U \rangle$ avec, pour $i > 0$, $c_i = \langle b^i; q_2, U_i \rangle$ et que c'est la seule dérivation infinie possible si U contient la lettre b. Si $U = a^\omega$ il existe aussi une dérivation infinie $c_0, c'_1, \ldots$, avec, pour $i > 0$, $c'_i = \langle a^i; q_1; a^\omega \rangle$.

On a donc :

$$\hat{P}(U) = \begin{cases} \{b^\omega\} & \text{si } U \neq a^\omega, \\ \{a^\omega, b^\omega\} & \text{sinon.} \end{cases}$$

La suite $(a^n b^\omega)_n$ converge vers a^ω, alors que la suite $(\hat{P}(a^n b^\omega))_n$, qui est la suite constante $(\{b^\omega\})_n$ ne converge pas vers $\hat{P}(a^\omega) = \{a^\omega, b^\omega\}$ et donc $\hat{P}$ n'est pas continue.

Toutefois nous démontrerons que $\hat{P}$ est une application multivoque continue si l'on définit la continuité des applications multivoques comme le font Eilenberg et Montgomery [4].

DÉFINITION 2.10 : Une application f de $\Sigma_1^\omega \times \ldots \times \Sigma_n^\omega$ dans $\mathscr{P}(\Sigma_0^\omega)$ est continue (au sens d'Eilenberg et Montgomery) si pour toute suite $(\vec{U}(i))_i$ d'éléments de $\Sigma_1^\omega \times \ldots \times \Sigma_n^\omega$ qui converge vers $\vec{U}$ et pour toute suite $(V(i))_i$ d'éléments de Σ_0^ω qui converge vers V, si $V(i) \in f(\vec{U}(i))$ pour tout i alors $V \in f(\vec{U})$.

PROPOSITION 2.3 : *Soit P un processus finitaire :*

(i) *soit p un entier positif.*

Pour tout vecteur $\vec{u} = \langle u_1, \ldots, u_n \rangle$, où chaque u_i est un mot de Σ_i^* de longueur p il existe un ensemble fini $\rho(\vec{u})$ de mots de Σ_0^* de longueur p tel que pour tout $V \in \hat{P}(\vec{U})$ si $\vec{u}$ est préfixe de $\vec{U}$ alors le préfixe de V est dans $\rho(\vec{v})$;

(ii) *soit $(\vec{U}(i))_i$ une suite d'éléments du domaine de définition de $\hat{P}$ qui converge vers $\vec{U}$ et soit pour tout i, $V(i) \in \hat{P}(\vec{U}(i))$. Alors la suite $(V(i))_i$ a une valeur d'accumulation dans $\hat{P}(\vec{U})$.*

Avant de démontrer cette proposition remarquons que le point (i) peut s'écrire plus succintement : l'image par $\hat{P}$ d'une boule fermée de rayon 2^{-p} est incluse dans une réunion finie de boules fermées de même rayon.

Démonstration : Soit $\vec{U} \in \Sigma_1^\omega \times \ldots \times \Sigma_n^\omega$ et soit $\vec{u}$ son préfixe de longueur p. Soit Γ_p l'ensemble des configurations $c = \langle v; q; \vec{W} \rangle$ telles qu'il existe une dérivation $c_0, c_1, \ldots, c_p = c$ de longueur p à partir de $c_0 = \langle \Lambda; q_0; \vec{U} \rangle$. Comme P est finitaire Γ_p est fini.

En appliquant le point (iii) puis le point (ii) de la proposition 2.1 on voit que le résultat $\vec{V}$ de toute dérivation infinie à partir de $\vec{U}'$ ayant $\vec{u}$ pour préfixe aura pour préfixe un mot v tel que $\langle v; q; \vec{W} \rangle \in \Gamma_p$. Comme Γ_p est fini et que v est de longueur supérieure à p, ceci démontre le point (i).

Soit maintenant une suite $(\vec{U}(i))_i$ qui converge vers $\vec{U}$ et soit $V_i \in \hat{P}(\vec{U}(i))$ pour tout i. Définissons Γ_p comme ci-dessus et posons :

$$\Delta_p = \{\langle v; q; \vec{W} \rangle \in \Gamma_p / \exists k : v < V(k)\}.$$

Il est clair que Δ_p est fini. Comme $(\vec{U}(i))_i$ converge vers $\vec{U}$ il existe j tel que $\vec{u} < \vec{U}(j)$ et donc une dérivation infinie à partir de $\langle \Lambda; q_0; \vec{U}(j) \rangle$ dont le résultat est $\vec{V}(j)$. En appliquant le point (iii) de la proposition 2.1 on en déduit qu'il existe une dérivation de longueur p : $c_0 = \langle \Lambda; q_0; \vec{U} \rangle, \ldots, c_p = \langle v; q; \vec{W} \rangle$, où v

est un préfixe de $\vec{V}(j)$ de longueur supérieure à p et donc que Δ_p n'est pas vide. De plus il est immédiat que si $c \in \Delta_{p+1}$, il existe $c' \in \Delta_p$ tel que $c' \underset{P}{\vdash} c$. On peut donc appliquer le lemme de Koenig : il existe une dérivation infinie $c_0, c_1, \ldots, c_i, \ldots$ à partir de $c_0 = \langle \Lambda; q_0; \vec{U} \rangle$ avec $c_i = \langle v(i); q(i); \vec{W}(i) \rangle$ telle que pour tout i, $v(i)$ est un préfixe de $V(k)$ de longueur supérieure à i pour un certain k. Le résultat V de cette dérivation qui appartient à $\hat{P}(\vec{U})$, a donc des préfixes arbitrairement longs communs avec des éléments de la suite $(V(i))_i$ il est donc valeur d'accumulation de cette suite. □

De cette proposition nous déduisons.

PROPOSITION 2.4 : *Soit P un processus finitaire :*

(i) *le domaine de définition de $\hat{P}$ est fermé;*

(ii) *$\hat{P}$ est continue au sens d'Eilenberg-Montgomery;*

(iii) *l'image par $\hat{P}$ de tout compact est compacte.*

Démonstration : Les points (i) et (ii) de cette proposition découlent immédiatement du point (ii) de la proposition 2.3 [pour démontrer la continuité de $\hat{P}$ il suffit de prendre la suite $(V(i))_i$ convergente dans 2.3 (ii) et de remarquer que son unique valeur d'accumulation est sa limite].

Soit $\mathcal{U}$ une partie compacte du domaine de définition de $\hat{P}$ et soit $V(i))_i$ une suite de $P(\mathcal{U})$. Il existe donc une suite $(\vec{U}(i))_i$ d'éléments de $\mathcal{U}$ telle que $V(i) \in \hat{P}(\vec{U}(i))$ et comme $\mathcal{U}$ est compact on peut extraire de cette suite une sous-suite $(\vec{U}(h(i)))_i$ qui converge vers $\vec{U} \in \mathcal{U}$. D'après la proposition 2.3 (ii) la sous-suite $(V(h(i)))_i$ admet une valeur d'accumulation et donc aussi la suite $(V(i))_i$. □

Le point (iii) de cette proposition a pour corollaire que l'image de tout point par $\hat{P}$ est compacte et donc fermée.

Un exemple de processus non-déterministe est la fusion (« merge » en anglais) de deux séquences [6] qui consiste à rassembler les éléments des deux séquences en une seule et qui peut être réalisée par la machine suivante :

EXEMPLE 2.4 : Soit :

$$P = \langle \Sigma_0, \Sigma_1, \Sigma_2, Q, q_0, \delta, R \rangle,$$

le processus à deux entrées et 1 sortie avec $\Sigma_0 = \Sigma_1 \cup \Sigma_2$; $Q = \{q_0, q_1, q_2\}$; $\delta(q_0) = \{1, 2\}$; $\delta(q_1) = \{1\}$; $\delta(q_2) = \{2\}$ et dont les règles sont, en supposant que Σ_1 et Σ_2 contiennent un symbole spécial $\Subset$ qui sert de marqueur de début de mot :

$$\langle q_0; \Subset, \Subset \rangle \to \langle \Subset; q_1 \rangle,$$

$$\langle q_0; \Subset, \Subset \rangle \to \langle \Subset; q_2 \rangle,$$

$$\left.\begin{array}{l}\langle q_1; x, \Lambda\rangle \to \langle x; q_1\rangle \\ \langle q_1; x, \Lambda\rangle \to \langle x; q_2\rangle\end{array}\right\} \text{ pour tout } x \in \Sigma_1 - \{\not\Subset\},$$

$$\left.\begin{array}{l}\langle q_2; \Lambda, y\rangle \to \langle y; q_1\rangle \\ \langle q_2; \Lambda, y\rangle \to \langle y; q_2\rangle\end{array}\right\} \text{ pour tout } y \in \Sigma_2 - \{\not\Subset\}.$$

Formellement la fusion de deux séquences $X \in \Sigma_1^\omega$ et $Y \in \Sigma_2^\omega$ donne l'ensemble des séquences de la forme $u_1 v_1 u_2 v_2 \ldots u_p v_p \ldots$ avec $u_i \in \Sigma_i^*$ et $v_i \in \Sigma_2^*$ telles que $X = u_1 u_2 \ldots u_p \ldots$ et $Y = v_1 v_2 \ldots v_p \ldots$

On voit que cette définition n'exclut pas le cas où tous les v_i (ou, symétriquement, tous les u_i) seraient égal au mot vide, et, de fait le processus de l'exemple précédent calcule bien (au marqueur $\not\Subset$ près) les séquences $u_1 v_1 u_2 v_2 \ldots u_p v_p U$ avec $u_1 u_2 \ldots u_p U = X$ et $v_1 \ldots v_p$ préfixe de Y (et symétriquement).

La fusion qui ne fournit que les séquences où tous les éléments de chacune des séquences d'entrée se retrouvent dans la séquence de sortie s'appelle fusion équitable (fair merge) et peut être réalisée par le processus suivant :

EXEMPLE 2.5 : Soit :

$$P = \langle \Sigma_0, \Sigma_1, \Sigma_2, Q, q_0, \delta, R\rangle,$$

avec Σ_0, Σ_1 et Σ_2 comme dans l'exemple précédent :

$$Q = \{q_0\} \cup \{q_i^1 \mid i \geqq 1\} \cup \{q_i^2 \mid i \geqq 1\},$$
$$\delta(q_0) = \{1, 2\}; \qquad \delta(q_i^1) = \{1\}; \qquad \delta(q_i^2) = \{2\}$$

et dont les règles sont :

$$\langle q_0; \not\Subset, \not\Subset\rangle \to \langle \not\Subset ; q_i^j\rangle \quad \text{pour tout } i \geqq 1 \quad \text{et tout } j \in \{1, 2\},$$
$$\langle q_{i+1}^1; x, \Lambda\rangle \to \langle x; q_i^1\rangle \quad \text{pour tout } x \in \Sigma_1 \quad \text{et tout } i \geqq 1,$$
$$\langle q_{i+1}^2; \Lambda, y\rangle \to \langle y; q_i^2\rangle \quad \text{pour tout } y \in \Sigma_2 \quad \text{et tout } i \geqq 1,$$
$$\langle q_1^1; x, \Lambda\rangle \to \langle x; q_i^2\rangle \quad \text{pour tout } x \in \Sigma_1 \quad \text{et tout } i \geqq 1,$$
$$\langle q_1^2; \Lambda, y\rangle \to \langle y; q_i^1\rangle \quad \text{pour tout } y \in \Sigma_2 \quad \text{et tout } i \geqq 1.$$

Ce processus n'est pas finitaire, ce qui est normal car :

PROPOSITION 2.5 : *La fusion équitable ne peut pas être réalisée par un processus finitaire.*

Démonstration : Supposons que la fusion équitable soit réalisée par un processus finitaire P et soient :

$$X = x_1 x_2 \ldots x_n \ldots \in \Sigma_1^\omega \qquad \text{et} \qquad Y = y_1 y_2 \ldots y_n \ldots \in \Sigma_2^\omega.$$

Il est clair que $\hat{P}(X, Y)$ contient tous les mots :

$$V_n = x_1 x_2 \ldots x_n y_1 y_2 \ldots y_n x_{n+1} y_{n+1} x_{n+2} y_{n+2} \ldots$$

et que la suite $(V_n)_n$ converge vers X. D'après la proposition 2.4. $\hat{P}(X, Y)$ est fermé et donc cet ensemble contient X. Or X ne peut manifestement pas être produit par la fusion équitable de X et de Y. P ne peut donc pas être finitaire. □

2.3. Les processus à plusieurs sorties

Si nous avons considéré jusqu'à présent des processus à une seule sortie ce n'est que pour avoir des notations simplifiées dans les démonstrations précédentes. Il n'y a en effet aucune raison de ne pas considérer des processus à plusieurs lignes de sortie.

