RAIRO. INFORMATIQUE THÉORIQUE

C. QUEINNEC

Une formalisation des systèmes conversationnels

RAIRO. Informatique théorique, tome 15, n° 4 (1981), p. 303-336

<http://www.numdam.org/item?id=ITA_1981__15_4_303_0>

R A I R O Informatique théorique/Theoretical Informatics
(vol 15, n° 4, 1981, p. 303 à 336)

UNE FORMALISATION DES SYSTÈMES CONVERSATIONNELS (*)

par C QUEINNEC (1)

Communiqué par J -F PERROT

Résumé. — *Cet article présente un aperçu sur certaines techniques nouvelles de définition de systèmes interactifs Ces derniers, formalisés par leur fonction d'évolution (au sens des automates), peuvent également être décrits par des relations d'équivalence qu'ils induisent sur leur langage d'entrée. Un choix restreint de telles relations, portant sur des séquences d'entrées particulièrement représentatives, permet en général d'en reconstituer la totalité Cette technique s'appuie sur la notion de tableau (généralisant les tables d'évolution des automates finis) et sur la possibilité de combiner ces tableaux Nous développons, en fin d'article, un exemple simple mais espéré démonstratif.*

Abstract — *New ways in Interactive Systems definition are presented here Such systems, usually formalized by their evolution function (in the sense of automata) can also be described by the equivalence relations they induce on the input language A partial choice of such relations, involving particularly representative input sequences, is generally sufficient to generate the whole definition This technique uses matrices (generalizing the evolution matrices of finite automatas) and combinations of such matrices An example hoped to be fully illustrative is appended to this paper*

0. INTRODUCTION

L'interactivité s'impose de plus en plus car elle permet de faire dialoguer un utilisateur et un système, chacun concourant dans le domaine qu'il maîtrise le mieux à l'élaboration d'un certain projet. De nombreux systèmes usent de cette faculté, citons seulement :

– DIALOG [4], APL [7], RDS [16] systèmes généraux de programmation;

– CSMP [19], MINNIE [28] systèmes spécialisés dans la construction et la simulation de réseaux;

– [6, 11, 22, 25] présentant d'originales caractéristiques pour éditeurs, traceurs ou autres utilitaires.

Peu d'études ont été menées sur ce sujet et si l'on s'accorde sur la nécessité de l'interactivité, il n'y a pas unisson sur ce que doivent offrir les systèmes conversationnels.

(*) Reçu janvier 1979, révisé janvier 1980
(1) Section d'Étude et de Fabrication des Télécommunications, Issy-les-Moulineaux, France.

R.A.I.R.O. Informatique théorique/Theoretical Informatics, 0399-0540/1981/303/$ 5.00

Les désirs en la matière peuvent grossièrement être répartis en trois classes :

1° L'utilisateur a la maîtrise du système, c'est-à-dire qu'il peut :

– personnaliser son environnement [9],

– interrompre les calculs (touche ATTENTION [19]) et ainsi accéder aux résultats intermédiaires,

– choisir le mode d'évaluation (compilation, interprétation...) [10];

2. Le système doit simplifier au maximum le travail de l'utilisateur :

– en éditant automatiquement et sous bonne forme les résultats [9],

– en proposant un mode calculateur de bureau [20],

– en possédant une fonction HELP permettant d'expliciter le système [19],

– par l'utilisation d'une syntaxe concise, simple et uniforme vis-à-vis des divers composants du système (éditeur, traceur...) [6].

3° Le système est efficace :

– par la rapidité des réponses aux sollicitations simples [9],

– et par la qualité de l'éditeur de texte [3].

Nous partons de l'idée que les utilisateurs ont besoin de descriptions formelles et lisibles des Systèmes Interactifs (SI) qu'ils manipulent (Manuel d'utilisation, fonction HELP...), nous proposons quelques techniques nouvelles autorisant la définition, la description et la formalisation des SI : la définition représente les désidérata du concepteur et constitue les spécifications que doit respecter l'implémenteur.

La définition s'exprime par une description du SI en terme de règles et de tableaux qui permettent la construction du manuel d'utilisation, explicitant les comportements conversationnels et interprétatifs du SI. Enfin, règles et tableaux sont des éléments formels propres à être manipulés.

Les travaux théoriques de Kupka [15, 17, 18] ont mis à jour la notion de Système Formel de Dialogue (FDS) qui capture précisément l'essence des SI en les assimilant à des automates abstraits. Nous renvoyons à Starke [29] pour les éléments de la théorie des automates.

Placer les SI dans ce cadre les rend justiciables de théorèmes généraux (réduction, canonisation...) et permet leur description par le biais de la description de leur fonction d'évolution ([16] est un exemple d'une telle formalisation) ou par la donnée de deux relations d'équivalence particulières :

– une relation d'équivalence entre états;

– et une relation d'équivalence entre sorties.

La théorie que nous présentons au chapitre II repose principalement sur l'utilisation systématique de ces deux importantes relations.

L'exemple suivant use de ces relations dont les définitions formelles et principales propriétés se trouvent au chapitre I.

La touche [C] *(pour « Clear ») figurant habituellement au nombre des touches d'une calculatrice de poche, se formalisera ainsi : pour toute séquence de touches f :*

f [C] *est, au point de vue des états, équivalente à* ε.

En effet, f [C] *(séquence quelconque de touches suivie de l'enfoncement de la touche* [C] *) conduit la calculatrice à l'état initial, état dans lequel est la calculatrice avant que l'on appuie sur une quelconque touche. Cet état est représenté par la séquence vide* ε.

Habituellement la calculatrice affiche le nombre zéro aussi bien après l'enfoncement de la seule touche [0] *qu'après l'enfoncement successif des touches composant la séquence f* [C]. *Nous dirons que f* [C] *est, au point de vue des sorties, équivalente à* [0].

Ces relations ci-dessus formalisent complètement le comportement de la touche [C] mais ne décrivent qu'incomplètement la calculatrice entière.

Un Système de Formalisation Partielle de Dialogue (SFPD) est un ensemble générateur de propositions concernant les équivalences d'états, les équivalences de sortie et les messages pouvant être émis. La cohérence d'un SFPD garantit l'existence d'au moins un FDS dont il est description fragmentaire. En II.1 est donné un critère de cohérence ainsi qu'une procédure dérivée permettant de tester celle-ci. La précédente formalisation de la touche [C] est un exemple de SFPD manifestement cohérent.

Les SFPD cohérents, définissant partiellement des FDS peuvent être combinés par deux opérateurs, l'union et le produit. Ces opérateurs, définis en II.3, permettent ainsi de construire des SFPD complexes.

Enfin un procédé de représentation visuelle des SFPD est introduit, il s'agit des tableaux. Ceux-ci généralisent les tables d'évolution des automates finis et mêlent définitions par équivalence et par fonctions d'évolution.

La théorie ici présentée offre un cadre formel aux notions d'équivalences et de cohérence. Pour définir un SI, on peut proposer un ensemble (dont on peut vérifier la cohérence) de SFPD combinés par les opérateurs sus mentionnés.

Chaque SFPD formalise une partie du SI initial. Les règles d'équivalence (comme, par exemple, celle citée plus haut : (pour toute séquence *f*, *f* [C] équivalent à ε) apparaissent comme des exemples de manipulation du SI et simplifient l'écriture du manuel d'utilisation. On en jugera sur l'exemple final.

Le chapitre I, introduit les notions théoriques nécessaires au développement de la théorie des SFPD exposée au chapitre II.

Des exemples simples illustreront divers points de ces deux chapitres. Un

exemple plus complet et mettant en jeu les techniques exposées est entièrement traité au chapitre III.

I. RAPPELS ET DÉFINITIONS

Ce chapitre expose les principaux résultats obtenus concernant les Systèmes Formels de Dialogue (qu'à la suite de Kupka [15] nous noterons FDS). Ce chapitre se clora sur la notion de FDS partiel.

Un FDS est un automate abstrait de Mealy, discret, déterministe et séquentiel [29] où les ensembles d'entrées (ou questions), de sorties (ou réponses) et d'états peuvent être de cardinalité infinie.

DÉFINITION I.1 : *Un FDS est représenté par un quintuplet* $(\mathscr{I}, \mathscr{O}, \Sigma, \Psi, s_0)$ *avec* $\mathscr{I}$ *ensemble des entrées;* $\mathscr{O}$ *ensemble des sorties;* Σ *espace d'état;* Ψ *fonction d'évolution :* $\mathscr{I} \times \Sigma \to \Sigma \times \mathscr{O}$; $s_0 \in \Sigma$ *état initial du FDS.*

On note $\mathscr{X}^*$ le monoïde librement engendré par les éléments de l'ensemble $\mathscr{X}$, considérés comme générateurs indépendants. ε est l'élément neutre (la séquence vide) de $\mathscr{X}^*$.

La concaténation sera exprimée en accolant les termes à concaténer.

On pose $\mathscr{X}^+ = \mathscr{X}^* - \{\varepsilon\}$. Pour toute séquence f de $\mathscr{X}^*$, $|f|$ représente la longueur de f, c'est-à-dire le nombre de symboles de $\mathscr{X}$ la composant.

Les définitions et résultats suivants découlent de techniques classiques [12, 29].

La fonction d'évolution Ψ est naturellement étendue en :

$$\Psi^* : \quad \mathscr{I}^* \times \Sigma \to \Sigma \times \mathscr{O}^*.$$

On note σ (resp. τ) la projection de Ψ (ou Ψ^*) sur Σ (resp. $\mathscr{O}$ ou $\mathscr{O}^*$). Ainsi donc, si f est une séquence d'entrées ($f \in \mathscr{I}^*$), $\sigma(f, s_0)$ est l'état résultant du FDS et $\tau(f, s_0)$ est la suite des sorties associées aux entrées de f.

DÉFINITION 1.2 : *On introduit la relation d'équivalence forte des états* (S) :

$$\forall\ f, g \in \mathscr{I}^*, \qquad f S g \quad \Leftrightarrow \quad \sigma(f, s_0) = \sigma(g, s_0)$$

et la relation d'équivalence des sorties finales (M) :

$$\forall\ a, b \in \mathscr{I},\ \forall\ f, g \in \mathscr{I}^*,$$
$$fa\, M\, gb \quad \Leftrightarrow \quad \tau(a, \sigma(f, s_0)) = \tau(b, \sigma(g, s_0)).$$

Ces deux relations vérifient :

$$\forall\ f, g \in \mathscr{I}^*, \qquad f S g \quad \Rightarrow \forall\ a \in \mathscr{I}, (fa\, S\, ga) \wedge (fa\, M\, ga)). \tag{1}$$

Cette importante relation sera systématiquement utilisée lors de la génération de FDS à partir de SFPD (*voir* II.1).

A partir de tout FDS $\mathscr{D}$, on peut construire le FDS canonique :

$$\mathscr{D}_c = (\mathscr{I}, \mathscr{I}^+/M, \mathscr{I}^*/S, \Psi_c, S(\varepsilon)),$$

avec $S(f)$ [resp. $M(f)$] la classe d'équivalence de f pour S (resp. M) et Ψ_c définie sans contradiction par :

$$\forall\ f \in \mathscr{I}^*,\ \forall\ a \in \mathscr{I}, \qquad \Psi_c(a, S(f)) = (S(fa), M(fa)),$$

$\mathscr{D}_c$ est Σ-$\mathscr{O}$-isomorphe à $\mathscr{D}$, c'est-à-dire $\mathscr{D}$ et $\mathscr{D}_c$ sont similaires à une redénomination près des états et des messages de sortie.