Mais dans ce cas la condition de réponse non différée est que la machine doit fournir, à chaque « mouvement » au moins un résultat sur chacune des lignes de sorties.

Formellement il suffit de remplacer dans la définition d'un processeur, l'alphabet de sortie par m alphabets $\Delta_1, \ldots, \Delta_m$ (un par ligne); les règles deviennent alors de la forme :

$$\langle q; u_1, \ldots, u_n \rangle \to \langle v_1, \ldots, v_m; q' \rangle \qquad \text{avec} \quad v_i \in \Delta_i^+ .$$

Tous les résultats obtenus précédemment restent alors valables en remplaçant Σ_0^ω par $\Delta_1^\omega \times \ldots \times \Delta_m^\omega$ muni de la topologie produit.

Un exemple important de processus à plusieurs sorties est le « duplicateur » (« fork » dans [6]) qui réalise plusieurs copies d'une même entrée.

EXEMPLE 2.6 : Soit le processus déterministe :

$$P = \langle \Delta_1, \Delta_2, \Sigma_1, \{q\}, q, \delta, R \rangle,$$

avec :

$$\Delta_1 = \Delta_2 = \Sigma_1, \qquad \delta(q) = \{1\}$$

et dont les règles sont $\langle q; x \rangle \to \langle x, x; q \rangle$ pour tout $x \in \Sigma_1$. Il est clair que $\hat{P}(U) = \langle U; U \rangle$.

3. LES RÉSEAUX DE PROCESSUS

Des processus forment un réseau lorsque des lignes de sortie de certains processus servent de lignes d'entrée à d'autres. Chaque processus d'un réseau a

donc deux sortes de lignes d'entrée : celles qui sont aussi des lignes de sortie et qui sont alimentées en données par le fonctionnement même du réseau et les autres dont nous supposerons qu'elles sont alimentées par l'environnement extérieur au réseau. Il conviendra par la suite de distinguer ces deux types de lignes d'entrée.

DÉFINITION 3.1 : Soient $P_1, \ldots, P_k, k$ processus, le processus P_i ayant n_i entrées et m_i sorties, et soit p un entier quelconque. Posons $n = n_1 + n_2 + \ldots + n_k$ et $m = m_1 + \ldots + m_k$. Un *réseau* R est la donnée de ces p processus, de l'entier p et de deux injections $\eta : [n] \to [p]$ et $\sigma : [m] \to [p]$.

La signification intuitive de cette définition est la suivante.

La j-ième entrée du processus P_i — on a donc $1 \leqq j \leqq n_i$ — reçoit le numéro $n_0 + n_1 + \ldots + n_{i-1} + j$, où $n_0 = 0$. De même la j'-ième sortie du processus P_i, reçoit le numéro $m_0 + m_1 + \ldots + m_{i'-1} + j'$. Les lignes du réseau sont numérotées de 1 à k. L'entier $\eta(l)$ est donc le numéro de la ligne connectée à l'entrée dont le numéro est l et l'entier $\sigma(l)$ est le numéro de la ligne connectée à la sortie dont le numéro est l.

Nous pouvons donc supposer que $\eta([n]) \cup \sigma([m]) = [p]$.

EXEMPLE 3.1 : Soient trois processus ayant chacun deux entrées et deux sorties. Les injections η et σ de [6] dans [8] sont définies par le tableau :

	η	σ
1	3	5
2	1	6
3	2	7
4	4	8
5	6	3
6	7	4

Le réseau peut être représenté graphiquement (*voir* figure page suivante).

DÉFINITION 3.2 : Soit R un réseau défini comme ci-dessus. Pour $i \in [k]$ et $j \in [n_i]$ nous posons $\eta_i(j) = \eta\ (n_0 + n_1 + \ldots + n_{i-1} + j)$ qui est le numéro de la ligne connectée à la j-ième entrée de P_i. De même :

$$\sigma_i(j) = \sigma(m_0 + m_1 + \ldots + m_{i-1} + j).$$

Nous posons aussi :

$$S_i = \sigma_i([m_i]) \quad \text{pour tout } i \in [k],$$

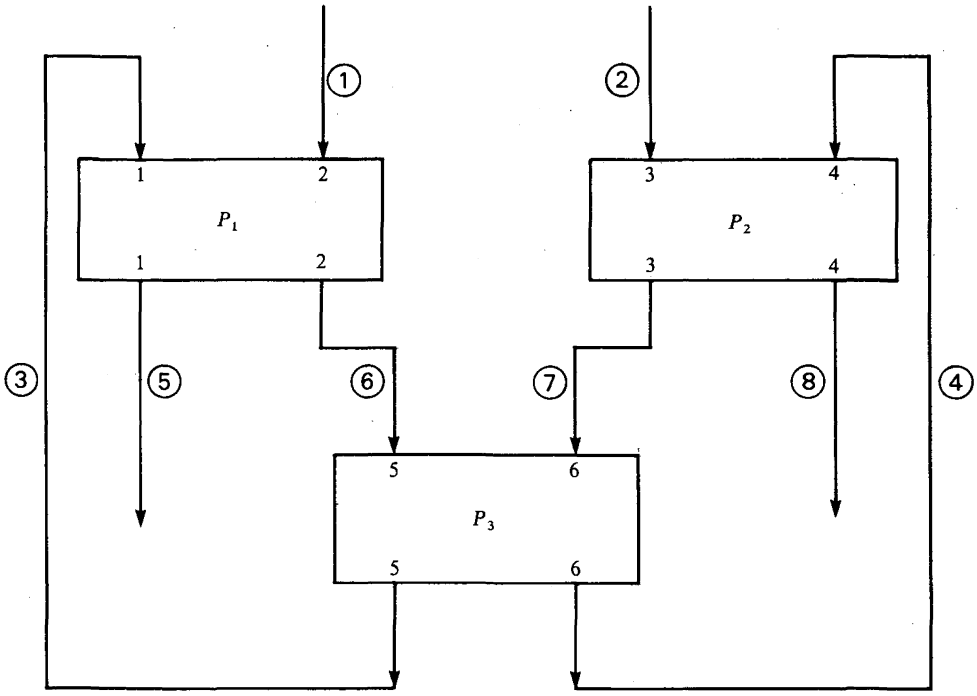
$$S = S_1 \cup S_2 \ \ldots \cup S_k = \sigma([n]),$$

$$\left.\begin{aligned} F_i &= \eta_i([n_i]) \\ E_i &= F_i \cap S \end{aligned}\right\} \quad \text{pour tout } i \in [k],$$

$$F = F_1 \cup F_2 \ldots \cup F_k = \eta([n]),$$

$$E = E_1 \cup E_2 \ldots \cup E_k = F \cap S.$$

Il est clair que pour $i \neq j$, $E_i \cap E_j = F_i \cap F_j = S_i \cap S_j = \emptyset$ et que $E \subset S$.



EXEMPLE 3.1 (*suite*) : Pour l'exemple précédent les valeurs des ensembles définis ci-dessus sont données par le tableau :

i	S_i	F_i	E_i
1	{5, 6}	{1, 3}	{3}
2	{7, 8}	{2, 4}	{4}
3	{3, 4}	{6, 7}	{6, 7}
	{3, 4, 5, 6, 7, 8}	{1, 2, 3, 4, 6, 7}	{3, 4, 6, 7}
	S	F	E

DÉFINITION 3.3 : Nous associons à chaque ligne i le symbole X_i. Le réseau R peut alors être représenté par le schéma :

$$\begin{cases} \langle X_{\eta_i(1)}, \ldots, X_{\eta_i(n_i)} \rangle \leftarrow P_i(X_{\sigma_i(1)}, \ldots, X_{\sigma_i(m_i)}), \\ i = 1, \ldots, k, \end{cases}$$

ou encore de manière plus compacte, en posant :

$$\vec{X} = \langle X_1, \ldots, X_p \rangle \quad \text{et} \quad \vec{P} = \langle P_1, \ldots, P_k \rangle,$$
$$\eta(\vec{X}) \leftarrow \vec{P}(\sigma(\vec{X})),$$

et même, par abus de notation :

$$\vec{X} \leftarrow \vec{P}(\vec{X}).$$

Si de plus on veut faire apparaître les lignes qui ne sont que des lignes d'entrées on leur associera le sysmbole Y_i au lieu du symbole X_i et on écrira, toujours par abus de notation :

$$\vec{X} \leftarrow \vec{P}(\vec{X}, \vec{Y}).$$

EXEMPLE 3.1 (*suite*) : Le réseau considéré dans cet exemple peut être défini sans ambiguïté par le schéma :

$$\langle X_5, X_6, X_7, X_8, X_3, X_4 \rangle \leftarrow \langle P_1, P_2, P_3 \rangle (X_3, Y_1, Y_2, X_4, X_6, X_7).$$

3.1. Évolution d'un réseau

Le but de ce paragraphe est de donner une définition formelle d'une sémantique opérationnelle des réseaux que nous croyons naturelle et raisonnable.

DÉFINITION 3.4 : Une *configuration* du réseau $\langle P_1, \ldots, P_k, \eta, \sigma \rangle$ est un multiplet $\langle u_1, \ldots, u_p, q_1, \ldots, q_k; V_1, \ldots, V_p \rangle$ tel que :

$$\forall_i \in [k], \quad q_i \text{ est un état de } P_i;$$
$$\forall_j \in [p], \quad \text{si } j \notin S \text{ alors } u_j = \Lambda,$$

sinon u_j est un mot fini sur le « bon » alphabet [i. e. sur l'alphabet de la i-ième ligne de sortie du l-ième processus, avec $j = \sigma_l(i)$] ;

$$\forall j \in [p], \quad \text{si } j \notin F \text{ alors } V_j = \Lambda,$$

sinon V_j est un mot (fini si $j \in E$, infini si $j \in F - E$) sur le « bon » alphabet [i. e. sur l'alphabet de la i-ième ligne d'entrée du l-ième processeur, avec $j = \eta_l(i)$].

DÉFINITION 3.5 : Une *configuration initiale* du réseau est une configuration $\langle u_1, \ldots, u_p; q_1, \ldots, q_k; V_1, \ldots, V_p \rangle$ telle que chaque u_i est le mot vide et chaque q_j est l'état initial du processus P_j.

DÉFINITION 3.6 : Une configuration $c=\langle u_1, \ldots, u_p; q_1, \ldots, q_k; V_1, \ldots, V_p \rangle$ *évolue* en une configuration $c'=\langle u'_1, \ldots, u'_p; q'_1, \ldots, q'_k; V'_1, \ldots, V'_p \rangle$ *par le processus* P_{i_0}, ce qui notera $c \underset{P_{i_0}}{\vdash} c'$ si :

– $\forall i \neq i_0$;

$$1 \leqq i \leqq k \quad \Rightarrow \quad q'_i = q_i,$$

– il existe une règle $\langle w_1, \ldots, w_l; q_{i_0} \rangle \rightarrow \langle q'_{i_0}; v_1, \ldots, v_{l'} \rangle$ du processus P_{i_0}, avec $l=n_{i_0}$ et $l'=m_{i_0}$ et un p-uple $\langle W_1, \ldots, W_p \rangle$ tels que $\forall j \in [p]$:

$$u'_j = u_j . \bar{v}_j,$$
$$V_j = \bar{w}_j W_j,$$
$$V'_j = W_j \bar{v}_j,$$

avec :

$$\bar{v}_j = \begin{cases} v_{j'} & \text{si } \ j=\sigma_{i_0}(j') \text{ et si } j \in F, \\ \Lambda & \text{sinon,} \end{cases}$$

et :

$$\bar{w}_j = \begin{cases} w_{j'} & \text{si } \ j=\eta_{i_0}(j'), \\ \Lambda & \text{sinon.} \end{cases}$$

Cela signifie que lorsque le processus P_{i_0} fonctionne il lit des symboles dans les mémoires-tampons associées à ses lignes d'entrées et que les symboles qu'il produit sur une ligne de sortie sont remis dans la mémoire-tampon associée à la ligne d'entrée à laquelle cette ligne de sortie est (éventuellement) reliée.

On voit que si $j \notin S$ alors $u_j=\Lambda$ et $\bar{v}_j=\Lambda$ d'où $u'_j=\Lambda$ et que si $j \notin F$ alors $V_j=\Lambda$, $\bar{w}_j=\Lambda$ et $\bar{v}_j=\Lambda$ d'où $V'_j=\Lambda$. La définition de la relation $\underset{P_i}{\vdash}$ est donc compatible avec la définition des configurations.

DÉFINITION 3.7 : On notera $\vdash$ la réunion des relations $\underset{P_i}{\vdash}$, et $\underset{r}{\vdash}$, où $r=P_{i_1}$, $P_{i_2} \ldots P_{i_l}$, le composé des relations $\underset{P_{i_1}}{\vdash}$, $\underset{P_{i_2}}{\vdash}$, $\ldots$, $\underset{P_{i_l}}{\vdash}$.

DÉFINITION 3.8 : Une *évolution de longueur* l à partir de la configuration c_0 est une suite $c_0\ P_{i_1}, c_1, \ldots, P_{i_l}, c_l$ telle que $c_{j-1} \underset{P_{i_{j-1}}}{\vdash} c_j$ pour tout $j \in [l]$.

Une *évolution infinie* à partir de la configuration c_0 est une suite $c_0, P_{i_1}, c_1, \ldots, P_{i_n}, c_n, \ldots$ telle que $c_j \underset{P_{i_j}}{\vdash} c_{j+1}$ pour tout $j \geqq 0$.

Une évolution infinie est *équitable* si pour tout $l \in [k]$ il existe une infinité d'entiers j tels que $i_j = l$. Autrement dit chaque processus intervient une infinité de fois dans cette évolution.

DÉFINITION 3.9 : Soit $c_0, P_{i_1}, c_1, P_{i_2}, \ldots$ une évolution infinie. Posons :

$$c_i = \langle u_1(i), \ldots, u_p(i); q_1(i), \ldots, q_k(i); V_1(i), \ldots, V_p(i) \rangle.$$

Par définition de la relation $\vDash$ chaque suite $(u_j(i))_i$ est croissante pour l'ordre préfixe et admet donc une borne supérieure U_j.

Nous appellerons *résultat* de l'évolution infinie le vecteur $\langle U_1, \ldots, U_n \rangle$.

Remarquons que si $j \notin S$ chaque $u_j(i)$ est égal à Λ et donc $U_j = \Lambda$. Nous dirons que l'évolution est *acceptable* si U_j est un mot infini pour tout $j \in S$.

PROPOSITION 3.1 : *Une évolution infinie est acceptable si et seulement si elle est équitable.*

Démonstration : Soit $c_0, P_{i_1}, c_1, \ldots$ une évolution infinie et posons :

$$c_j = \langle u_1(j), \ldots, u_p(j); q_1(j), \ldots, q_k(j); V_1(j), \ldots, V_p(j) \rangle.$$

Il est clair que pour $r \in S_l$,

$$|u_r(j)| \geq |u_r(0)| + \text{Card}\{s \leq j / i_s = l\},$$

d'où U_r est un mot infini si Card $\{s/i_s = l\} = \infty$. Comme $S = \bigcup_{l=1,\ldots,k} S_l$, l'évolution est acceptable si elle est équitable.

Réciproquement supposons que l'évolution ne soit pas équitable. Il existe donc $l \in [k]$ et un entier j tel que $s > j \Rightarrow i_s \neq l$. Pour tout $r \in S_l$ on a donc $s > j \Rightarrow u_r(s) = u_r(j)$ d'où U_r est un mot fini et l'évolution n'est pas acceptable. □

DÉFINITION 3.10 : Soit $\vec{V} = \langle V_1, \ldots, V_n \rangle$ tel que :
si $i \in E$, V_i est un mot fini;
si $i \in F - E$, V_i est un mot infini;
si $i \notin F$, $V_i = \Lambda$.

Nous noterons Res$(\vec{V})$ l'ensemble des résultats des évolutions infinies acceptables à partir de la configuration initiale :

$$\langle \Lambda, \ldots, \Lambda; q_1, \ldots, q_k; V_1, \ldots, V_p \rangle.$$

3.2. La condition de déclenchement

Soit une évolution finie $c_0, P_{i_1}, c_1, \ldots, P_i, c_l$ de longueur l. Si cette évolution ne peut pas se prolonger en une évolution de longueur $l+1$ c'est qu'aucun processus P_j ne peut faire évoluer la configuration c_l.

Ceci peut être due à deux raisons :

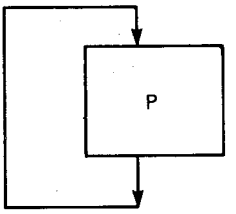
– les entrées que lit le processus P_l sont telles qu'il se bloque;

– la mémoire-tampon associée à l'une des entrées sur lesquelles P doit lire une donnée est vide.

Nous allons dans cette section étudier les réseaux pour lesquels cette seconde situation ne peut pas se présenter. Mais auparavant nous donnons un exemple de ces deux possibilités.

EXEMPLE 3.2 : Soit P le processus à 1 entrée et 1 sortie sur le même alphabet $\{a, b\}$ ayant un seul état q et dont l'unique règle est $\langle q; b \rangle \to \langle b; q \rangle$.

Soit le réseau défini par $X = P(X)$ représenté graphiquement par :



Le processus P ne peut pas faire évoluer la configuration initiale $\langle \Lambda; q; a \rangle$ parce que P bloque sur l'entrée a (1^{ex} cas). Il ne peut pas faire évoluer non plus la configuration initiale $\langle \Lambda; q; \Lambda \rangle$ parce qu'il n'a pas de donnée en entrée.

Par contre, à partir de la configuration initiale $\langle \Lambda; q; b \rangle$ on obtient l'évolution infinie acceptable :

$$\langle \Lambda; q; b \rangle,\ P,\ \langle b; q; b \rangle,\ P,\ \langle bb; q; b \rangle,\ \ldots,\ P,\ \langle b^n; q; b \rangle \ldots$$

dont le résultat est b^ω.

Pour que le processus P_i puisse faire évaluer une configuration c il faut donc qu'il y ait des données sur toutes les entrées que le processus P_i va examiner dans l'état où il se trouve. Or par définition d'une configuration il y a des mots infinis aux entrées qui ne sont pas dans E. Ceci nous amène à poser les définitions suivantes.

DÉFINITION 3.11 : Soit $c = \langle u_1, \ldots, u_p; q_1, \ldots, q_k; V_1, \ldots, V_p \rangle$ une configuration.

Nous posons $\gamma(c) = \{ i \in E / V_i \neq \Lambda \}$.

Pour chaque $i \in [k]$ nous posons $\bar{\delta}_i(c) = \eta_i(\delta(q_i)) \cap E$.

Comme $\delta(q_i) \subset [n_i]$, $\eta_i(\delta(q)) \subset F_i$ et donc :

$$\bar{\delta}_i(c) \subset F_i \cap E = F_i \cap S = E_i.$$

Nous dirons que le processus P_i *est déclenchable dans la configuration* c si :

$$\bar{\delta}_i(c) \subset \gamma(c).$$

PROPOSITION 3.2 : *Soient c et c' deux configurations et $i \in \{1, \ldots, k\}$ tels que $c \underset{P_i}{\vdash} c'$. Alors P_i est déclenchable dans c et :*

$$(\gamma(c) - \bar{\delta}_i(c)) \cup (S_i \cap E) \subset \gamma(c') \subset \gamma(c) \cup (S_i \cap E)$$

Démonstration : Soient :

$$c = \langle u_1, \ldots, u_p; q_1, \ldots, q_k; V_1, \ldots, V_p \rangle$$

et :

$$c' = \langle u'_1, \ldots, u'_p; q'_1, \ldots, q'_k; V'_1, \ldots, V'_p \rangle.$$

Par définition de la relation $\underset{P_i}{\vdash}$ on a $u'_j = u_j . \bar{v}_j$; $V_j = \bar{w}_j W_j$; $V'_j = W_j \bar{v}_j$ avec $\bar{w}_j = \Lambda$ si $j \notin F_i$, $\bar{v}_j = \Lambda$ si $j \notin S_i \cap F = S_i \cap E$, et :

$$\langle q_i; \bar{w}_{\eta_i(1)}, \ldots, \bar{w}_{\eta_i(n_i)} \rangle \to \langle \bar{v}_{\sigma_i(1)}, \ldots, \bar{v}_{\sigma_i(m_i)}; q'_i \rangle$$

est une règle de P_i.

On a alors $\bar{w}_{\eta_i(l)} \neq \Lambda$ si et seulement si $l \in \delta(q_i)$, d'où $l' \in \bar{\delta}_i(c)$ entraîne $\bar{w}_{l'} \neq \Lambda$ et donc $V_{l'} \neq \Lambda$ d'une part, $l' \in E$ d'autre part d'où $l' \in \gamma(c)$.