Un FDS $\mathscr{D}$ (et tous ses Σ-isomorphes) est alors plus aisément caractérisé par le quintuplet $(\mathscr{I}, \mathscr{O}, S, M, \text{msg})$ où S et M sont des relations d'équivalence sur $\mathscr{I}^*$ et $\mathscr{I}^+$ vérifiant (1).

msg : $\mathscr{I}^+/M \to \mathscr{O}$ (injective) est la fonction des messages permettant d'associer un message de sortie à une classe d'équivalence de séquences d'entrées non vides.

Outre la possibilité de définir pratiquement des FDS par ce quintuplet, un autre avantage de l'emploi des relations S et M réside en l'utilisation du treillis complet $\mathscr{T}$ des couples de relations d'équivalence sur $\mathscr{I}^*$ et $\mathscr{I}^+$, liées par (1).

Deux FDS, ayant même ensemble d'entrées $\mathscr{I}$, sont *équivalents* s'ils ont même relation d'équivalence des sorties finales et même fonction des messages.

La complétude de $\mathscr{T}$ permet d'assurer l'existence pour tout FDS $\mathscr{D}$ d'un FDS $\mathscr{D}_r = (\mathscr{I}, \mathscr{O}, S_r, M, \text{msg})$, équivalent à $\mathscr{D}$, unique à un Σ-isomorphisme près et *réduit*, c'est-à-dire tel que S_r est la plus grossière relation d'équivalence vérifiant $(S_r, M) \in \mathscr{T}$. S_r peut être directement caractérisée par :

$$\forall f, g \in \mathscr{I}^*, \qquad (\forall h \in \mathscr{I}^+, fh\, M\, gh) \ \Rightarrow\ f S_r g.$$

Le quadruplet $(\mathscr{I}, \mathscr{O}, M, \text{msg})$ est donc la représentation minimale suffisante pour décrire un FDS.

On trouvera dans Kupka [15] un exposé plus complet de ces propositions.

Avant d'aborder le chapitre II, nous prolongeons la théorie des FDS suivant deux voies qui conduisent aux notions de tableaux et de SFPD et qui introduisent les problèmes que ces notions résoudront : il s'agit des automates sous-jacents et des FDS partiels.

I.1. Automates finis sous-jacents

Si S et M sont des relations finies (il n'existe qu'un nombre fini de classes d'équivalence) alors on associe aisément au FDS, l'automate fini dit « sous-jacent », c'est-à-dire le FDS :

$$(\{T_{ijk} \mid T_{ijk} = \{a \in \mathscr{I} \mid \forall f \in S_i, fa \in Sj \cup Mk\}\}, \mathscr{O}, S, M, \text{msg})$$

où les S_i (res. M_j) sont les classes d'équivalence pour S (resp. M). Les T_{ijk} représentent une classification des entrées du FDS.

Pour un FDS quelconque ($\mathscr{I}$, $\mathscr{O}$, S, M, msg) l'existence d'un automate fini sous-jacent est subordonnée à l'existence d'un couple de relations finies (S_f, M_f) et plus grossier que (S, M). Il n'existe pour un FDS quelconque $\mathscr{D}$ qu'un unique automate $\mathscr{A}$ fini et réduit sous-jacent et optimal (tout autre automate fini réduit et sous-jacent à $\mathscr{D}$ est sous-jacent à $\mathscr{A}$), mais ce dernier n'est pas nécessairement non trivial, c'est-à-dire tel que son ensemble d'entrées et son espace d'états ne soient pas simultanément réduits à un seul élément.

La trivialité de l'automate fini sous-jacent réduit optimal constitue un critère séparant les systèmes « mathématiques » (LISP, APL...) où les entrées peuvent être éminemment complexes et où l'automate fini sous-jacent est trivial, des systèmes « à désignations » (systèmes graphiques...) où les entrées à contenu informatif faible doivent être accumulées pour former une demande complète (par exemple désignation d'une commande sur un menu, remplissage de grilles d'arguments...) [2].

On trouvera dans [23] des exemples de ces propositions ainsi que d'autres considérations utilisant le treillis $\mathscr{T}$, notamment les puissances comparées des FDS et la réduction de l'espace des messages.

I.2. FDS Partiels

Un grand nombre de SI usent de « menus » : l'utilisateur lance un traitement en désignant (à l'aide, par exemple d'un photostyle) un des libellés qui lui sont proposés sur son écran.

Ces systèmes sont au nombre des SI où l'ensemble des entrées permises en un état dépend de cet état. Les FDS partiels offrent un cadre formel pour ces systèmes.

DÉFINITION 1.3 : *Un* FDS *partiel est un quintuplet* ($\mathscr{I}$, $\mathscr{O}$, S, M, msg) *avec :*

$$\text{Dom}(S) \subset \mathscr{I}^*,$$

et :

$$\forall\ f \in \text{Dom}(S),\ \forall\ g, h \in \mathscr{I}^*, \quad f = gh \ \Rightarrow\ g \in \text{Dom}(S)$$

$$\forall\ f, g \in \text{Dom}(S),\ \forall\ a \in \mathscr{I},$$

$$(f\,S\,g \wedge ga \in \text{Dom}(S)) \ \Rightarrow\ fa \in \text{Dom}(S)$$

et :

$$\mathrm{Dom}(M) = \mathrm{Dom}(S) \cap \mathscr{I}^+$$

$$\forall\ f, g \in \mathrm{Dom}(S),$$

$$f\,S\,g \ \Rightarrow\ (\forall a \in \mathscr{I}_{S(f)}, fa\,S\,ga \wedge fa\,M\,ga)$$

où $\mathscr{I}_{S(f)}$ est l'ensemble des entrées permises pour les états de $S(f)$:

$$\mathscr{I}_{S(f)} = \{ a \in \mathscr{I} \mid \forall g \in S(f),\ ga \in \mathrm{Dom}(S) \}.$$

Remarque : Les FDS partiels sont très étroitement liés aux FDS que dès lors, l'on qualifiera, par opposition, de totaux.

En effet à tout FDS partiel, on peut associer le FDS total obtenu en imposant que toute entrée non permise pour la FDS partiel ne fait pas évoluer le FDS total, et s'attire une réponse unique : le message « entrée non permise ».

Plus formellement, le FDS total $(\mathscr{I}, \mathscr{O}', S', M', \mathrm{msg}')$ associé est défini par :

$$\mathscr{O}' = \mathscr{O} \cup \{ \text{« entrée non permise »} \} \text{ avec « entrée non permise »} \notin \mathscr{O},$$

$$\forall\ f, g \in \mathscr{I}^*, \quad (f\,S'\,g) \ \Leftrightarrow\ (f, g \in \mathrm{Dom}(S), f\,S\,g)$$
$$\vee\ (\exists f', g' \in \mathrm{Dom}(S), \exists f'', g'' \in \mathscr{I}^+ - \mathscr{I}_{S(f')}, f = f'f'', g = g'g'', f'\,S\,g')$$

$$\forall\ f, g \in \mathscr{I}^+, \quad (f\,M'\,g) \ \Leftrightarrow\ (f, g \in \mathrm{Dom}(M), f\,M\,g) \vee (f, g \notin \mathrm{Dom}(M))$$

$$\forall\ f \in \mathscr{I}^+, \quad \mathrm{msg}'(M'(f)) = \textit{Si } f \in \mathrm{Dom}(M)$$
$$\textit{alors } \mathrm{msg}\,(M(f))$$
$$\textit{sinon } \text{« entrée non permise »}.$$

I.3. Conclusion et exemple

Ces deux prolongements théoriques directement liés aux FDS, ont été développés ici afin de mettre en évidence les deux points suivants, préliminaires au chapitre II.

1° Le FDS partiel permet la description des SI où l'ensemble d'entrées permises en un état dépend de cet état. Il peut aussi être vu comme une description lacunaire d'un FDS plus complexe (d'où le SFPD).

2° L'existence d'un automate fini sous-jacent à un FDS (on étend sans difficulté cette notion aux FDS partiels) rend plus aisée la description du FDS. En effet les techniques de représentation d'automates finis peuvent être mises à contribution pour représenter grossièrement le FDS par le biais de son automate fini sous-jacent. Les transitions de ce dernier classent les compléments de description nécessaires à la définition du FDS initial (ce dernier point est à l'origine des tableaux).

Le court exemple qui suit justifie entre autres ces remarques.

Exemple I.1 : Le FDS (noté PS 1) que nous allons construire dispose de deux variables **x** et **y** (initialement indéfinies) prenant leur valeur dans $\mathbb{N}$ (ensemble des entiers naturels). On peut modifier la valeur courante de ces variables par les entrées :

$$\mathbf{x} = \langle \text{entier} \rangle \qquad \text{ou} \qquad \mathbf{y} = \langle \text{entier} \rangle$$

A tout moment il est possible de connaître leur valeur courante par les entrées :

$$\mathbf{x}\ ? \qquad \text{ou} \qquad \mathbf{y}\ ?$$

La définition formelle du FDS total PS 1 use des notations suivantes :

un symbole syntaxique indexé représente une réalisation particulière de ce symbole. Il joue ainsi le rôle d'un paramètre;

si $\langle \text{entier} \rangle_1$ est un tel symbole, « $f \in \langle \text{question} \rangle^* \mathbf{x} = \langle \text{entier} \rangle_1$ signifiera que f est une séquence d'entrées où toutes les entrées autres que la dernière sont des questions quelconques et non liées entre elles, la dernière entrée étant « $\mathbf{x} = \langle \text{entier} \rangle_1$ ».

Les éléments de $\mathscr{I}$ sont engendrés par la grammaire suivante :

$$\langle \text{entrée} \rangle ::= \langle \text{question} \rangle \mid \langle \text{assignation-x} \rangle \mid \langle \text{assignation-y} \rangle;$$

$$\langle \text{question} \rangle ::= \mathbf{x}\ ? \mid \mathbf{y}\ ?;$$

$$\langle \text{assignation-x} \rangle ::= \mathbf{x} = \langle \text{entier} \rangle;$$

$$\langle \text{assignation-y} \rangle ::= \mathbf{y} = \langle \text{entier} \rangle;$$

$$\langle \text{entier} \rangle ::= \langle \text{chiffre non nul} \rangle \langle \text{entier} \rangle \mid \langle \text{chiffre} \rangle;$$

$$\langle \text{chiffre} \rangle ::= \mathbf{0} \mid \langle \text{chiffre non nul} : \rangle;$$

$$\langle \text{chiffre non nul} \rangle ::= \mathbf{1} \mid \mathbf{2} \mid \mathbf{3} \mid \mathbf{4} \mid \mathbf{5} \mid \mathbf{6} \mid \mathbf{7} \mid \mathbf{8} \mid \mathbf{9}.$$

L'ensemble des messages de sortie est défini par $\mathscr{O} = \mathbb{N} \cup \{ \text{« **indéfini** »} \}$.

Définition de la relation d'équivalence d'état (S) :

$$\forall f, g \in \mathscr{I}^*, \quad f S g \Leftrightarrow (f, g \in \langle \text{question} \rangle^*)$$

$$\vee (\exists \langle \text{assignation-x} \rangle_1, \qquad (a)$$

$$f, g \in \{ \langle \text{assignation-}x \rangle, \langle \text{question} \rangle \}^* \langle \text{assignation-x} \rangle_1 \langle \text{question} \rangle^*)$$

$$\vee (\exists \langle \text{assignation-y} \rangle_1, \qquad (b)$$

$$f, g \in \{ \langle \text{assignation-y} \rangle, \langle \text{question} \rangle \}^* \langle \text{assignation-y} \rangle_1 \langle \text{question} \rangle^*)$$

$$\vee (\exists \langle \text{assignation-x} \rangle_1, \exists \langle \text{assignation-y} \rangle_1, \qquad (c)$$

$$(f.\ g \in \langle \text{entrée} \rangle^* \langle \text{assignation-x} \rangle_1 \{ \langle \text{question} \rangle, \langle \text{assignation-y} \rangle \}^* \langle \text{assignation-y} \rangle_1 \langle \text{question} \rangle^*)$$
$$\vee (f, g \in \langle \text{entrée} \rangle^* \langle \text{assignation-}y \rangle_1 \{ \langle \text{question} \rangle, \langle \text{assignation-}x \rangle \}^* \langle \text{assignation-x} \rangle_1 \langle \text{question} \rangle^*)) \quad (d).$$

Dans (*a*) ni **x**, ni **y** ne sont définis.

Dans (*b*) (resp. (*c*)) **x** (resp. **y**) est seul défini.

Dans (*d*) **x** et **y** sont définis (la dernière variable modifiée pouvant être **x** ou **y**).

D'après cette définition de la relation d'équivalence d'état, on a notamment :

$$\mathbf{x}\ ?\ \ S\ \ \mathbf{x}\ ?\ \ \mathbf{x}\ ?\ \ \mathbf{y}\ ?\ \ S\ \ \varepsilon,$$
$$\mathbf{x}\ ?\ \ \mathbf{x}=\mathbf{1}\ \ \mathbf{y}=\mathbf{2}\ \ S\ \ \mathbf{y}=\mathbf{2}\ \ \mathbf{y}\ ?\ \ \mathbf{x}=\mathbf{2}\ \ \mathbf{x}=\mathbf{1}\ \ \mathbf{x}\ ?,$$
$$\mathbf{x}=\mathbf{3}\ \ \mathbf{x}=\mathbf{4}\ \ \mathbf{x}=\mathbf{5}\ \ S\ \ \mathbf{x}=\mathbf{5}\ \ \mathbf{x}\ ?$$

Définition de la relation d'équivalence des sorties finales (et des messages associés) (*M*) :

$$\forall f, g \in \mathscr{S}^+, \quad f\,M\,g \Leftrightarrow (f, g \in \{ \langle \text{question} \rangle, \langle \text{assignation-x} \rangle \}^* \mathbf{y}\ ?$$
$$\mathbf{V} f, g \in \{ \langle \text{question} \rangle, \langle \text{assignation-y} \rangle \}^* \mathbf{x}\ ?).$$

/★ le message associé est « **indéfini** » ★/.

$$\vee\ (\exists \langle \text{entier} \rangle_1,$$
$$f, g \in \langle \text{entrée} \rangle^* \mathbf{x} = \langle \text{entier} \rangle_1 \{ \langle \text{question} \rangle, \langle \text{assignation-y} \rangle \}^* \mathbf{x}\ ?$$
$$\vee\ f, g \in \langle \text{entrée} \rangle^*\ \mathbf{x} = \langle \text{entier} \rangle_1 \ \vee f, g \in \langle \text{entrée} \rangle^* \mathbf{y} = \langle \text{entier} \rangle_1$$
$$\vee\ f, g \in \langle \text{entrée} \rangle^*\ \mathbf{y} = \langle \text{entier} \rangle_1 \{ \langle \text{question} \rangle, \langle \text{assignation-x} \rangle \}^* \mathbf{y}\ ?).$$

/★ Le message associé est l'entier naturel représenté par $\langle \text{entier} \rangle_1$ ★/.

Ainsi donc :

$$\mathbf{x}\ ?\ \ \mathbf{x}\ ?\ \ \mathbf{y}\ ?\ \ M\ \ \mathbf{y}=\mathbf{3}\ \ \mathbf{x}\ ?$$
$$\mathbf{x}=\mathbf{1}\ \ \mathbf{x}=\mathbf{2}\ \ \mathbf{y}=\mathbf{3}\ \ M\ \ \mathbf{x}\ ?\ \ \mathbf{x}=\mathbf{3}\ \ \mathbf{y}\ ?\ \ \mathbf{x}\ ?$$

On pourra vérifier que la définition ici donnée est celle d'un FDS.

On peut formuler plusieurs remarques quant à cette définition :

1° Certains points non mentionnés dans la définition informelle sont aisément déductibles de la formalisation. En particulier le message de sortie associé à une assignation est la valeur de l'entier figurant dans l'assignation.

2° L'ensemble des entrées possibles de ce FDS est inclus dans $\{$ **0, 1, 2, 3, 4, 5, 6, 7, 8, 9, x, y,** =, ? $\}^*$, on peut donc le considérer comme un FDS partiel sur ce

dernier ensemble d'entrées, auquel cas on peut lui associer un FDS total par la méthode exposée plus haut. Le FDS total auquel on aboutit n'est pas le seul FDS qu'on peut associer, il serait aussi simple de modifier la définition pour faire en sorte par exemple que toute entrée, non reconnaissable en une question ou une assignation, s'attire le message « **erreur syntaxe** » et ramène à l'état initial.

Le FDS ici décrit correspond aux désirs d'un concepteur, les compléments de définition mentionnés plus haut seront laissés (par exemple) au libre choix de l'implémenteur. Dans les deux cas la définition finale du FDS implémenté permet à l'utilisateur de connaître les choix retenus.

Si l'on se reporte à la définition de la relation d'équivalence d'état (S), l'on verra qu'usage a été fait de l'automate fini sous-jacent suivant :

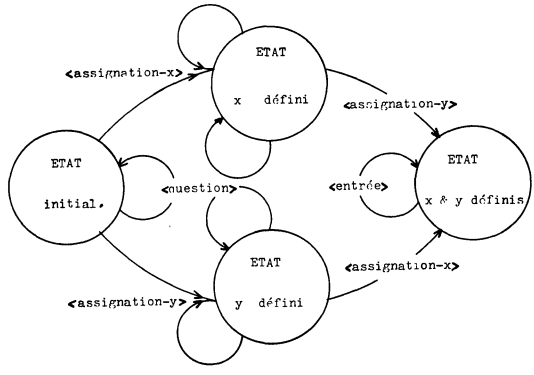

Figure 1. – Automate sous-jacent de PS 1

A chaque état correspond une des clauses de la définition de S; nous avons utilisé les entrées de l'automate fini en donnant la grammaire de l'ensemble des entrées du FDS.

L'utilisation de cet automate fini sous-jacent sera présentée au chapitre II dans le cadre plus large des SFPD.

II. SYSTÈMES DE FORMALISATION PARTIELLE DE DIALOGUES

Le chapitre II introduit successivement, les systèmes de formalisation partielle de dialogues (SFPD), la cohérence des SFPD, ainsi que deux opérateurs permettant de les combiner. Le chapitre se termine sur la notion de tableau. Tous ces concepts concourent à améliorer les techniques pratiques de définition de SI.

DÉFINITION II.1 : *Un* SFPD *est un quintuplet* $(\mathscr{I}, \mathscr{O}, S, M, \text{msg})$ *où* $\mathscr{I}$ *et* $\mathscr{O}$ *sont les ensembles d'entrées et de sorties,* S *(resp.* M*) est une relation d'équivalence de domaine* Dom (S) *[resp.* dom (M)*] sous ensemble de* $\mathscr{I}^*$ *(resp.* $\mathscr{I}^+$*),* msg *est la fonction injective partielle des messages* msg : Dom $(M)/M \to \mathscr{O}$.

Un SFPD apparaît comme un ensemble de propositions décomposables en trois groupes :

– les équivalences d'état;

– les équivalences de sortie finale; deux séquences sont réputées posséder le même message, associé à leur dernière transition (le message n'est pas spécifié);

– les associations à des messages, a une classe d'équivalence de sortie finale est associé un message déterminé.

Exemple II.1 : Le SFPD suivant (noté PS 2) use des notations employées dans l'exemple du chapitre I (PS 1). Les ensembles d'entrées $(\mathscr{I})$ et de sorties $(\mathscr{O})$ restent les mêmes :

notation supplémentaire :

$$\langle \text{variable} \rangle ::= \mathbf{x} \mid \mathbf{y},$$

ainsi donc :

$$\langle \text{question} \rangle ::= \langle \text{variable} \rangle ?,$$

et :

$$\langle \text{assignation} \rangle ::= \langle \text{assignation-x} \rangle \mid \langle \text{assignation-y} \rangle,$$
$$::= \langle \text{variable} \rangle = \langle \text{entier} \rangle.$$

équivalence d'état :

$$\langle \text{variable} \rangle_1 = \langle \text{entier} \rangle_1 \quad \langle \text{variable} \rangle_1 = \langle \text{entier} \rangle_2 S$$
$$\langle \text{variable} \rangle_1 = \langle \text{entier} \rangle_2$$
$$\langle \text{assignation-x} \rangle_1 \langle \text{assignation-y} \rangle_1 \, S \, \langle \text{assignation-y} \rangle_1 \langle \text{assignation-x} \rangle_1$$

équivalence des sorties finales :

$$\forall \ f, g \in \{ \langle \text{entrée} \rangle \}^*, \quad f\,\mathbf{x} = \langle \text{entier} \rangle \, M \, g\,\mathbf{y} = \langle \text{entier} \rangle,$$

messages de sortie :

$$\text{msg}(M(\mathbf{x} = \langle \text{entier} \rangle)) = \langle \text{entier} \rangle.$$

PS 2 apparaît nettement comme un « sous ensemble » du FDS PS 1.

PROPOSITION II.1 : *Tout* FDS *partiel ou total est un* SFPD.

Cette proposition découle directement des définitions.

En guise de réciproque nous énoncerons.

DÉFINITION II.2. *Un* SFPD $(\mathscr{I}, \mathscr{O}, S, M, \text{msg})$ *est* cohérent *si et seulement s'il existe au moins un* FDS *total* $(\mathscr{I}', \mathscr{O}', S', M', \text{msg}')$ *tel que :*

$$\forall\ f, g \in \text{Dom}(S), \quad f\,S\,g \Rightarrow f\,S'\,g,$$

$$\forall\ f, g \in \text{Dom}(M), \quad f\,M\,g \Rightarrow f\,M'\,g,$$

$$\forall\ f \in \text{Dom}(M \circ \text{msg}), \quad \text{msg}(M(f)) = \text{msg}'(M'(f)).$$

On dira alors que $(\mathscr{I}, \mathscr{O}, S, M, \text{msg})$ est une description fragmentaire de $(\mathscr{I}', \mathscr{O}', S', M', \text{msg}')$.

Un SFPD n'est, en général, pas cohérent.