Soit maintenant $j \in E$ tel que $V'_j = \Lambda_c$ [c'est-à-dire $j \notin \gamma(c')$]. Comme $\bar{v}_l \neq \Lambda$ pour $l \in \sigma_i([m_i]) = S_i$, $j \notin S_i \cap E$. On a alors $\bar{v}_j = \Lambda$ et donc $V'_j = W_j = \Lambda$, d'où $V_j = \bar{w}_j$. Supposons que $j \in \gamma(c) - \bar{\delta}_i(c)$; puisque $j \in \gamma(c)$, $V_j = \bar{w}_j \neq \Lambda$ et puisque $j \notin \bar{\delta}_i(c)$, $\bar{w}_j = \Lambda$ d'où une contradiction. Il en résulte que $j \notin \gamma(c')$ entraîne $j \notin S_i \cap E$ et $j \notin \gamma(c) - \bar{\delta}_i(c)$.

Soit enfin $j \in \gamma(c')$, c'est-à-dire $j \in E$ et $V'_j \neq \Lambda$. On a donc $\bar{v}_j \neq \Lambda$ ou $W_j \neq \Lambda$. Si $\bar{v}_j \neq \Lambda$ alors $j \in S_i \cap E$. Si $W_j \neq \Lambda$ alors $V_j = \bar{w}_i W_j \neq \Lambda$ et $j \in \gamma(c)$. □

LEMME 3.3 : *Soient $G_0, G_1, \ldots, G_l$ une suite finie de parties de E, I une partie propre de $[k]$ et h une surjection de $[l]$ sur I telles que :*

$$\forall i \in [l], \quad (G_{i-1} - E_{h(i)}) \cup S'_{h(i)} \subset G_i \subset G_{i-1} \cup S'_{h(i)},$$

où $S'_j = S_j \cap E$.

Alors pour tout $j \notin I$ et pour toute partie Δ de E_j :

$$\Delta \cap G_l = \Delta \cap (G_0 \cup \bigcup_{i \in I} S_i).$$

Démonstration : Montrons par récurrence que :

$$\forall i \in \{0, \ldots, l\}, \quad \Delta \cap G_i = \Delta \cap \left(G_0 \cup \bigcup_{s=1}^{i} S_{h(s)} \right).$$

Cette égalité est trivialement vraie pour $i=0$. Supposons qu'elle soit vraie pour un i quelconque et calculons $\Delta \cap G_{i+1}$.

Par hypothèse :

$$\Delta \cap [(G_i - E_{h(i+1)}) \cup S'_{h(i+1)}] \subset \Delta \cap G_{i+1} \subset \Delta \cap (G_i \cup S'_{h(i+1)}).$$

Comme Δ est inclus dans E_j avec $j \neq h(i+1)$ et que tous les E_i sont disjoints deux à deux, $\Delta \cap (G_i - E_{h(i+1)}) = \Delta \cap G_i$.

D'autre part, puisque $\Delta \subset E$, $\Delta \cap S'_{h(i+1)} = \Delta \cap S_{h(i+1)}$. On en déduit que $\Delta \cap G_{i+1} = (\Delta \cap G_i) \cup (\Delta \cap S_{h(i+1)})$ et, d'après l'hypothèse de récurrence :

$$\Delta \cap G_{i+1} = \Delta \cap \left(G_0 \cup \bigcup_{s=1}^{i+1} S_{h(s)} \right). \quad \square$$

PROPOSITION 3.4 : *Soit une évolution $c_0, P_{\theta(1)}, c_1, \ldots, P_{\theta(k)}, c_k$, où θ est une permutation de $[k]$. Alors le processus $P_{\theta(1)}$ est déclenchable dans la configuration c_k.*

Démonstration : Pour simplifier les notations nous supposerons que la permutation θ est l'identité. Il suffit pour cela de renuméroter convenablement les processus.

D'après la proposition 3.2, on a :

$$S'_i \cup (\gamma(c_{i-1}) - \bar{\delta}_i(c_{i-1})) \subset \gamma(c_i) \subset \gamma(c_{i-1}) \cup S'_i$$

et *a fortiori*, puisque $\bar{\delta}_i(c_{i-1}) \subset E_i$:

$$(\gamma(c_{i-1}) - E_i) \cup S'_i \subset \gamma(c_i).$$

On peut donc appliquer le lemme 3.3 avec $G_i = \gamma(c_{i+1})$, $I = \{2, \ldots, k\}$ et $\Delta = E_1$, ce qui donne :

$$\gamma(c_k) \cap E_1 = E_1 \cap \left(\gamma(c_1) \cup \bigcup_{i=2}^{k} S_i \right).$$

Mais $S'_1 \subset (\gamma(c_0) - E_1) \cap S'_1 \subset \gamma(c_1)$, d'où :

$$E_1 \cap S'_1 = E_1 \cap S_1 \subset E_1 \cap \gamma(c_1)$$

et

$$E_1 \cap \left(\bigcup_{i=1}^{k} S_i \right) \subset E_1 \cap \left(\gamma(c_1) \cup \bigcup_{i=2}^{k} S_i \right) = \gamma(c_k) \cap E_1.$$

Mais, par définition $E \subset \bigcup_{i=1}^{k} S_i$, d'où $E_1 \subset \gamma(c_k) \cap E_1$ et donc $E_1 \subset \gamma(c_k)$.

Il en résulte que P_1 est déclenchable dans la configuration c_k puisque $\bar{\delta}_1(c_k) \subset E_1$.

On voit d'après cette proposition que s'il existe une permutation des processus telle qu'il existe une évolution où les processus sont déclenchés selon l'ordre indiqué par θ, alors en poursuivant l'évolution par le déclenchement cyclique des processus selon θ on atteindra toujours une configuration dans laquelle le processus suivant sera déclenchable.

Ceci nous amène à poser la définition suivante :

DÉFINITION 3.12 : Pour tout $i \in [k]$ posons $\Delta_i = \eta_i(\delta(q_i(0))) \cap E$, où $q_i(0)$ est l'état initial du processus P_i.

Nous dirons qu'une configuration initiale c du réseau *vérifie la condition de déclenchement* s'il existe une suite $\Gamma_0, \Gamma_1, \ldots, \Gamma_{k-1}$ de parties de E et une permutation θ de $[k]$ telles que :

$$\Gamma_0 = \gamma(c); \qquad \Delta_{\theta(1)} \subset \Gamma_0;$$

et :

$$\forall i \in [k-1], \quad \Gamma_i = (\Gamma_{i-1} - \Delta_{\theta(i)}) \cup (S_{\theta(i)} \cap E) \qquad \text{et} \qquad \Delta_{\theta(i+1)} \subset \Gamma_i.$$

Par définition cette condition ne dépend que des états initiaux des processus du réseau et de $\gamma(c)$ c'est-à-dire de la présence ou de l'absence de données sur les entrées du réseau dans sa configuration initiale.

Il est clair que cette condition de déclenchement est décidable.

PROPOSITION 3.5 : *Si il existe une évolution infinie équitable à partir d'une configuration initiale c, alors cette configuration vérifie la condition de déclenchement.*

Démonstration : Soit $c_0, P_{i_1}, c_1, P_{i_2} \ldots$ une évolution infinie équitable à partir de la configuration initiale $c = c_0$.

Posons $I_0 = \emptyset$ et pour tout entier $j \geqq 1$, $I_j = \{i_1, \ldots, i_j\}$. La suite $(I_j)_j$ est la suite croissante de parties de $[k]$. Comme l'évolution est équitable il existe un j tel

que $I_j=[k]$. De plus le cardinal de I_{j+1} est au plus supérieur de 1 à celui de I_j.

Nous pouvons donc définir, pour tout $l\in\{0, \ldots, k\}$, $\mu(l)$ comme étant le plus petit entier j tel que $\text{Card}(I_j)=l$ et la permutation θ de $[k]$ par :

$$\theta(l) \quad \text{est l'unique élément de } I_{\mu(l)}-I_{\mu(l-1)}.$$

Il résulte de cette définition que pour tout $l\in[k]$ et tout j tel que $\mu(l-1)\leqq j<\mu(l)$, $I_j=I_{\mu(l-1)}=\{\theta(i)/1\leqq i<l\}$ et $\theta(l)\notin I_j$.

Posons aussi :

$$\Gamma_0=\gamma(c_0),$$

et pour tout $i\in\{1, \ldots, k-1\}$:

$$\Gamma_i=(\Gamma_{i-1}-\Delta_{\theta(i)})\cup(S_{\theta(i)}\cap E).$$

Pour que c_0 vérifie la condition de déclenchement il suffit que $\Delta_{\theta(i-1)}\subset\Gamma_i$ pour tout $i\in\{0, \ldots, k-1\}$.

Par définition la première configuration dans laquelle se déclenche le processus $P_{\theta(j)}$ est $c_{\mu(j)-1}$ et alors $P_{\theta(j)}$ est dans l'état initial. On a donc, d'après la proposition 3.2, $\Delta_{\theta(j)}\subset\gamma(c_{\mu(j)-1})$ et donc $\Delta_{\theta(j)}=\Delta_{\theta(j)}\cap\gamma(c_{\mu(j)-1})$.

Mais, à cause de la proposition 3.2, la suite $(\gamma(c_i))_{i=0, \ldots, \mu(j)-1}$ vérifie les hypothèses du lemme 3.3 avec $I=I_{\mu(j)-1}=I_{\mu(j-1)}$ et comme $\Delta_{\theta(j)}\subset E_{\theta(j)}$ et que $\theta(j)\notin I_{\mu(j-1)}$:

$$\Delta_{\theta(j)}=\Delta_{\theta(j)}\cap\gamma(c_{\mu(j)-1})=\Delta_{\theta(j)}\cap(\gamma(c_0)\cup\bigcup_{i\in I_{\mu(j-1)}}S_{\theta(i)})$$

Mais la suite $(\Gamma_i)_{i=0, \ldots, j-1}$ vérifie aussi les hypothèses du lemme 3.3 et :

$$\Delta_{\theta(j)}\cap\Gamma_{j-1}=\Delta_{\theta(j)}\cap(\Gamma_0\cup\bigcup_{1\leqq i<j}S_{\theta(i)}).$$

Comme $\Gamma_0=\gamma(c_0)$ par définition, il en résulte que $\Delta_{\theta(j)}\cap\Gamma_{j-1}=\Delta_{\theta(j)}$ et donc $\Delta_{\theta(j)}\subset\Gamma_{j-1}$ et ce pour tout $j\in[k]$. □

4. POINTS FIXES DE RÉSEAUX

Comme chaque processus définit une fonction multivoque sur un espace de mots infinis, on peut associer aussi à un réseau une fonction multivoque.

DÉFINITION 4.1 : Soit un réseau R donné dans la notation abrégée $\eta(\vec{X})\leftarrow\vec{P}(\sigma(\vec{X}))$.