Par exemple s'il existe $f, g \in \text{Dom}(S)$ et $a \in \mathscr{I}$ tels que :

$$f\,S\,g \quad \text{et} \quad \text{msg}(M(fa)) \neq \text{msg}(M(ga)).$$

S'assurer de la cohérence d'un SFPD est le problème primordial, car de là dépend la non vacuité des travaux reposant sur ce SFPD.

Nous appuyant sur le lemme technique suivant, nous proposerons une procédure permettant de tester la cohérence d'un SFPD ou d'en exhiber éventuellement les incohérences. Suivant les cas où cette procédure devra s'appliquer, celle-ci pourra se simplifier.

II.1. Extension de SFPD

L'extension d'un SFPD consiste à construire l'ensemble des propositions dérivables du SFPD initial, par application de la relation (1).

LEMME II.2 : *Si* (S, M) *est un couple de relations d'équivalence partielles sur les domaines respectifs* $\text{Dom}(S)$ *et* $\text{Dom}(M)$, *alors il existe un couple unique* (S_α, M_α) *vérifiant :*

$$\text{Dom}(S_\alpha) = \mathscr{I}^*,$$

$$\text{Dom}(M_\alpha) = \mathscr{I}^+,$$

$$\forall\ f, g \in \text{Dom}(S), \quad f\,S\,g \Rightarrow f\,S_\alpha\,g,$$

$$\forall\ f, g \in \text{Dom}(M), \quad f\,M\,g \Rightarrow f\,M_\alpha\,g,$$

(S_α, M_α) *vérifie* (1)

et pour tout autre couple (S', M') *vérifiant les cinq conditions précédentes :*

$$\forall\ f, g \in \mathscr{I}^*, \quad f\,S_\alpha\,g \Rightarrow f\,S'\,g,$$

$$\forall\ f, g \in \mathscr{I}^+, \quad f\,M_\alpha\,g \Rightarrow f\,M'\,g.$$

Démonstration : Définissons S_α comme la clôture transitive de la relation $\overline{S}$ définie par :

$$\forall\ f, g \in \mathscr{I}^*, \quad f \overline{S} g \ \Leftrightarrow\ (f = g) \vee (\exists h \in \mathscr{I}^*, \exists f', g' \in \mathscr{I}^*, f = f'h, g = g'h, f' S g'),$$

définissons M_α comme la clôture transitive de la relation $\overline{M}$ définie par :

$$\forall\ f, g \in \mathscr{I}^+, \quad f \overline{M} g \ \Leftrightarrow\ (f M g) \vee (\exists h \in \mathscr{I}^+, \exists f', g' \in \mathscr{I}^*, f = f'h, g = g'h, f' S g'),$$

(S_α, M_α) vérifient par construction et de façon évidente les cinq premières propositions du lemme. La sixième propriété se démontre ainsi :

$$\forall\ f, g \in \mathscr{I}^*, \quad f \overline{S} g \ \Rightarrow\ (f = g) \vee (\exists h \in \mathscr{I}^*, \exists f', g' \in \mathscr{I}^*, f = f'h, g = g'h, f' S g') \ \Rightarrow\ (f S' g) \vee (\exists h \in \mathscr{I}^*, \exists f', g' \in \mathscr{I}^*, f = f'h, g = g'h, f' S' g') \ \Rightarrow\ f S' g.$$

La propriété se conserve par passage à la clôture transitive et donc :

$$\forall\ f, g \in \mathscr{I}^*, \quad f S_\alpha g \ \Rightarrow\ f S' g.$$

La propriété se démontre de manière similaire pour M_α et M'. ■

Ce lemme permet de caractériser simplement les SFPD cohérents :

PROPOSITION II.3 : $\mathscr{D} = (\mathscr{I}, \mathscr{O}, S, M, \text{msg})$ *est cohérent ssi :*

$$\forall\ f, g \in \text{Dom}(M \circ \text{msg}), \quad f M_\alpha g \ \Rightarrow\ \text{msg}(M(f)) = \text{msg}(M(g)),$$

avec M_α défini par le lemme précédent.

Démonstration : Si $\mathscr{D}$ décrit fragmentairement $(\mathscr{I}', \mathscr{O}', S', M', \text{msg}')$ alors nécessairement :

$$\forall\ f, g \in \text{Dom}(M \circ \text{msg}), \quad f M_\alpha g \ \Rightarrow\ f M' g \ \text{(lemme II.2)} \Rightarrow\ \text{msg}'(M'(f)) = \text{msg}'(M'(g)) \ \Rightarrow\ \text{msg}(M(f)) = \text{msg}(M(g)),$$

réciproquement, on montrera que $\mathscr{D}$ est cohérent en exhibant un FDS $(\mathscr{I}, \mathscr{O}', S', M', \text{msg}')$ dont $\mathscr{D}$ est une description fragmentaire. Ce FDS sera construit de la manière suivante :

$\mathscr{O}' = \mathscr{O} \cup \{\perp\}$ en supposant $\perp \notin \mathscr{O}$,

$S' = S_\alpha$,

M' est défini par :

$$\forall f, g \in \mathscr{I}^+,$$

$$f\, M'\, g \quad \Leftrightarrow \quad (\exists h \in \mathrm{Dom}(M \circ \mathrm{msg}) \Rightarrow f\, M\, h\, M\, g),$$

est définie par :

$$\mathrm{msg}' : \quad \mathscr{I}^+ / M' \to \mathscr{C}',$$

$$\forall\ f \in \mathrm{Dom}(M'), \quad \mathrm{msg}'(M'(f)) = Si : \exists h \in \mathrm{Dom}(M \circ \mathrm{msg}),\ h\, M_\alpha\, f.$$

$$Alors : \mathrm{msg}(M(f)).$$

$$Sinon : \perp.$$

Le soin de la vérification est laissé au lecteur. ■

On peut alors énoncer :

COROLLAIRE II.4 : *Tout* FDS *partiel considéré comme* SFPD *est cohérent.*

La démonstration de ce corollaire est identique à celle de la remarque suivant la définition I.3.

Exemple II.2 : A titre d'exemple, voici la définition de l'extension de PS 2.

Nous utilisons toujours les notations de PS 1 et PS 2.

équivalence d'état :

$$\forall\ f, g \in \mathscr{I}^*, \qquad f\, S_\alpha\, g \quad \Leftrightarrow \quad (f = g) \vee (\exists \langle \text{variable} \rangle_1, \exists \langle \text{entier} \rangle_1,$$

$$f, g \in \{\langle \text{variable} \rangle_1 = \langle \text{entier} \rangle\}^* \langle \text{variable} \rangle_1 = \langle \text{entier} \rangle_1)$$

$$\vee (\exists \langle \text{assignation-x} \rangle_1, \exists \langle \text{assignation-y} \rangle_1,$$

$$(f, g \in \langle \text{assignation} \rangle^* \langle \text{assignation-x} \rangle_1 \langle \text{assignation-y} \rangle^* \langle \text{assignation-y} \rangle_1)$$

$$\vee (f, g \in \langle \text{assignation} \rangle^* \langle \text{assignation-y} \rangle_1$$

$$\langle \text{assignation-x} \rangle^* \langle \text{assignation-x} \rangle_1)),$$

équivalence de sortie finale :

$$\forall\ f, g \in \mathscr{I}^+, \qquad f\, M, g \quad \Leftrightarrow \quad (f = g)$$

$$\vee\ (\exists \langle \text{entier} \rangle_1, f, g \in \{\langle \text{entrée} \rangle\}^* \langle \text{variable} \rangle = \langle \text{entier} \rangle_1).$$

L'extension permet d'exhiber et donc de déduire de PS 2 que :

$$\mathbf{x=1} \quad \mathbf{x=2} \quad \mathbf{x=3} \quad S \quad \mathbf{x=3},$$

$$\mathbf{x=1} \quad \mathbf{x=2} \quad \mathbf{y=3} \quad S \quad \mathbf{y=3} \quad \mathbf{x=2},$$

mais ne révèle rien sur l'éventuelle équivalence d'état ou de sortie finale entre :

$$\mathbf{x=1} \quad \mathbf{x?} \quad \text{et} \quad \mathbf{x=2} \quad \mathbf{y?} \quad \mathbf{x=1} \quad \mathbf{x?}$$

PS2 correspond en fait à un SFPD cohérent et à un FDS total (obtenu par extension) si l'on restreint l'ensemble d'entrée ($\mathscr{I}$) aux seules assignations.

II.2. Test de cohérence

Cette section que l'on peut sauter en première lecture, présente une procédure de test de cohérence.

PROPOSITION II.5 : *Soit un* SFPD $=(\mathscr{I}, \mathscr{O}, S, M, \text{msg})$, *et soit la procédure de cohérence C définie ci-après :*

alors, si $C(\{\varepsilon\})=$VRAI, $\mathscr{D}$ *est cohérent;*

si $C(\{\varepsilon\})=$FAUX, $\mathscr{D}$ *est incohérent.*

Définition de la procédure de cohérence :

Procédure booléenne $C(E : E \subset \mathscr{I}^*)$.

Variables globales utilisées par la procédure :

$\mathscr{S}$ initialisée à $\lambda f \in \emptyset . \perp$;

$\mathscr{M}$ initialisée à $\lambda f \in \emptyset . \perp$;

m initialisée à $\lambda x \in \emptyset . \perp$;

/★ Ces trois fonctions partielles (de domaine initial vide) ont les significations suivantes :

$\mathscr{S}$ associe à une séquence de $\mathscr{I}^*$, l'ensemble des séquences que C sait lui être équivalentes pour les états;

$\mathscr{M}$ associe à une séquence de $\mathscr{I}^+$, l'ensemble des séquences que C sait lui être équivalentes pour les sorties finales;

m associe à une classe d'équivalence pour $\mathscr{M}$, le message de $\mathscr{O}$ que C sait lui être associé ★/.

Corps de la procédure :

Si $(E=\emptyset)$.

Alors VRAI.

Sinon, Si $(\exists h \in E,\ h\mathscr{I}^+ \cap (\text{Dom}(M) \cup \text{Dom}(S))=\emptyset)$.

Alors $C(E-\{h\})$.

Sinon, Si $(\exists h \in \mathscr{I}^*, E \subset \mathscr{S}(h))$.

Alors VRAI.

Sinon, Faire.

/★, mise à jour de $\mathscr{S}$ et $\mathscr{M}$ ★/.

$\mathscr{S} \leftarrow \lambda f \in \text{Dom}(\mathscr{S}) \cup E.$ *Si* $(f \in E)$ *Alors E Sinon* $\mathscr{S}(f)$.

$\mathscr{M} \leftarrow \lambda f \in \text{Dom}(\mathscr{S})\mathscr{I} \cup M(E\mathscr{I})$. *Si* $(\exists f' \in \mathscr{I}^+,\ f' \in M(f), \exists a \in \mathscr{I}, \exists g' \in \mathscr{I}^*,$
$f'=g'a, \exists g \in \text{Dom}(\mathscr{S}),\ g' \in \mathscr{S}(g))$.

$$\textit{Alors} : \cup\, \mathscr{S}(g)\, a \underset{h\in\mathscr{S}(g)}{\quad} \mathscr{M}(ha) \underset{h\in\mathscr{S}(g)}{\cup} M(ha).$$

/★ mise à jour de m. Cette mise à jour détecte les incohérences s'il y en a ★/.

Si :

$$(\exists h_1, h_2 \in E, \exists a \in \mathscr{I}, (\mathscr{M}(h_1\, a), \mathscr{M}(h_2\, a) \in \mathrm{Dom}(m)) \wedge (m(\mathscr{M}(h_1\, a)) \neq m(\mathscr{M}(h_2\, a)))).$$

Alors FAUX,

Sinon, Faire :

$$m \leftarrow \lambda\, x \in \{\mathscr{M}(f) \mid f \in \mathrm{Dom}(M \circ \mathrm{msg})\}.$$

Si $(\exists f \in x \cap \mathrm{Dom}(M \circ \mathrm{msg}))$

Alors $\mathrm{msg}(M(f))$;

/★ valeur de la procédure ★/ :

$$\underset{a\in\mathscr{I}}{\wedge}\, C(\mathscr{S}(E\, a) \cup S(E\, a) \cup E\, a);$$

fin de la définition de la procédure.

Avant d'exposer la démonstration de la proposition II.5, effectuons quelques remarques sur cette procédure.

C vise à construire incrémentalement S_α et M_α (tels que définis par le lemme II.2) ainsi qu'à étendre la fonction des messages de sortie. On a en effet :

LEMME II.6 :

$$\forall\ f, g \in \mathscr{I}^*, \quad f \in \mathscr{S}(g) \Rightarrow f\, S_\infty\, g,$$
$$\forall\ f, g \in \mathscr{I}^+, \quad f \in \mathscr{M}(g) \Rightarrow f\, M_\infty\, g,$$

Démonstration : Les propriétés sont manifestement vraies avant appel à C et le restent après chaque modification de $\mathscr{S}$ ou $\mathscr{M}$. ■

Plusieurs tests ont été introduits en tête de la procédure.

La clause $(\exists h \in \mathscr{I}^*, E \subset \mathscr{S}(h))$ permet de ne pas réappliquer la procédure dans un cas déjà étudié.

Les clauses $(E = \phi)$ et :

$$(\exists h \in E, h\mathscr{I}^+ \cap (\mathrm{Dom}(M) \cup \mathrm{Dom}(S)) = \phi,$$

permettent de négliger certaines branches de l'arborescence dont on est sûr qu'elles ne peuvent apporter d'incorrections. En effet :

LEMME II.7 : *Si $\mathscr{D}$ est incohérent et si f est tel que $f\mathscr{I}^+ \cap (\mathrm{Dom}(M) \cup \mathrm{Dom}(S)) = \emptyset$ alors la preuve de l'incohérence de $\mathscr{D}$ peut être faite sans intervention de séquences de fI^+.*

Démonstration : Si $\mathscr{D}$ est incohérent, $\exists g_1, g_2 \in \mathrm{Dom}(M \circ \mathrm{msg})$ tels que :

$$\mathrm{msg}(M(g_1)) \neq \mathrm{msg}(M(g_2)) \qquad \text{et} \qquad g_1 M_\alpha g_2.$$

Si donc :

$$g_1 M_\alpha g_2, \qquad \exists h_1, h_2 \ldots h_n \in \mathscr{I}^+,$$
$$h_1 = g_1, \qquad h_n = g_2 \quad \text{et} \quad \forall\ i, \quad 1 \leqq i < n, \quad h_i \overline{M} h_{i+1}.$$

Si :

$$h_i \in f\mathscr{I}^+, \qquad \exists a \in \mathscr{I}, \quad h_{i-1} = h'_{i-1} a,$$
$$h_{i+1} = h'_{i+1} a \qquad \text{et} \qquad h'_{i-1} S_\alpha h'_{i+1},$$

d'où :

$$h_{i-1} \overline{M} h_{i+1},$$

on réitère le processus jusqu'à complète disparition des éléments de $f\mathscr{I}^+$ ■

Ces dernières clauses ont été introduites pour accélérer les calculs de C nous pouvons dès lors exposer la démonstration de la proposition II.5.

Démonstration : 1^er^ *cas :* Si $C(\{\varepsilon\}) = \mathrm{FAUX}$,

$$\exists E \subset \mathscr{I}^*, \quad \exists h_1, h_2 \in E, \quad \exists a \in \mathscr{I}, \qquad \mathrm{msg}(M(h_1 a)) \neq \mathrm{msg}(M(h_2 a))$$

et comme :

$$h_1, h_2 \in E \quad \Rightarrow \quad h_1 \in \mathscr{S}(h_2) \quad \Rightarrow \quad h_1 S_\alpha h_2 \quad \Rightarrow \quad h_1 a M, h_2 a,$$

$\mathscr{D}$ est incohérent.

2^e^ *cas :* Si $C(\{\varepsilon\}) = \mathrm{VRAI}$, comme $\forall f \in \mathrm{Dom}(S), \exists E \ni f, C(E) = \mathrm{VRAI}$, on a manifestement :

$$\forall\ f, g \in \mathrm{Dom}(S), \qquad f S g \quad \Rightarrow \quad f \mathscr{S} g,$$

et de façon analogue :

$$\forall\ f, g \in \mathrm{Dom}(M), \qquad f M g \quad \Rightarrow \quad f \mathscr{M} g,$$

mais alors par application du lemme fondamental II.2 et du lemme II.6 :

$$\forall\ f, g \in \mathrm{Dom}(\mathscr{S}), \qquad f\, S_\alpha\, g \ \Leftrightarrow\ f\, \mathscr{S}\, g,$$
$$\forall\ f, g \in \mathrm{Dom}(\mathscr{M}), \qquad f\, M_\alpha\, g \ \Leftrightarrow\ f\, \mathscr{M}\, g$$

et donc :

$$\forall\ f, g \in \mathrm{Dom}(M \circ \mathrm{msg}) \subset \mathrm{Dom}(M) \subset \mathrm{Dom}(\mathscr{M}),$$
$$f\, M_\alpha\, g \ \Rightarrow\ f\, \mathscr{M}\, g \ \Rightarrow\ \mathrm{msg}(M(f)) = \mathrm{msg}(M(g)),$$

prouvant que $\mathscr{D}$ est cohérent. ■

Exemple II.3 : Donnons un exemple d'application de cette procédure en montrant que PS 2 est cohérent.

Appel de $C(\{\varepsilon\})$.

Mise à jour de $\mathscr{S}$, $\mathscr{M}$, m, on sait alors que :

$$\varepsilon\, \mathscr{S}\, \varepsilon,$$
$$\langle \text{assignation} \rangle^* x = \langle \text{entier} \rangle_1\ \mathscr{M}\ \langle \text{assignation} \rangle^* y = \langle \text{entier} \rangle_1$$

et que ces dernières séquences sont associées au message $\langle \text{entier} \rangle_1$. Il faut alors calculer :

$$C(\{\langle \text{assignation-x} \rangle_1, \langle \text{assignation-x} \rangle_2 \langle \text{assignation-x} \rangle_1\}),$$

et :

$$C(\{\langle \text{assignation-y} \rangle_1, \langle \text{assignation-y} \rangle_2 \langle \text{assignation-y} \rangle_1\})$$

cette dernière valeur se déduira de la première.

Appel de :

$$C(\{\langle \text{assignation-x} \rangle_1, \langle \text{assignation-x} \rangle_2 \langle \text{assignation-x} \rangle_1\})$$

mise à jour de $\mathscr{S}$ ($\mathscr{M}$ et m ne sont plus modifiés), on sait alors que :

et :

$$\varepsilon\, \mathscr{S}\, \varepsilon$$
$$\langle \text{assignation-x} \rangle_1\ \mathscr{S}\ \langle \text{assignation-x} \rangle_2 \langle \text{assignation-x} \rangle_1.$$

Il faut alors calculer :

$$C(\{\langle \text{assignation-x} \rangle_1,$$
$$\langle \text{assignation-x} \rangle_2 \langle \text{assignation-x} \rangle_1,$$
$$\langle \text{assignation-x} \rangle_3 \langle \text{assignation-x} \rangle_2 \langle \text{assignation-x} \rangle_1\})$$

et :

$$C(\{\langle \text{assignation-x} \rangle_1 \langle \text{assignation-y} \rangle_1,$$
$$\langle \text{assignation-y} \rangle_1 \langle \text{assignation-x} \rangle_1,$$
$$\langle \text{assignation-x} \rangle_2 \langle \text{assignation-x} \rangle_1 \langle \text{assignation-y} \rangle_1 \}).$$

L'appel à :

$$C(\langle \text{assignation-x} \rangle_1,$$
$$\langle \text{assignation-x} \rangle_1 \langle \text{assignation-x} \rangle_1,$$
$$\langle \text{assignation-x} \rangle_3 \langle \text{assignation-x} \rangle_2 \langle \text{assignation-x} \rangle_1 \}),$$

est égal à :

$$C(\{\langle \text{assignation-x} \rangle_1, \langle \text{assignation-x} \rangle_2 \langle \text{assignation-x} \rangle_1 \}),$$

est un cas déjà traité.

L'appel à :

$$(\{\langle \text{assignation-x} \rangle_1 \langle \text{assignation-y} \rangle_1,$$
$$\langle \text{assignation-y} \rangle_1 \langle \text{assignation-x} \rangle_1,$$
$$\langle \text{assignation-x} \rangle_2 \langle \text{assignation-x} \rangle_1 \langle \text{assignation-y} \rangle_1 \}).$$

est égal à :

$$C(\{\langle \text{assignation-x} \rangle_1 \langle \text{assignation-y} \rangle_1,$$
$$\langle \text{assignation-y} \rangle_1 \langle \text{assignation-x} \rangle_1 \})$$

entraînant la mise à jour de $\mathcal{S}$, on sait alors que :

$$\varepsilon \mathcal{S} \varepsilon,$$
$$\langle \text{assignation-x} \rangle_1 \mathcal{S} \langle \text{assignation-x} \rangle_2 \langle \text{assignation-x} \rangle_1$$
$$\langle \text{assignation-x} \rangle_1 \langle \text{assignation-y} \rangle_2 \mathcal{S} \langle \text{assignation-y} \rangle_1 \langle \text{assignation-x} \rangle$$

et il faut calculer :

$$C(\{\langle \text{assignation-x} \rangle_1 \langle \text{assignation-y} \rangle_1 \langle \text{assignation-x} \rangle_2,$$
$$\langle \text{assignation-y} \rangle_1 \langle \text{assignation-x} \rangle_1 \langle \text{assignation-x} \rangle_2 \})$$

et :

$$C(\{\langle \text{assignation-x} \rangle_1 \langle \text{assignation-y} \rangle_1 \langle \text{assignation-y} \rangle_2,$$
$$\langle \text{assignation-y} \rangle_1 \langle \text{assignation-x} \rangle_1 \langle \text{assignation-y} \rangle_2 \}),$$

dont les valeurs se ramènent à $C(\phi)$ donc VRAI.

$C(\{\varepsilon\})$ vaut donc VRAI et PS 2 est cohérent, son extension conduit donc à un FDS total.

II.3. Opérations sur SFPD : union et produit

Nous présentons maintenant deux opérations simples sur les SFPD l'union et le produit.

L'union consiste à agglomérer deux ensembles de propositions pour n'en faire qu'un. Cette fusion n'est possible que si l'on observe quelques précautions.

Le produit consiste à concaténer un ensemble de propositions à chaque séquence d'un ensemble lié par des relations d'équivalence d'états.

L'exemple que nous présenterons sur le chapitre III usera de ces deux opérations et l'on pourra juger de leur commodité.