Notons $\mathscr{E}$ l'ensemble des p-uplets de mots $\vec{W} = \langle W_1, \ldots, W_p \rangle$ tels que :

$$\begin{cases} W_i = \Lambda & \text{si } i \notin F, \\ W_i & \text{est un mot infini sur le bon alphabet sinon.} \end{cases}$$

et $\mathscr{S}$ l'ensemble des p-uplets de mots $\vec{U} = \langle U_1, \ldots, U_p \rangle$ tels que :

$$\begin{cases} U_i = \Lambda & \text{si } i \notin S, \\ U_i & \text{est un mot infini sur le bon alphabet sinon.} \end{cases}$$

Au réseau est associée l'application $\hat{R}$ de $\mathscr{E}$ dans $\mathscr{P}(\mathscr{S})$ définie par :

$$\langle U_1, \ldots, U_p \rangle \in \hat{R}(W_1, \ldots, W_p)$$
$$\text{ssi } \forall i \in \{1, \ldots, h\}, \quad \langle U_{\sigma_i}(1), \ldots, U_{\sigma_i}(m_i) \rangle \in \hat{P}_i(\langle W_{\eta_i}(1), \ldots, W_{\eta_i}(n_i) \rangle).$$

DÉFINITION 4.2: Soit $\mathscr{C}$ l'ensemble des n-uplets de mots $\vec{V} = \langle V_1, \ldots, V_p \rangle$ tels que :

si $i \in E$ alors V_i est un mot fini sur le bon alphabet;

si $i \in F - E$ alors V_i est un mot infini sur le bon alphabet;

si $i \notin F$ alors $V_i = \Lambda$.

Les éléments de $\mathscr{C}$ sont ceux qui permettent de définir les configurations initiales (*cf.* définition 3.10). De façon plus précise il existe une bijection entre $\mathscr{C}$ et les configurations initiales : celle qui à $\langle V_1, \ldots, V_p \rangle \in \mathscr{C}$ associe :

$$\langle \Lambda, \Lambda, \ldots, \Lambda; q_1, \ldots, q_k; V_1, \ldots, V_p \rangle.$$

DÉFINITION 4.3 : Pour tous mots $\vec{V}$ et $\vec{W}$ nous définissons $\vec{V} \star \vec{W}$ par :

$$(\vec{V} \star \vec{W}) = \begin{cases} V_i W_i & \text{si } i \in F, \\ \Lambda & \text{sinon.} \end{cases}$$

PROPOSITION 4.1 : *Si $\vec{V} \in \mathscr{C}$ et $\vec{W} \in \mathscr{S}$ alors $\vec{V} \star \vec{W} \in \mathscr{E}$.*

Démonstration : D'après la définition de $\vec{V} \star \vec{W}$ il suffit de montrer que si $i \in F$ alors $V_i . W_i$ est infini. Or soit $i \in E = F \cap S \subset S$ et alors W_i est infini, soit $i \in F - E$ et V_i est infini d'où $V_i . W_i = V_i$ est aussi infini. □

Cette définition de l'opération $\star$ permet de rendre cohérente la définition suivante.

DÉFINITION 4.4 : A tout élément $\vec{V}$ de $\mathscr{C}$ nous associons l'application $\hat{R}_{\vec{V}}$ de $\mathscr{S}$ dans $\mathscr{P}(\mathscr{S})$ définie par $\hat{R}_{\vec{V}}(\vec{W}) = \hat{R}(\vec{V} \star \vec{W})$.

Nous pouvons alors définir aisément les points fixes de $\hat{R}_{\vec{V}}$.

DÉFINITION 4.5 : Un élément $\vec{U}$ de $\mathscr{S}$ est un point fixe de $\hat{R}_{\vec{V}}$ ssi $\vec{U} \in \hat{R}_{\vec{V}}(\vec{U})$. Nous noterons $\mathrm{Pf}(\hat{R}_{\vec{V}})$ l'ensemble des points fixes de $\hat{R}_{\vec{V}}$.

Si R n'est constitué que de processus déterministes alors $\hat{R}_{\vec{V}}$ peut s'identifier à une application partielle de $\mathscr{S}$ dans $\mathscr{S}$ et ses points fixes vérifient $\vec{U} = \hat{R}_{\vec{V}}(\vec{U})$.

D'après la définition d'une évolution infinie acceptable (définition 3.9) le résultat d'une telle évolution est un élément de $\mathscr{S}$.

Nous sommes donc en mesure de comparer les points fixes de $\hat{R}_{\vec{V}}$ et les résultats des évolutions acceptables à partir de $\vec{V}$ ou, autrement dit, la sémantique dénotationnelle et la sémantique opérationnelle du réseau dans la configuration $\vec{V}$.

PROPOSITION 4.2 : *Soit* $c_0, P_{i_j}, c_1, \ldots$ *une évolution infinie équitable à partir de la configuration initiale* c_0, *avec* $c_i = \langle \vec{u}(i); \vec{q}(i); \vec{V}(i) \rangle$ *dont le résultat est* $\vec{U}$. *Alors pour tout* $i \geqq 0$, *il existe* $\vec{X}(i)$ *et* $\vec{w}(i)$ *tels que :*

$$\vec{U} = \vec{u}(i) . \vec{X}(i),$$
$$\vec{V}(0) . \vec{u}(i) = \vec{w}(i) . \vec{V}(i).$$

Démonstration : Le premier point de cette proposition découle immédiatement de la définition de $\vec{U}$. Le second se démontre par récurrence sur i. Il est trivialement vrai pour $i = 0$ en prenant $\vec{W}(i) = \vec{\Lambda}$, puisque $\vec{u}(i) = \vec{\Lambda}$. Supposons qu'il soit vrai pour i. D'après la définition de $\vdash$:

$$\vec{u}(i+1) = \vec{u}(i) . \vec{v}, \qquad \vec{V}(i) = \vec{w} . \vec{W} \qquad \text{et} \qquad \vec{V}(i+1) = \vec{W} . \vec{V}.$$

Comme $\vec{V}(0) . \vec{u}(i) = \vec{w}(i) . \vec{V}(i)$, on obtient, en posant :

$$\vec{w}(i+1) = \vec{w}(i) . \vec{w},$$
$$\vec{V}(0) . \vec{u}(i+1) = \vec{V}(0) . \vec{u}(i) . \vec{v},\ \vec{w}(i) . \vec{V}(i) . \vec{v}$$
$$= \vec{w}(i) . \vec{w} . \vec{W} . \vec{v} = \vec{w}(i+1) . \vec{V}(i+1). \quad \square$$

PROPOSITION 4.3 : *Pour tout* $\vec{V} \in \mathscr{C}$, $\mathrm{Res}(\vec{V}) \subset \mathrm{Pf}(\hat{R}_{\vec{V}})$.

Démonstration : Soit :

$$c_0, P_{i_1}, c_1, P_{i_2}, \ldots \qquad \text{avec} \quad c_j = \langle \vec{u}(j); \vec{q}(j); \vec{V}(j) \rangle,$$

une dérivation infinie équitable à partir de la configuration initiale associée à $\vec{V} = \vec{V}(0)$ et dont le résultat est $\vec{U}$. Posons $\vec{W} = \vec{V} \star \vec{U}$. Nous allons montrer que $\vec{U} \in \hat{R}_{\vec{V}}(\vec{U}) = \hat{R}(\vec{W})$, c'est-à-dire :

$$\langle U_{\sigma_i}(1), \ldots, U_{\sigma_i}(m_i) \rangle \in \hat{P}_i(W_{\eta_i}(1), \ldots, W_{\eta_i}(n_i)) \quad \text{pour tout } i \in [k].$$

Soit i fixé appartenant à $[k]$ et soit $I \subset R$ l'ensemble des entiers l tels que $i_l = i$.

Comme l'évolution est équitable cet ensemble est infini et peut être considéré comme l'image de R par une application h strictement croissante qui est étendue à $\mathbb{N}$ par $h(0)=0$. Posons aussi $\vec{X}(i)$ et $\vec{W}(i)$ comme dans la proposition précédente.

A partir de la configuration initiale $c'_0=\langle \vec{\Lambda}; q(0); U'_1(0), \ldots, U'_{n_i}(0)\rangle$, où $q(0)$ est l'état initial du processus P_i et $U'_s(0)=W_{\eta_i(s)}$ nous allons construire par récurrence une dérivation $c'_0, c'_1, \ldots$ pour le processus P_i, qui vérifie, en posant :

$$c'_j=\langle w'_1(j), \ldots, w'_{m_i}(j); q'(j); U'_1(j), \ldots, U'_{n_i}(j)\rangle$$

($\star$) $h(j)\leqq l<h(j+1)\Rightarrow$

(i) $q'(j)=q_i(l)$;

(ii) $\forall s\in[m_i]$, $w'_s(j)=u_{\sigma_i(1)}(l)$;

(iii) $\forall s\in[n_i]$, $U'_s(j)=V_{\eta_i(s)}(l).X_{\eta_i(s)}(l)$.

Pour $l=0, j=0$; le point (i) est vrai puisque $q'(0)=q_i(0)$ est l'état initial de P_i; $\forall s\in[m_i]$, $w'_s(0)=u_{\sigma_i(s)}(0)=\Lambda$, d'où (ii) est vrai; et par définition de $U'_s(0)$:

$$U'_s(0)=W_{\eta_i(s)}=V_{\eta_i(s)}.U_{\eta_i(s)},$$

mais $U_{\eta_i(s)}=u_{\eta_i(s)}(0).X_{\eta_i(s)}(0)$ et comme $u_{\eta_i(s)}(0)=\Lambda$:

$$U'_s(0)=V_{\eta_i(s)}(0).X_{\eta_i(s)}(0)$$

et donc (iii) est vrai.

Supposons que ($\star$) est vrai pour l et montrons le pour $l+1$; supposons aussi que $h(j)\leqq l<h(j+1)$.

Soit P'_i le processus déclenché pour passer de c_l à c_{l+1}. Par définition de $\vdash_{P'_i}$ on a :

pour tout $s\in[p]$:

$$u_s(l+1)=u_s(l).\bar{v}_s,$$
$$V_s(l)=\bar{w}_s.W_s,$$
$$V_s(l+1)=W_s.\bar{v}_s,$$

avec :

$$\bar{v}_s=\Lambda \quad \text{si} \quad s\notin\sigma_{i'}([m_{i'}])\cap F$$

et :

$$\bar{w}_s=\Lambda \quad \text{si} \quad s\notin\eta_{i'}([n_{i'}]).$$

et :

$$r = \langle \bar{w}_{\eta_{i'}(1)}, \ldots, w_{\eta_{i'}(n_{i'})}, q_i(l) \rangle \rightarrow \langle q_{i'}(l+1), \bar{v}_{\sigma_{i'}(1)}, \ldots, \bar{v}_{\sigma_{i'}(m_{i'})} \rangle$$

est une règle de $P_{i'}$.

On déduit alors aisément de la proposition 4.2 que $\bar{v}_s . X_s(l+1) = X_s(l)$ pour tout $s \in [p]$:

– si $i \neq i'$ alors $i_{l+1} \neq i$ et donc $l+1 < h(j+1)$.