DÉFINITION II.3 : *Soient :*

$$\mathscr{F}_1=(\mathscr{I}_1, \mathscr{O}_1, S_1, M_1, \mathrm{msg}_1) \qquad et \qquad \mathscr{F}_2=(\mathscr{I}_2, \mathscr{O}_2, S_2, M_2, \mathrm{msg}_2),$$

deux SFPD *quelconques vérifiant :*

$$\forall\ f\in \mathrm{Dom}(M_1\circ \mathrm{msg}_1)\cap \mathrm{Dom}(M_2\circ \mathrm{msg}_2),$$
$$\mathrm{msg}_1(M_1(f))=\mathrm{msg}_2(M_2(f)), \tag{2}$$

alors leur union sera le SFPD *:*

$$\mathscr{F}_1\vee\mathscr{F}_2=(\mathscr{I}_1\cup\mathscr{I}_2, \mathscr{O}_1\cup\mathscr{O}_2, S_1\vee S_2, M_1\vee M_2, \mathrm{msg}),$$

avec $S_1\vee S_2$ *relation d'équivalence définie par :*

$$\mathrm{Dom}(S_1\vee S_2)=\mathrm{Dom}(S_1)\cup\mathrm{Dom}(S_2),$$
$$\forall\ f, g\in \mathrm{Dom}(S_1\vee S_2),$$
$$f\,S_1\vee S_2\,g \iff \exists h_1, \ldots, h_n\in \mathrm{Dom}(S_1\vee S_2),$$
$$h_1=f, \qquad g=h_n,$$
$$(\forall i\in\mathbb{N},\ 1\leqq i<n,\ (h_i S_1 h_{i+1})\vee(h_i S_2 h_{i+1})),$$

$M_1\vee M_2$ *est défini similairement :*

$$\mathrm{msg}: \mathrm{Dom}(M_1\vee M_2)/M_1\vee M_2\to \mathscr{O}_1\cup\mathscr{O}_2,$$
$$\forall\ i\in\{1, 2\},\ \forall\ f\in \mathrm{Dom}(M_i\circ \mathrm{msg}_i),$$
$$\mathrm{msg}(M_1\vee M_2(f))=\mathrm{msg}_i(M_i(f)).$$

La condition (2) permet d'assurer l'existence de la fonction msg du SFPD union.

L'exemple suivant montre que l'union de deux SFPD cohérents n'est pas nécessairement cohérente.

Soit $\mathscr{F}_1$ défini par :

$$f S_1 g \qquad \text{et} \qquad \text{msg}_1(M_1(fa)) = m_1,$$

Soit $\mathscr{F}_2$ défini par :

$$g S_2 h, \qquad \text{msg}_2(M_2(ha)) = m_2,$$

alors $\mathscr{F}_1 \vee \mathscr{F}_2$ est manifestement non cohérent si $m_1 \neq m_2$.

La cohérence d'une union peut se rechercher par l'emploi de la condition énoncée dans la proposition II.3. Mentionons cependant l'important cas particulier.

COROLLAIRE II.8 : *Si $\mathscr{F}_1$ est un* SFPD *cohérent d'ensemble d'entrées $\mathscr{I}_1$; si $\mathscr{F}_2$ est un* SFPD *cohérent d'ensemble d'entrées $\mathscr{I}_2$, et si $\mathscr{I}_1 \cap \mathscr{I}_2 = \emptyset$ alors $\mathscr{F}_1 \vee \mathscr{F}_2$ est un* SFPD *cohérent.*

La démonstration est évidente.

DÉFINITION II.4 : *Soient $\mathscr{F}_1 = (\mathscr{I}_1, \mathscr{O}_1, S_1, M_1, \text{msg}_1)$ et :*

$$\mathscr{F}_2 = (\mathscr{I}_2, \mathscr{O}_2, S_2, M_2, \text{msg}_2),$$

deux SFPD *quelconques vérifiant :*

$$\left.\begin{array}{l} \forall\ f \in \text{Dom}(S_1)\,\text{Dom}(M_2 \circ \text{msg}_2), \qquad \exists!\, m \in \mathscr{O}_2,\ \forall\ f_1 \in \text{Dom}(S_1), \\ \forall\ f_2 \in \text{Dom}(M_2 \circ \text{msg}_2), \qquad f = f_1 f_2, \qquad \text{msg}_2(M_2(f_2)) = m. \end{array}\right\} \quad (3)$$

Alors on définit le SFPD *produit par $(\mathscr{I}_1 \cup \mathscr{I}_2, \mathscr{O}_2, S, M, \text{msg}) = \mathscr{F}_1 \to \mathscr{F}_2$ avec S cloture transitive de la relation $\overline{S}$ définie par :*

$$\forall\ f, g \in \text{Dom}(S_1)\,\text{Dom}(S_2),$$

$$f \overline{S} g \iff \forall\ f_1, g_1 \in \text{Dom}(S_1),$$

$$\forall\ f_2, g_2 \in \text{Dom}(S_2), \qquad f = f_1 f_2, \qquad g = g_1 g_2, \qquad f_1 S_1 g_1, \qquad f_2 S_2 g_2.$$

M cloture transitive de la relation $\overline{M}$ définie par :

$$\forall\ f, g \in \text{Dom}(S_1)\,\text{Dom}(M_2),$$

$$f \overline{M} g \iff \forall\ f_1, g_1 \in \text{Dom}(S_1),$$

$$\forall\ f_2, g_2 \in \text{Dom}(M_2), \qquad f = f_1 f_2, \qquad g = g_1 g_2, \qquad f_1 S_1 g_1, \qquad f_2 M_2 g_2.$$

msg *fonction des messages de sortie.*

$\text{msg} : \text{Dom}(S_1)\ \text{Dom}(M_2)/M_2 \to \mathcal{O}_2$:

$$\forall\ f \in \text{Dom}(S_1)\,\text{Dom}(M_2 \circ \text{msg}_2),\ \forall\ f_1 \in \text{Dom}(S_1),$$
$$\forall\ f_2 \in \text{Dom}(M_2 \circ \text{msg}_2), \qquad f = f_1 f_2, \qquad \text{msg}(f) = \text{msg}_2(f_2).$$

La condition (3) assure la correction de la définition de la fonction des messages de sortie. Les entités $\mathcal{O}_1$, M_1, msg_1 n'interviennent nullement dans cette définition, aussi sait-on réaliser le produit d'un ensemble quelconque de séquences de $\mathcal{I}^*$ par un SFPD.

DÉFINITION II.5 : *Si* $P \subset \mathcal{I}^*$ *et si* $\mathcal{F}$ *est un* SFPD, *on notera* $P \to \mathcal{F}$ *le* SFPD *défini par le produit* $\mathcal{F}_P \to \mathcal{F}$ *où* $\mathcal{F}_P = (\mathcal{I}, \phi, =|_P, \phi, \phi)$.

Mentionnons l'utile proposition suivante :

PROPOSITION II.9 : *Si* $\mathcal{F}_1$ *est cohérent et si* $\mathcal{F}_1 \to \mathcal{F}_2$ *existe, alors* $\mathcal{F}_1 \to \mathcal{F}_2$ *est cohérent.*

Démonstration : Si $\mathcal{F}_2$ est une description fragmentaire de $(\mathcal{I}'_2, \mathcal{O}'_2, S'_2, M'_2, \text{msg}'_2)$ alors on vérifiera que $\mathcal{F}_1 \to \mathcal{F}_2$ est une description fragmentaire de $(\mathcal{I}_1 \cup \mathcal{I}'_2, \mathcal{O}'_2 \cup \{\perp\}, S', M', \text{msg}')$ où $\perp \notin \mathcal{O}'_2$.

S' relation d'équivalence définie par la clôture transitive de $\overline{S'}$:

$$\forall f, g \in (\mathcal{I}_1 \cup \mathcal{I}'_2)^+,$$

$$f\,\overline{S'}\,g \quad \Leftrightarrow \quad (f = g) \vee (\exists f_1, g_1 \in \text{Dom}(S_1),\ f = f_1 f_2,\ g = g_1 g_2, f_1\ S_1\ g_1, f_2\ S'_2\ g_2).$$

M' relation d'équivalence définie par la clôture transitive de $\overline{M'}$:

$$\forall f, g \in (\mathcal{I}_1 \cup \mathcal{I}'_2)^+,$$

$$f\,\overline{M'}\,g \quad \Leftrightarrow \quad (\exists f_1, g_1 \in \text{Dom}(S_1), f = f_1 f_2, g = g_1 g_2 \quad \Rightarrow \quad f_2\ M'_2\ g_2).$$

msg′ fonction des messages de sortie.

$$\text{msg}' : (\mathcal{I}_1 \cup \mathcal{I}'_2)^+ / M' \to \mathcal{O}'_2 \cup \{\perp\},$$

$\text{msg}'(M'(f)) =$ *Si* $\exists f_1 \in \text{Dom}(S_1), \quad f = f_1 f_2,$

Alors $\text{msg}'_2(M'_2(f_2))$,

Sinon $\perp$. ■

Les opérations d'union et de produit entretiennent d'étroits rapports entre elles, toutefois leurs relations ne sont pas toujours harmonieuses.

Exemple d'union et de produit II.4 : Soit le SFPD PS 3 défini par (notations de PS 1 et PS 2) :

équivalence d'état :

$$\langle \text{question} \rangle \quad S \quad \varepsilon,$$

équivalence de sorties finales :

néant,

soit le SFPD PS 4 défini par :

équivalence d'état :

néant,

équivalence de sorties finales :

$$\langle \text{entrée} \rangle^* \ \mathbf{x} = \langle \text{entier} \rangle_1 \ \{ \langle \text{question} \rangle, \langle \text{assignation-y} \rangle \}^* \mathbf{x} ?$$

M

$$\langle \text{entrée} \rangle^* \ \mathbf{y} = \langle \text{entier} \rangle_1 \ \{ \langle \text{question} \rangle, \langle \text{assignation-x} \rangle \}^* \mathbf{y} ?$$

M

$$\langle \text{entrée} \rangle^* \ \langle \text{variable} \rangle = \langle \text{entier} \rangle_1$$

et :

$$\{ \langle \text{question} \rangle, \langle \text{assignation-x} \rangle \}^* \ \mathbf{y} \ ? \ M \ \{ \langle \text{question} \rangle, \langle \text{assignation-y} \rangle \}^* \mathbf{x} ?$$

message de sortie :

$$\text{msg}\ (M\ (\mathbf{x}\ ?)) = \textbf{indéfini}.$$

Étant donné les SFPD PS 1, PS 2, PS 3 précédemment définis, on vérifiera que :

$$\text{PS}\,1 = (\{ \langle \text{entrée} \rangle \}^* \rightarrow (\text{PS}\,2 \vee \text{PS}\,3)) \vee \text{PS}\,4,$$

PS 4 spécifie entièrement les sorties associées à une question. Le produit permet de faire jouer les équivalences d'état de PS 2 et PS 3 sur n'importe quelle séquence d'entrées. Par exemple on peut maintenant déterminer que :

$$\mathbf{x=1} \quad \mathbf{x}\,? \qquad \text{et} \qquad \mathbf{x=2} \quad \mathbf{y}\,? \quad \mathbf{x=1} \quad \mathbf{x}\,?$$

sont équivalentes pour les sorties finales (PS 4) et pour les états car :

$$\mathbf{x=1} \quad \mathbf{x}\,? \quad S \quad \mathbf{x=1}$$

et :

$$\mathbf{x=2}\quad \mathbf{y?}\quad \mathbf{x=1}\quad \mathbf{x?}\quad S\quad \mathbf{x=2}\quad \mathbf{y?}\quad \mathbf{x=1},$$

or :

$$\mathbf{x=2}\quad S\quad \mathbf{x=2}\quad \mathbf{y?}$$

donc :

$$\mathbf{x=2}\quad \mathbf{x=1}\quad S\quad \mathbf{x=2}\quad \mathbf{y?}\quad \mathbf{x=1}$$

et comme :

$$\mathbf{x=2}\quad \mathbf{x=1}\quad S\quad \mathbf{x=1}.$$

l'équivalence d'état est bien démontrée.

II.4. Tableaux

Les tableaux, dernières entités que nous présentons ici, constituent une généralisation des tables d'évolution des automates finis, permettant une représentation visuelle des relations d'équivalence.

Ces tableaux forment une synthèse entre les définitions « mécanistes » des FDS (donnée d'une fonction d'évolution) et « syntaxiques » (explication des relations d'équivalence). Ils permettent à tous niveaux de combiner ces deux types de définition.

Les tableaux auxquels on peut associer des SFPD, sont justiciables des mêmes prédicats :

cohérence;

totalité (peut-on associer au tableau un FDS total?);

réduction (le FDS associé est-il réduit?).

L'intérêt des tableaux est d'utiliser à fond et implicitement l'extension déjà décrite.

On détermine un ensemble de séquences représentatives c'est-à-dire pour tout état, on retient une ou plusieurs séquences de cet état pour le représenter. Puis on définit les opérateurs (associés aux transitions entre états) qui appliqués à une séquence représentative d'un état, ont pour valeur une séquence représentative de l'état résultant. Un tableau n'est intéressant que si l'on peut regrouper des états « similaires » c'est-à-dire justiciables des mêmes opérateurs pour des entrées communes.

Gigantisme des tableaux et complexité des opérateurs y figurant sont des paramètres antagonistes.

DÉFINITION III.6 : *Un tableau est un sextuplet* $(\mathcal{I}, \mathcal{C}, \{\rho_i\}_i, \{T_j\}_j, \{\sigma_{ij}\}_{ij}, \{\tau_{ij}\}_{ij})$ *avec* $\{\rho_i\}_i$ *famille de sous-ensembles disjoints de* $\mathcal{I}^*$, $\{T_j\}_j$ *famille de sous-*

ensembles disjoints de $\mathscr{I}$, $\{\sigma_{ij}\}_{ij}$ *famille de fonctions partielles* $\rho_i \times T_j \to \mathscr{I}^*$, $\{\tau_{ij}\}_{ij}$ *famille de fonctions partielles* $\rho_i \times T_j \to \mathscr{C}$.

On représentera commodément un tableau par :

	T	T
ρ.....	σ_{11}/τ_1	$\sigma_{12}/$
ρ.....	$/\tau_{21}$	/

Les fonctions manquantes τ_{12}, σ_{21}... sont des fonctions partielles à domaine vide.

On utilise les tableaux de la manière suivante :

Si la séquence représentative de l'état est f et que l'entrée est a alors, si $\exists\, i,j,(f,a) \in \mathrm{Dom}(\sigma_{ij})$ la nouvelle séquence représentative est $\sigma_{ij}(f,a)$.

Le message associé à cette transition est :

si $(f,a) \in \mathrm{Dom}(\tau_{ij})$, $\quad \tau_{ij}(f,a)$.

Le mode d'emploi des tableaux en fait naturellement un SFPD.

DÉFINITION II.7 : *Si* $\mathscr{T}$ *est un tableau* $(\mathscr{I}, \mathscr{C}, \{\rho_i\}_i, \{T_j\}_j, \{\sigma_{ij}\}_{ij}, \{\tau_{ij}\}_{ij})$ *on lui associera le* SFPD $(\mathscr{I}, \mathscr{C}, S, M, \mathrm{msg})$ *où* S *est la clôture transitive et symétrique de la relation* σ *définie par :*

$$\forall\ f, g \in \mathscr{I}^*,\ \forall\ a \in \mathscr{I},$$
$$fa\,\sigma\,g \iff \exists\ i,j,\quad (g = fa) \vee (\sigma_{ij}(f,a) = g).$$

M est la relation :

$$\forall\ f, g \in \mathscr{I}^*,\ \forall\ a, b \in \mathscr{I},$$
$$fa\ M\ gb \iff \exists\ i,j,k,l,\quad \tau_{ij}(f,a) = \tau_{kl}(g,b),$$

msg :

$$\mathrm{Dom}(M)/M \to \bigcup_{i,j} \mathrm{Im}(\tau_{ij}),$$
$$\forall\ f \in \mathscr{I}^*,\ \forall\ a \in \mathscr{I},\quad fa \in \mathrm{Dom}(M),$$
$$\exists\ i,j,\quad \mathrm{msg}(M(fa)) = \tau_{ij}(f,a).$$

DÉFINITION II.8 : *Un tableau sera cohérent si son* SFPD *associé l'est aussi.*

Exemple de tableau II.5 : PS 1 peut être représenté par le tableau suivant, les notations employées sont toujours celles de PS 1 et PS 2 hormis les symboles ← et ↑ dont les sens sont les suivants :

←, représente la séquence représentative de l'état actuel;

↑, représente l'entrée actuelle.

	x ?	y ?	$x = \langle$ entier $\rangle$	$y = \langle$ entier $\rangle$
ε	←/ **indéfini**	←/ **indéfini**	↑/ ⟨ entier ⟩	↑/ ⟨ entier ⟩
$x = \langle$ entier $\rangle_1$	←/ ⟨ entier ⟩$_1$	←/ **indéfini**	↑/ ⟨ entier ⟩	← ↑/ ⟨ entier ⟩
$y = \langle$ entier $\rangle_1$	←/ **indéfini**	←/ ⟨ entier ⟩$_1$	↑ ←/ ⟨ entier ⟩	↑/ ⟨ entier ⟩
$x = \langle$ entier $\rangle_1$ $y = \langle$ entier $\rangle_2$	←/ ⟨ entier ⟩$_1$	←/ ⟨ entier ⟩$_2$	↑ $y = \langle$ entier $\rangle_2$ ⟨ entier ⟩	$x = \langle$ entier $\rangle_1$ ↑ ⟨ entier ⟩

On peut facilement vérifier que le tableau est cohérent et que le SFPD associé au tableau spécifie PS 1 puisque les relations S et M de PS 1 sont manifestement vraies sur le tableau. Par exemple :

si $f, g \in \{ \langle$ assignation-y $\rangle, \langle$ question $\rangle \}^* \langle$ assignation-y $\rangle_1 \langle$ question $\rangle^*$

(donc f, g représentent des états équivalents pour PS 1).

Alors f et g sont deux séquences plus agréablement représentées par la séquence ⟨ assignation-y ⟩$_1$ (de par l'utilisation de tableau) et sont donc équivalentes pour le tableau.

III. EXEMPLE

Les techniques développées ci-avant nous ont permis de décrire trois Systèmes Interactifs bien différents (*voir* [23]) :

une calculatrice de poche (SR 50 de Texas Instruments) [30];

un éditeur graphique de réseaux;

un système Lisp interactif [24].

Les formalisations auxquelles nous avons abouti ne se ressemblent guère. Nous avions 15 tableaux (totalisant 40 lignes et 17 colonnes) formés d'opérateurs simples pour la calculatrice, mais seulement un unique tableau (2 lignes sur 1 colonne) muni d'opérateurs très complexes pour LISP. Il ne nous est pas possible d'exposer ici les détails de tels exemples, aussi présenterons-nous une petite calculatrice de poche apparentée au modèle SR 50 de Texas Instruments [30].

Une vision, toute informelle, de cette calculatrice est donnée dans la figure 2.

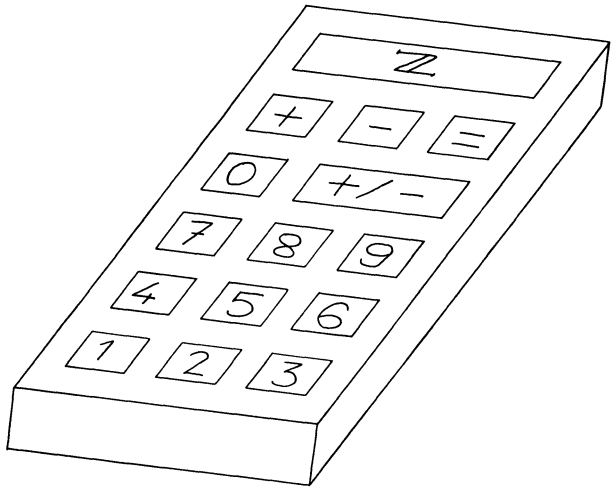

Figure 2. – **Calculatrice** (schéma).

Donnons de cette calculatrice la sémantique informelle suivante :
les touches peuvent être divisées en deux groupes :

$\{\mathbf{0, 1, 2, 3, 4, 5, 6, 7, 8, 9}\}$ touches permettant l'introduction des nombres entiers positifs ($+/-$ permet d'affecter le signe négatif au nombre affiché);

$\{+, -, =\}$ opérateurs d'addition, de soustraction et d'égalité.

Ce dernier permet d'obtenir le résultat d'une opération.

III.1. Définition formelle de la calculatrice

Cette description s'effectuera en trois tableaux.

Ensemble des entrées : $\mathscr{I} = \{\mathbf{0, 1, 2, 3, 4, 5, 6, 7, 8, 9}, +/-, +, -, =\}$.

Ensemble des sorties $\mathscr{O} = \mathbb{Z}$ (ensemble des entiers relatifs).

III.1.1. *Tableau* N *décrivant l'insertion d'un nombre*

Ce tableau est un tableau de Moore. Les messages seront portés en regard des états d'arrivée.

Nous emploierons les notations suivantes :

$$c \in \{\mathbf{1, 2, 3, 4, 5, 6, 7, 8, 9}\} \text{ chiffre non nul,}$$

$\uparrow$ représente l'actuelle entrée c'est-à-dire le symbole associé à la colonne.

La fonction $- : \{c\}\{c, \mathbf{0}\}^* \to \mathbb{N}$ est la fonction qui associe un nombre entier à la chaîne de caractères la codant.

	0	c	+/−	Message
ε.	↑	↑	+/−**0**	
0.	↑	↑	+/−**0**	**0**
+/−**0**.	+/− ↑	+/− ↑	0	**0**
c{c, **0**}*.	c{c, **0**}* ↑		+/−c{c, **0**}*	c{c, **0**}*
+/−c{c, **0**}*. . . .	+/−c{c, **0**}* ↑		c{c, **0**}*	−c{c, **0**}*

Les remarques suivantes peuvent être faites :

1° il y a élimination systématique des zéros non significatifs par exemple;

$$\mathbf{00007} \quad S \quad \mathbf{7};$$

2° la place du signe négatif peut être normalisée à gauche :

$$+/-\mathbf{37} \quad S \quad \mathbf{3}+/-\mathbf{7} \quad S \quad \mathbf{37}+/-;$$

3° il n'y a pas de restrictions sur la taille d'un nombre.