On a aussi :

$$\begin{aligned}
&- q_i(l+1) = q_i(l) = q'(j);\\
&- \bar{v}_{\sigma_i(s)} = \Lambda \quad \text{d'où } u_{\eta_i(s)}(l+1) = u_{\eta_i(s)}(l) = w'_s(j);\\
&- \bar{w}_{\eta_i(s)} = \Lambda \quad \text{d'où } V_{\eta_i(s)}(l+1)\\
&= V_{\eta_i(s)}(l) . v_{\eta_i(s)} \qquad \text{et} \qquad V_{\eta_i(s)}(l+1) . X_{\eta_i(s)}(l+1)\\
&\qquad = V_{\eta_i(s)}(l) . \bar{v}_{\eta_i(s)} . X_{\eta_i(s)}(l+1) = V_{\eta_i(s)}(l) . X_{\eta_i(s)}(l) = U'_s(j).
\end{aligned}$$

– si $i = i'$, alors $h(j+1) = l+1$ et comme r est une règle de P_i, $c'_j \vdash c'_{j+1}$ avec $q'(j+1) = q_i(l+1)$:

$$W'_s(j+1) = w'_s(j) . \bar{v}_{\sigma_i(s)},$$
$$U'_s(j+1) = W_{\eta_i(s)} . X_{\eta_i(s)}(l)$$

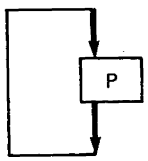
[puisque $U'_s(j) = V_{\eta_i(s)}(l) . X_{\eta_i(s)}(l) = w_{\eta_i(s)} . W_{\eta_i(s)} . X_{\eta_i(s)}(l)$].

On en déduit :

$$\begin{aligned}
&- w'_s(j+1) = u_{\sigma_i(s)}(l) . \bar{v}_{\sigma_i(s)} = u_{\sigma_i(s)}(l+1),\\
&- U'_s(j+1) = W_{\eta_i(s)} . X_{\eta_i(s)}(l)\\
&\quad = W_{\eta_i(s)} . v_{\eta_i(s)} . X_{\eta_i(s)}(l+1) = V_{\eta_i(s)}(l+1) . X_{\eta_i(s)}(l+1). \quad \square
\end{aligned}$$

Par contre tout point fixe de $\hat{R}_{\bar{V}}$ n'est pas forcément le résultat d'une évolution infinie équitable comme le montre l'exemple suivant :

EXEMPLE 4.1 : Soit le réseau représenté par :



et soit P le processus à une entrée et une sortie sur un alphabet Σ défini par :

Soit W un mot infini sur Σ et soit $W(n)$ sa n-ième lettre; l'ensemble des états de P est $\{q_i / i \geqq 0\}$, l'état initial est q_0 et les règles sont $\{\langle q_{i-1}; W(i)\rangle \to \langle W(i); q_i\rangle / \ i>0\}$.

Il est facile de voir que P est déterministe, que son domaine de définition est réduit à $\{W\}$ et que $\hat{P}(W)=W$. Le mot infini W est donc le seul point fixe de $\hat{P}_\Lambda$.

Examinons donc les évolutions infinies à partir de la configuration initiale $\langle \Lambda; q_0; \Lambda\rangle$. Comme le processus P, dans l'état initial q_0, doit lire un symbole sur sa ligne d'entrée et que cette ligne d'entrée est vide il ne peut pas être déclenché. Il n'y a donc pas d'évolution possible à partir de cette configuration initiale et $\mathrm{Res}(\Lambda)=\emptyset$.

On voit sur cet exemple que ce qui empêche le point fixe d'être le résultat d'une évolution infinie c'est l'absence de la condition de déclenchement et en effet on peut démontrer que tout se passe bien si cette condition est vérifiée. Mais auparavant introduisons de nouvelles définitions.

DÉFINITION 4.6 : Soient, pour tout entier $j \geqq 0$, $d(j)$ et $r(j)$ les deux entiers tels que j s'écrit de façon unique $j=k.d(j)+r(j)$ avec $r(j)<k$.

Soit θ une permutation de $[k]$. Une évolution infinie $c_0, P_h(0), c_1, P_{h(1)}, \ldots$ est dite θ-*cyclique* si pour tout $j \geqq 0$, $h(j)-\theta(r(j)+1)$.

Une évolution infinie est dite *cyclique* si elle est θ-cyclique pour une certaine permutation θ.

Nous noterons $\mathrm{Res}_{\theta C}(\vec{V})$ et $\mathrm{Res}_C(\vec{V})$ les ensembles de résultats d'évolutions θ-cycliques et cycliques à partir de $\vec{V}$.

Il est clair que toute évolution θ-cyclique est cyclique et que toute évolution cyclique est équitable, d'où $\mathrm{Res}_{\theta C}(\vec{V}) \subset \mathrm{Res}_C(\vec{V}) \subset \mathrm{Res}(\vec{V})$.

PROPOSITION 4.4 : *Si $\vec{V}$ vérifie la condition de déclenchement alors* $\mathrm{Pf}(\hat{R}_{\vec{V}}) \subset \mathrm{Res}_{\theta C}(\vec{V})$ *pour une permutation* θ.

Démonstration : Supposons que $\vec{V}$ vérifie la condition de déclenchement et soit θ la permutation figurant dans cette condition.

Soit $\vec{U} \in \mathrm{Pf}(\hat{R}_{\vec{V}})$ et soit $\vec{W}=\vec{V}\star\vec{U}$. Pour tout $i \in [k]$ on a $\langle U_{\sigma_i(1)}, \ldots, U_{\sigma_i(m_i)}\rangle \in \hat{P}_i(W_{\eta_i(1)}, \ldots, W_{\eta_i(n_i)})$ et il existe donc une dérivation infinie $c_i(0), \ldots, c_i(j), \ldots$ avec :

$$c_i(j)=\langle u_{\sigma_i(1)}(j), \ldots, u_{\sigma_i(m_i)}(j); q_i(j); W_{\eta_i(1)}(j), \ldots, W_{\eta_i(n_i)}(j)\rangle$$

et donc $u_{\sigma_i(s)}(0)=\Lambda$ et $W_{\eta_i(s)}(0)=W_{\eta_i(s)}$.

Nous allons construire par récurrence une évolution infinie θ-cyclique $c'_0, P_{h(0)}, c'_1, \ldots$ avec :

$$c'_j = \langle u'_1(j), \ldots, u'_p(j); q'_1(j), \ldots, q'_k(j); V_1(j), \ldots, V_p(j) \rangle,$$

telle que pour tout $j \geqq 0$, et en posant :

$$r = r(j), \qquad d = d(j); \qquad T = \theta([r(j)]),$$

on ait :

(i) $$q'_i(j) = \begin{cases} q_i(d+1) & \text{si } i \in T, \\ q_i(d) & \text{sinon;} \end{cases}$$

(ii) $$u'_{\sigma_i(s)}(j) = \begin{cases} u_{\sigma_i(s)}(d+1) & \text{si } i \in T, \\ u_{\sigma_i(s)}(d) & \text{sinon;} \end{cases}$$

(iii) $$\forall l \in [p] \exists w_l(j) \text{ tq } w_l(j).W_l(j) = W_l(0)$$

et

$$w_{\eta_i(s)}(j).V_{\eta_i(s)}(j) = V_{s'}.u'_s(d') \leqq W_{s'}(0)$$

avec :

$$s' = \eta_i(s) \qquad \text{et} \qquad d' = \begin{cases} d+1 & \text{si } i \in T, \\ d & \text{sinon;} \end{cases}$$

(iv) $P_{\theta(r+1)}$ est déclenchable dans la configuration c'_j.

Il est clair que, par construction, le résultat de cette évolution est $\vec{U}$.

Construisons donc c'_j par récurrence sur j.

(A) Pour $j=0$, on a $r=0$, $d=0$, $T=\emptyset$.

On pose :

$$q'_i(0) = q_i(0),$$
$$u'_{\sigma_i(s)}(0) = \Lambda = u_{\sigma_i(s)}(0),$$
$$V_{\eta_i(s)}(0) = V_{\eta_i(s)}.$$

On voit que c'_0 est la configuration initiale associée à $\vec{V}$ et qu'elle vérifie (i)-(ii)-(iii). De plus comme $\vec{V}$ vérifie la condition de déclenchement, $\Delta_{\theta(1)} \subset \Gamma_0 = \gamma(c'_0)$ et donc $P_{\theta(1)}$ est déclenchable dans c'_0 d'où (iv) est aussi vérifiée.

(B) Supposons que c'_j vérifie les conditions (i)-(iv) avec $r=r(j)$, $d=d(j)$ et $T=\theta([r(j)])$, et posons $i=\theta(r+1)$, d'où $i \notin T$.

Par hypothèse $c_i(d) \underset{P_i}{\vdash} c_i(d+1)$; il existe donc une règle :

$$r = \langle q; x_1, \ldots, x_{n_i} \rangle \to \langle v_1, \ldots, v_{m_i}; q' \rangle$$

telle que :

$$q = q_i(d) \quad \text{et} \quad q' = q_i(d+1), \tag{1}$$

$$\forall s \in [n_i] \left\{ \begin{array}{c} x_s \text{ est une lettre si } s \in \delta(q_i(d)), \\ \text{est égal à } \Lambda \text{ sinon,} \end{array} \right\} \tag{2}$$

$$\forall s \in [m_i], \quad u_{\sigma_i(s)}(d+1) = u_{\sigma_i(s)}(d).v_s, \tag{3}$$

$$\forall s \in [n_i], \quad W_{\eta_i(s)}(d) = x_s.W_{\eta_i(s)}(d+1). \tag{4}$$

Mais d'après (iv) P_i est déclenchable dans c'_j; on a donc $\overline{\delta}_i(c'_j) \subset \gamma(c'_j)$ c'est-à-dire :

$$l \in \eta_i(\delta(q'_i(j)) \Rightarrow V_l(j) \neq \Lambda. \tag{5}$$

Comme d'après (i) $q'_i(j) = q_i(d)$ et que d'après (iii) $V_{\eta_i(s)}(j)$ est un préfixe de $W_{\eta_i(s)}(d)$, de (4) et de (5) on déduit :

$$s \in \delta(q_i(d)) \Rightarrow V_{\eta_i(s)}(j) \text{ est un préfixe non vide de } x_s.W_{\eta_i(s)}(d+1). \tag{6}$$

Pour $s \in \delta(q_i(d))$, $V_{\eta_i(s)}(j)$ peut donc s'écrire $x_s.W_{\eta_i(s)}$ et nous posons $\overline{W}_s = V_s(j)$ pour $s \notin \eta_i(\delta(q_i(d)))$.

Nous pouvons alors faire évoluer la configuration c'_j en appliquant la règle r et nous obtenons :

$$c'_{j+1} = \langle \vec{u}'(j+1); \vec{q}'(j+1); \vec{V}(j+1) \rangle,$$

avec :

$$u'_s(j+1) = \left\{ \begin{array}{ll} u'_s(j) & \text{si } s \notin S_i, \\ u'_s(j).v_{s'} & \text{si } s = \sigma_i(s'), \end{array} \right\} \tag{7}$$

$$V_s(j+1) = \left\{ \begin{array}{ll} \overline{W}_s & \text{si } s \notin S_i, \\ \overline{W}_s.v_{s'} & \text{si } s = \sigma_i(s'), \end{array} \right\} \tag{8}$$

$$q'_l(j+1) = \left\{ \begin{array}{ll} q_i(d+1) & \text{si } i = l, \\ q'_l(j) & \text{sinon.} \end{array} \right\} \tag{9}$$

De (9) et (i) on déduit :

$$q_l(j+1) = \left\{ \begin{array}{ll} q_l(d+1) & \text{si } l \in T \cup \{i\}, \\ q_l(d) & \text{sinon.} \end{array} \right\} \tag{10}$$

De (7), (3) et (ii) on déduit :

$$u'_{\sigma_i(s)}(j+1) = \left\{ \begin{array}{ll} u_{\sigma_i(s)}(s+1) & \text{si } l \in T \cup \{i\}, \\ u_{\sigma_i(s)}(d) & \text{sinon.} \end{array} \right\} \tag{11}$$

Et de (8), (3), (4), (iii) et de la définition de $\overline{W}_s$ on tire, en posant :

$$\left.\begin{array}{l} w_s(j+1)=\begin{cases} w_s(j) & \text{si } s\notin F_i, \\ w_s(j).x_{s'} & \text{si } s=\eta_i(s'), \end{cases} \\ w_l(j+1).W_l(j+1)=W_l(s) \\ \text{et} \\ w_{\eta_i(s)}(j+1).V_{\eta_i(s)}(j+1)=V_{s'}.U_{s'}(d'), \\ \text{avec} \\ s'=\eta_l(s) \quad \text{et} \quad d'=\begin{cases} d+1 & \text{si } l\in T\cup\{i\}, \\ d & \text{sinon} \end{cases} \end{array}\right\} \quad (12)$$

On voit alors que soit $T\cup\{i\}=[k]$ et dans ce cas $d\ (j+1)=d+1$ et $r\ (j+1)=0$, soit $T\cup\{i\}\neq[k]$ et $d\ (j+1)=d\ (j)$, $r\ (j+1)=r\ (j)+1$. Dans les deux cas c'_{j+1} vérifie donc (i), (ii), (iii).

Il reste à montrer que c'_{j+1} vérifie aussi (iv).

Dans le cas où $j\geqq k$, ceci découle de la proposition 3.4. Dans le cas où $j<k$ ceci est une conséquence immédiate de la condition de déclenchement, car on montre aisément par récurrence, en utilisant la proposition 3.2, que $\{i\in E/V_i(j)\neq\Lambda\}\supset\Gamma_j$. □

D'après la proposition 3.5 il suffit pour que la condition de déclenchement soit vérifiée par une configuration initiale, qu'il existe une évolution infinie équitable à partir de cette configuration, ou, autrement dit, que $\mathrm{Res}(\vec{V})\neq\emptyset$. Des propositions 4.3 et 4.4 on déduit alors :

THÉORÈME 4.1 : *Si* $\mathrm{Res}(\vec{V})\neq\emptyset$ *alors* $\mathrm{Res}(\vec{V})=\mathrm{Pf}(\hat{R}_{\vec{V}})$.

En d'autres termes les sémantiques opérationnelle et dénotationnelle d'un réseau de processus sont identiques, pourvu que la sémantique opérationnelle existe.

5. LES RÉSEAUX COMME DES PROCESSUS UNIQUES

Kahn [5] a donné beaucoup d'importance au fait que l'on pouvait considérer un réseau de processus comme un processus unique. C'est ce résultat que nous allons démontrer.

Nous avons déjà vu que les lignes d'un réseau pouvaient être partagées en trois :

– celles connectant l'extérieur à des entrées $(F-E)$;

– celles connectant des entrées et des sorties de processus du réseau $(E=F\cap S)$;

– celles connectant des sorties à l'extérieur $(S-E)$.

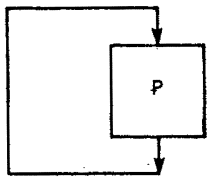
Si l'on veut examiner le fonctionnement global d'un réseau il est logique de considérer que ses lignes d'entrées sont $F-E$, ses lignes de sorties $S-E$.

Nous noterons $\vec{V}_E$ (resp. $\vec{V}_{F-E}$, $\vec{V}_{S-E}$) tout vecteur dont les composantes sont indicées par l'ensemble E (resp. $F-E, S-E$). Comme $[p]$ est la réunion des trois ensembles disjoints $E, F-E, S-E$, tout vecteur $\vec{V}=\langle V_1, \ldots, V_p\rangle$ peut donc s'écrire comme un triplet $\langle \vec{V}_E, \vec{V}_{F-E}, \vec{V}_{S-E}\rangle$. Ainsi, en particulier, pour tout vecteur $\vec{V}$ de $\mathscr{C}$, $\vec{V}_E$ est un vecteur de mots infinis et $\vec{V}_{S-E}$ un vecteur de mots vides. De plus il faut remarquer que la condition de déclenchement ne dépend que de $\vec{V}_E$.

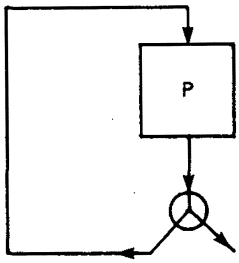
Le comportement global d'un réseau dans la configuration initiale $\vec{V}=\langle \vec{V}_E, \vec{V}_{F-E}, \vec{\Lambda}_{S-E}\rangle$ est la fonction qui à $\vec{V}_{F-E}$ associe $\{\vec{U}_{S-E}/\vec{U}\in \mathrm{Res}(\vec{V})\}$.

Signalons ici que le fait de ne considérer que les sorties de $S-E$ n'est pas réellement une restriction : toute information circulant sur une ligne de sortie de $S\cap E$ peut être considérée comme circulant également sur une ligne de $S-E$ en dédoublant cette ligne grâce à un opérateur de duplication (exemple 2.6), l'introduction d'un tel duplicateur ne changeant en rien les propriétés du réseau.

EXEMPLE 5.1 : Représentons par Ⓐ l'opérateur de duplicateur. L'unique ligne du réseau :



peut être dédoublée en une ligne de sortie interne et une ligne de sortie externe par :



THÉORÈME 5.1 : *Pour tout réseau R et tout vecteur $\vec{v}_E$ de mots finis il existe un processus P à* Card $(F-E)$ *entrées et* Card $(S-E)$ *sorties tel que pour tout vecteur $\vec{V}_{F-E}$ de mots infinis :*

$$\hat{P}(\vec{V}_{F-E}) = \{ \vec{U}_{S-E} / \vec{U} \in \text{Res}\,(\langle \vec{v}_E, \vec{V}_{F-E}, \vec{\Lambda}_{S-E} \rangle) \}.$$

De plus si R ne contient que des processus déterministes (resp. finitaires) alors P est déterministe (resp. finitaire).

Démonstration : Si pour tout $\vec{V}_{F-E}$, Res $(\langle \vec{v}_E, \vec{V}_{F-E}, \vec{\Lambda}_{S-E} \rangle)$ Q est vide, il suffit de prendre comme processus P un processus déterministe sans règles : $\hat{P}$ n'est alors défini nulle part.

S'il existe un $\vec{V}_{F-E}$ tel que Res $(\langle \vec{v}_E, \vec{V}_{F-E}, \vec{\Lambda}_{S-E} \rangle)$ n'est pas vide alors, d'après la proposition 3.5, $\langle \vec{v}_E, \vec{V}_{F-E}, \vec{\Lambda}_{S-E} \rangle$ vérifie la condition de déclenchement et comme cette condition ne dépend que de $\vec{v}_E$, la condition de déclenchement est vérifiée, avec le même θ, pour le vecteur $\langle \vec{v}_E, \vec{V}_{F-E}, \vec{\Lambda}_{S-E} \rangle$, quel que soit $\vec{V}_{F-E}$.

On déduit alors des propositions 4.4 et 4.5 que Res $(\vec{V}) = \text{Res}_{\theta c}(\vec{V})$, pour $\vec{V} \in \mathscr{C}$ tel que $\vec{V}_E = \vec{v}_E$.

Nous construisons le processus P de la façon suivante :

$Q = Q_1 \times \ldots \times Q_k \times \mathscr{V}$, où Q_i est l'ensemble des états de P_i et $\mathscr{V}$ est l'ensemble des vecteurs de mots finis indicés par E.

– l'état initial de P est $\langle q_1(0), \ldots, q_k(0), \vec{v}_E \rangle$;

– pour tout état $\bar{q} = \langle q_1, \ldots, q_k, \vec{w}_E \rangle$ de Q :

$$\delta(\bar{q}) = \bigcup_{i \in [k]} (\eta_i(\delta(q_i)) - E).$$

– Les règles de P seront obtenues en composant une règle de $P_{\theta(1)}$ puis une règle de $P_{\theta(2)}$ et ainsi de suite jusque $P_{\theta(k)}$.

De façon plus précise :

$$\langle \vec{x}_{F-E}; \vec{q}, \vec{w}_E \rangle \to \langle \vec{q}', \vec{w}'_E; \vec{y}_{S-E} \rangle$$

est une règle de P ssi il existe une séquence :

$$\vec{w}_0 = \vec{w}_{E'}, \quad \vec{w}(1), \vec{w}(2), \ldots, \vec{w}(2k-1), \quad \vec{w}(2k) = w'_E$$

d'éléments de $\mathscr{V}$ et pour tout $i \in [k]$ une règle $r_i = \langle \vec{x}\ (\theta(i)); q_{\theta(i)} \rangle \to \langle q'_{\theta(i)}; \vec{y}(\theta(i)) \rangle$ de $P_{\theta(i)}$ tels que :

– pour tout $i \in [k]$:

$$\forall j \in [n_i], \quad \eta_i(j) \in F - E \quad \Rightarrow \quad x_{\eta_i(j)} = x_j(i),$$
$$\forall j \in [m_i], \quad \sigma_i(j) \in S - E \Rightarrow y_{\sigma_i(j)} = y_j(i),$$

$$\forall j \in E, \quad w_j(2i-2) = \begin{cases} w_j(2i-1) & \text{si } j \notin E_{\theta(i)}, \\ x_l(\theta(i)).w_j(2i-1) & \text{si } j = \eta_{\theta(i)}(l), \end{cases}$$

$$\forall j \in E, \quad w_j(2i) = \begin{cases} w_j(2i-1) & \text{si } j \notin S'_{\theta(i)}, \\ w_j(2i-1).y_l(\theta(i)) & \text{si } j = \sigma_{\theta(i)}(l). \end{cases}$$

De la construction des règles de P il résulte que :

$$\langle \vec{U}_{S-E}; \vec{q}, \vec{w}_E; \vec{V}_{F-E} \rangle \underset{P}{\vdash} \langle \vec{U}'_{S-E}; \vec{q}', \vec{w}'_E; \vec{V}'_{F-E} \rangle,$$

ssi il existe $\vec{u}_E$ et $\vec{u}'_E$ tels que :

$$\langle \vec{U}_{S-E}, \vec{u}_E, \vec{\Lambda}_{F-E}; \vec{q}; \vec{V}_{F-E}, \vec{w}_E, \vec{\Lambda}_{S-E} \rangle$$

$$\vdash_{P_{\theta(1)} \cdots P_{\theta(k)}} \langle \vec{U}'_{S-E}, \vec{u}'_E, \vec{\Lambda}_{F-E}; \vec{q}'; \vec{V}'_{F-E}, \vec{w}'_E, \vec{\Lambda}_{S-E} \rangle.$$

La première partie du théorème découle alors du fait que tout résultat peut être obtenu par une évolution θ-cyclique. La seconde partie est immédiate par construction. □

Nous pouvons donc appliquer aux réseaux obtenus sur les processus. En particulier nous obtenons :

PROPOSITION 5.1 : *Soit* R *un réseau de processus déterministes et soit* $\vec{w}_E$ *un vecteur de mots finis. La fonction partielle* $\vec{V}_{F-E} \rightsquigarrow \text{Res}\,(\langle \vec{V}_{F-E}, \vec{v}_E, \vec{\Lambda}_{S-E} \rangle)_{S-E}$ *est univoque et continue et son domaine de définition est fermé.*

Démonstration : D'après le théorème 5.1 il existe un processus déterministe P tel que $\hat{P}\,(\vec{V}_{F-E}) = \text{Res}\,(\langle \vec{V}_{F-E}, \vec{v}_E, \vec{\Lambda}_{S-E} \rangle)_{S-E}$. On applique alors la proposition 2.2. □

On remarquera que par le biais du théorème 5.1 on obtient que le résultat de l'évolution infinie équitable d'un réseau de processus déterministes est unique alors qu'il peut y avoir un nombre infini de telles évolutions.

Pour terminer nous allons caractériser les fonctions qui peuvent être « calculées » par des processus.

PROPOSITION 5.2 : *Soit* f *une application de* $\Sigma_1^\omega \times \ldots \times \Sigma_n^\omega$ *dans* $\mathscr{P}\,(\Sigma_1'^\omega \times \ldots \times \Sigma_m'^\omega)$. *Il existe un processus* P *tel que* $\hat{P} = f$.

Démonstration : Pour tout vecteur $\vec{V}$ de $\Sigma_1^\omega \times \ldots \times \Sigma_n^\omega$ (resp. $\Sigma_1'^\omega \times \ldots \times \Sigma_m'^\omega$) notons $\vec{V}/i/$ le vecteur de $\Sigma_1 \times \ldots \times \Sigma_n$ (resp. $\Sigma'_1 \times \ldots \times \Sigma'_m$) dont chaque composante est la i-ième lettre de chaque composante de $\vec{V}$.

Posons $Q = \{ q^{i}_{\vec{v},\vec{u}} / \vec{u} \in f(\vec{v}), i > 0 \} \cup \{ q_0 \}$ et définissons le processus P dont l'ensemble des états est Q et dont l'état initial est q_0 par l'ensemble de règles :

$$\{ \langle \vec{x}; q_0 \rangle \to \langle q^{1}_{\vec{V},\vec{U}}; \vec{y} \rangle / \vec{x} = \vec{V} / 1 /, \vec{y} = \vec{U} / 1 / \}$$
$$\cup \{ \langle \vec{x}; q^{i}_{\vec{V},\vec{U}} \rangle \to \langle q^{i+1}_{\vec{V},\vec{U}} \quad ; y \rangle / \vec{x} = \vec{V} / i+1 /, \vec{y} = \vec{U} / i+1 /, i > 0 \}.$$

Il est clair qu'il existe une dérivation infinie :

$$c_0 = \langle \Lambda; q_0; \vec{V} \rangle, c_1, \ldots, c_i, \ldots$$

ssi il existe $\vec{U} \in f(\vec{V})$ tel que :

$$\forall i > 0, \quad c_i = \langle \vec{U}(i), q^{i}_{\vec{V},\vec{U}}, \vec{V}(i) \rangle \qquad \text{avec} \quad \vec{u}(i) = \vec{U} / 1 / \ldots \vec{U} / i+1 /$$

et $\vec{V} = \vec{V} / 1 / \ldots \vec{V} / i+1 / \vec{V} / i /$ et donc que $\hat{P}(\vec{V}) = f(\vec{V})$. □

PROPOSITION 5.3 : *Soit f une application partielle de $\Sigma_1^{\omega} \times \ldots \times \Sigma_n^{\omega}$ dans $\Sigma_1'^{\omega} \times \ldots \times \Sigma_m'^{\omega}$ dont le domaine de définition* Df *est fermé et telle que pour $\vec{V}$, $\vec{V}' \in$ Df, $d(f(\vec{V}), f(\vec{V}')) \leqq d(\vec{V}, \vec{V}')$. Alors il existe un processus déterministe P tel que $\hat{P} = f$.*

Démonstration : Soit $\vec{v}$ le préfixe de longueur n d'un élément $\vec{V}$ de Df. Il existe alors $\vec{w}$ tel que $\vec{w}$ est le préfixe de longueur n de $f(\vec{V}')$ pour tout $\vec{V}' \in$ Df ayant $\vec{v}$ comme préfixe. Notons $f(\vec{v})$ le vecteur $\vec{w}$.

Posons $Q = \{ q_{\vec{v}} / \vec{v}$ est le préfixe de longueur n d'un élément de Df$\} \cup \{ q_{\vec{\Lambda}} \}$. Soit P le processus déterministe ayant Q pour ensemble d'états, $q_{\vec{\Lambda}}$ pour état initial et dont les règles sont :

$$\{ \langle \vec{x}; q_{\vec{v}} \rangle \to \langle q_{\vec{v}\vec{x}}; \vec{y} \rangle / f(\vec{v} . \vec{x}) = f(\vec{v}) . \vec{y} \}.$$

Il est clair que si $\vec{V} \in$ Df l'unique dérivation infinie à partir de $\langle \vec{V}; q_{\vec{\Lambda}}; \vec{V} \rangle$ a pour résultat $f(\vec{V})$. Réciproquement soit $c_0, \ldots, c_1, \ldots, c_i, \ldots$ une dérivation infinie à partir de $c_0 = \langle \vec{\Lambda}; q_{\vec{\Lambda}}; \vec{V} \rangle$ avec $c_i = \langle \vec{u}(i); q_{\vec{v}(i)}; \vec{V}(i) \rangle$. Pour tout $i \geqq 0$ on a $\vec{v}(i) . \vec{V}(i) = \vec{V}$; de plus il existe $\vec{W}(i) \in$ Df ayant $\vec{v}(i)$ pour préfixe de longueur i. Le vecteur $\vec{V}$ est donc limite de la suite $(\vec{W}_{(i)})_i$ d'éléments de Df et appartient donc à Df puisque Df est fermé. □

Compte tenu de la proposition 2.2, cette proposition fournit une caractérisation des fonctions calculées par des processus déterministes.

De même les propositions 2.3 et 2.4 ainsi que la proposition suivante fournissent une caractérisation des fonctions calculées par des processus finitaires. Plus précisément il suffit que les points 2.3 (i), 2.4 (i) et 2.4 (ii) soient vérifiés et de fait ce n'est qu'un exercice de topologie de montrer que les points 2.4 (iii) et (2.3) (i) sont des conséquences des précédentes.

PROPOSITION 5.4 : *Soit f une application de $\Sigma_1^\omega \times \ldots \times \Sigma_n^\omega$ dans $\mathscr{P}(\Sigma_1'^\omega \times \ldots \times \Sigma_m'^\omega)$, continue au sens d'Eilenberg-Montgomery, dont le domaine de définition est fermé et telle que l'image de toute boule fermée de rayon 2^{-P} est recouverte par une réunion finie de boules fermées de même rayon. Alors il existe un processus finitaire P tel que $f = \hat{P}$.*

Démonstration : Soit P le processus finitaire dont les états sont tous les $q_{\vec{v}, \vec{w}}$, où $\vec{v}$ est le préfixe d'une certaine longueur d'un élément du domaine de définition de f et où $\vec{w}$ appartient à l'ensemble $\rho(\vec{v})$ défini dans la proposition 2.3. L'état initial de P est $q_{\vec{\Lambda}, \vec{\Lambda}}$ et ses règles sont :

$$\{\langle x; \vec{q}_{\vec{v}, \vec{w}} \rangle \to \langle q_{\vec{v}\vec{x}, \vec{w}.\vec{y}}; \vec{y} \rangle\}.$$

Il est clair que ce processus est finitaire et que si $\vec{W} \in f(\vec{U})$ il existe une dérivation infinie $c_0, c_1, \ldots$ avec :

$$c_i = \langle \vec{w}(i); q_i; \vec{u}(i) \rangle,$$

telle que :

$$\vec{w}(i) = \vec{W}/1/\ldots \vec{W}/i/, \quad q_i = q_{\vec{v}(i), \vec{w}(i)} \qquad \text{et} \qquad \vec{U} = \vec{v}(i).\vec{U}(i),$$

dont le résultat est $\vec{W}$.

Réciproquement, soit $c_0, \ldots, c_i \ldots$ une dérivation infinie avec $c_i = \langle \vec{w}(i); q_i; \vec{U}(i) \rangle$ à partir de la configuration initiale $\langle \vec{\Lambda}; q_{\vec{\Lambda}, \vec{\Lambda}}; \vec{U} \rangle$. On montre aisément par induction que $q_i = q_{\vec{v}(i), \vec{w}(i)}$ avec $\vec{v}(i).\vec{U}(i) = \vec{U}$. Il en résulte d'après la définition de P que pour tout i il existe $\vec{V}(i)$ dans le domaine de définition de f tel que $\vec{v}(i)$ est préfixe de $\vec{V}(i)$ et $\vec{W}(i) \in f(\vec{V}(i))$ tel que $\vec{w}(i)$ est le préfixe de longueur i de $\vec{W}(i)$. Comme la suite $\vec{v}(i)$ est croissante, et que sa borne supérieure est $\vec{U}$, la suite $(\vec{V}(i))_i$ converge vers $\vec{U}$ qui appartient donc au domaine de définition de f. Pour les mêmes raisons la suite $(\vec{W}(i))_i$ converge vers $\vec{W}$, et, comme f est continue, $\vec{W} \in f(\vec{U})$. □

REFERENCES

1. A. ARNOLD et M. NIVAT, *Non Deterministic Recursive Program Schemes*, in Fundamentals of Computation Theory, Poznan, 1977, M. KARPINSKI, éd., Lecture Notes in Computer Science, n° 56, Springer-Verlag, 1977, p. 12-21.
2. A. ARNOLD et M. NIVAT, *The Metric Space of Infinite Trees. Algebraic and Topological Properties*, Rapport I.R.I.A.-Laboria, n° 323, 1978.
3. L. BOASSON et M. NIVAT, *Adherences of Languages*, J. Comput. Sys. Sci. vol. 20, 1980, p. 285-309.
4. S. EILENBERG et D. MONTGOMERY, *Fixed Point Theorems for Multi-Valued Transformations*, Amer. J. Math., vol. 68, 1946, p. 214-222.

5. G. KAHN, *The Semantics of a Simple Language for Parallel Processing*, Proc. I.F.I.P. Congress, 1974.
6. R. M. KELLER, *Denotational Models for Parallel Programs with Indeterminate Operators*, in [11], p. 337-363.
7. R. MILNER, *Processus: a Mathematical Model of Computing Agents*, in Proc. Logic Colloquium, Bristol, North-Holland Pub. Co., 1973, p. 157-173.
8. M. NIVAT, *Mots infinis engendrés par une grammaire algébrique*, R.A.I.R.O., Informatique théorique, vol. 11, 1977, p. 311-327.
9. G. D. PLOTKIN, *A Powerdomain Construction*, S.I.A.M. J. Comp., vol. 5, 1976, p. 452-486.
10. E. WIEDMER, *Exaktes Rechnen mit reellen Zahlen und anderen unendlichen objekten*, Doctoral Dissertation, E.T.H., Zürich, 1977.
11. *Formal Descriptions of Programming Concepts*, E. J. NEUHOLD, éd., North-Holland Pub. Co., 1978.