4° L'état initial (ε) et l'état de la calculatrice après la séquence formée de l'unique touche **0** peuvent être confondus.

Ce tableau est cohérent sur l'ensemble $\{c, \mathbf{0}, +/-\}^*$ et donc sur $\{\mathbf{0}, \mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}, \mathbf{5}, \mathbf{6}, \mathbf{7}, \mathbf{8}, \mathbf{9}, +/-\}^*$. En effet les propositions initiales concernant le SFPD associé au tableau N sont :

$$+/- \quad S \quad +/-\mathbf{0} \quad S \quad \mathbf{0}+/- \quad S \quad +/-\mathbf{00},$$

$$\mathbf{00} \quad S \quad \mathbf{0} \quad S \quad +/-\mathbf{0}+/-,$$

$$\mathbf{0}\,c \quad S\,c,$$

$$+/-\mathbf{0}\,c \quad S+/-c,$$

$$c\{c, \mathbf{0}\}^*+/-S \quad +/-c\{c, \mathbf{0}\}^*,$$

$$+/-c\{c, \mathbf{0}\}^*+/- \quad S \quad c\{c, \mathbf{0}\}^*.$$

$+/-\mathbf{0}+/-M \circ M+/-M+/-\mathbf{0}\ M \circ\circ M\ \mathbf{0}+/-M+/-\mathbf{00}$ associé au message **0**,

$\mathbf{0}\,c \quad M \quad c$ associé au message $\underline{c}$,

$+/-\mathbf{0}\,c \quad M \quad +/-c$ associé au message $-\underline{c}$,

$c\{c, \mathbf{0}\}^* +/- \quad M \quad +/-c\{c, \mathbf{0}\}^*$ associé au message $-c\{c, \mathbf{0}\}^*$,

$+/-c\{c, \mathbf{0}\}^* +/- \quad M \quad c\{c, \mathbf{0}\}^*$ associé au message $c\{c, \mathbf{0}\}^*$.

Le seul cas où la cohérence ne semble pas immédiatement assurée (par application de la procédure C) est :

$$+/- \quad S \quad +/-\mathbf{0},$$

mais :

$$C(+/-, +/-\mathbf{0}) = C(+/-\mathbf{0}, +/-\mathbf{00}),$$
$$\wedge C(+/-c, +/-\mathbf{0}\,c),$$
$$\wedge C(+/- \; +/-, +/-\mathbf{0}+/-),$$

relations valant toutes VRAI car déjà connues ou ne pouvant apporter d'incohérences.

La cohérence du tableau provient d'un choix correct de séquences représentatives des états. Par exemple, la séquence $+/-\mathbf{1}$ correspond à la classe des séquences appartenant à :

$$\{\{\mathbf{0}\}^* +/-\{\mathbf{0}\}^* +/-\}^* \{\mathbf{0}\}^* \{+/-\{\mathbf{0}\}^* \mathbf{1}, \mathbf{1}\{+/- \; +/-\}^* +/-\},$$

ceci se vérifie simplement sur le graphe suivant, directement extrait du tableau N.

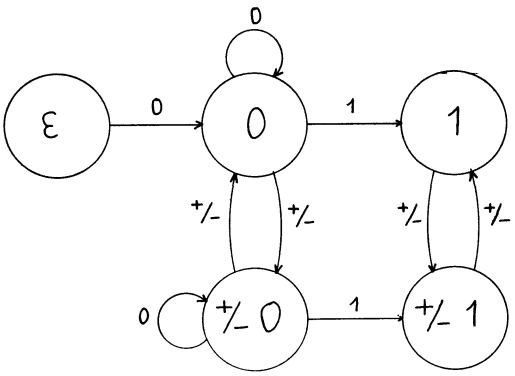

Figure 3. – Automate associé à la séquence $+/-\mathbf{1}$.

III. 1.2. *Tableau U décrivant les opérateurs*

Ce tableau est un tableau de Moore.

Nous emploierons les notations suivantes :

$$\mathbf{s} \in \{+, -\},$$
$$\mathbf{u} \in \{+, -, =\},$$
$$\mathbf{n} \in \{+/-, \varepsilon\}\{\mathbf{c}, \mathbf{0}\}^*,$$

	u	Messages
ε	0u	0
n.	nu	$\underline{\mathbf{n}}$
nu.	n↑	$\underline{\underline{\mathbf{n}}}$
$\mathbf{n}_1 \mathbf{s} \mathbf{n}_2$	$\gamma(\mathbf{n}_1 \mathbf{s} \mathbf{n}_2 \mathbf{u})$	$\underline{\mathbf{n}_2}$

Les remarques suivantes peuvent être formulées :

1° **0** est assumé s'il n'y a pas de nombre précédant l'opérateur par exemple :

$$+\mathbf{2} \quad S \quad \mathbf{0}+\mathbf{2};$$

2° un opérateur nouveau dépile l'ancien :

$$\mathbf{2}+-+\mathbf{3} \quad S \quad \mathbf{2}+\mathbf{3};$$

3° la fonction σ est définie par les équivalences suivantes où $\{\ldots\}_i^*$ dénote une chaîne de caractères intervenant identiquement plusieurs fois.

$$\sigma(\{\mathbf{c}\}_1^*\mathbf{c}_1\{\mathbf{0}\}_1^* + \{\mathbf{c}\}_2^*\mathbf{c}_2\{\mathbf{0}\}_1^*\mathbf{u}) \quad S$$
$$\sigma(\sigma(\{\mathbf{c}\}_1^*\mathbf{0}\{\mathbf{0}\}_1^* + \{\mathbf{c}\}_2^*\mathbf{0}\{\mathbf{0}\}_1^* +)\sigma(\mathbf{c}_1\{\mathbf{0}\}_1^* + \mathbf{c}_2\{\mathbf{0}\}_2^*\mathbf{u})),$$
$$\sigma(\{\mathbf{c}\}_1^*\mathbf{0}\{\mathbf{0}\}_1^* + \mathbf{c}\{\mathbf{0}\}_1^* u) \quad S \quad \{\mathbf{c}\}_1^*\mathbf{c}\{\mathbf{0}\}_1^* u,$$
$$\sigma(\mathbf{c}\{\mathbf{0}\}_1^* + \{\mathbf{c}\}_1^*\mathbf{0}\{\mathbf{0}\}_1^* u) \quad S \quad \{\mathbf{c}\}_1^*\mathbf{c}\{\mathbf{0}\}_1^* u,$$
$$\sigma(\mathbf{1}\{\mathbf{0}\}_1^* + \mathbf{1}\{\mathbf{0}\}_1^*\mathbf{u}) \quad S \quad \mathbf{2}\{\mathbf{0}\}_1^*\mathbf{u},$$
$$\sigma(\mathbf{1}\{\mathbf{0}\}_1^* + \mathbf{2}\{\mathbf{0}\}_1^*\mathbf{u}) \quad S \quad \mathbf{3}\{\mathbf{0}\}_1^*\mathbf{u},$$
$$\ldots \qquad \ldots \qquad \ldots$$
$$\sigma(\mathbf{9}\{\mathbf{0}\}_1^* + \mathbf{9}\{\mathbf{0}\}_1^*\mathbf{u}) \quad S \quad \mathbf{18}\{\mathbf{0}\}_1^*\mathbf{u}.$$

La définition de la soustraction par σ est omise mais pourra être simplement rétablie.

Les calculs nécessaires pour l'addition peuvent s'effectuer syntaxiquement sur les séquences d'entrées. Cette particularité n'existe pas pour tout langage, en particulier si l'on remplaçait les touches d'entrée + et − par (par exemple) sinus et cosinus on ne pourrait représenter syntaxiquement les calculs nécessaires pour obtenir le sinus d'un nombre. La définition de σ revient à définir les règles d'équivalence d'états en supplément de celles déjà données dans le tableau. On

peut percevoir là les différences entre les comportements conversationnels (exprimés par les tableaux) et interprétatifs (σ) de la calculatrice. Ces règles sont orientées car elles permettent, appliquées de la gauche vers la droite de déterminer au sein d'un état la séquence représentative de cet état.

Par exemple :

$$\sigma\ (\mathbf{19+2=})\quad S\quad \mathbf{19+2=}$$

$$S\quad \sigma(\sigma(\mathbf{10+0+})\sigma(\mathbf{9+2=}))\quad S\quad \mathbf{10+0+9+2=}$$

$$S\quad \sigma(\mathbf{10+11})\quad S\quad \mathbf{10+11=}$$

$$S\quad \sigma(\sigma(\mathbf{10+10+})\sigma(\mathbf{0+1=}))\quad S\quad \mathbf{10+10+0+1=}$$

$$S\quad \sigma(\mathbf{20+1=})\quad S\sigma\ \mathbf{20+1=}$$

$$S\quad \mathbf{21=}.$$

III.1.3. *Tableau T, insertion second nombre*

Le tableau T est le suivant :

	c	**+/−**
n=.....	**c**	N**(n, +/−)=**

On peut remarquer d'après ce tableau qu'il est possible de changer le signe d'un résultat, la fonction N est la fonction associée au tableau N, dont la valeur est la séquence représentative associée à l'état de 1er argument et l'entrée de 2^{e} argument.

T est cohérent : sa fonction des messages a un domaine vide. En outre $N \cup T$ est cohérent puisqu'aucune incohérence ne saurait surgir de l'ensemble de proposition associée à N augmenté des propositions de T qui sont :

$$\mathbf{n=c}\quad S\quad \mathbf{c},$$

$$\mathbf{n=+/-}\quad S\quad N\mathbf{(n,\ +/-)=+/-},$$

$$\mathbf{n=c}\quad M\quad \mathbf{c},$$

$$\mathbf{n=+/-}\quad M\quad N\mathbf{(n,\ +/-)=+-}.$$

III.2. Définition complète de la calculatrice

La calculatrice complète et définie par la formule :

$$N \cup T \cup U \cup (\{\mathbf{ns}\} \rightarrow N).$$

Cette équation conduit à un SFPD cohérent qui aurait pu sans inconvénient être défini en un seul tableau. Toutefois la présentation en plusieurs tableaux permet de bien dissocier les différentes phases d'utilisation de la calculatrice. Elle permet de plus de s'assurer rapidement de ce qu'aucune entrée en aucun état n'a été oubliée.

L'exemple suivant permettra de se familiariser avec la manipulation des tableaux :

« quelle est la séquence représentative de l'état dans lequel sera la calculatrice après la séquence d'entrée **1**+/−**5**−**4**=+/−+**2**= ? la réponse sera :

Entrées successives	Séquences représentatives
1.	**1**
+/−.	+/**1**
5.	+/−**15**
−.	+/−**15**−
4.	+/−**15**−**4**
=.	+/−**19**=
+/−.	**19**=
+.	**19**+
2.	**19**+**2**
=.	**21**=

La calculatrice que nous venons de définir n'est qu'un jouet assez simple. Mais les opérateurs que nous avons introduits permettent d'aisément la compliquer, Par exemple, si nous envisageons d'ajouter une touche d'effacement [C] (clear) à notre machine, ainsi qu'une touche [CE] effaçant le dernier opérande et seulement celui-là (imprécise sémantique s'il en fut !) il suffit pour obtenir une telle machine, d'ajouter le tableau :

	C	CE
n.	ε	ε
n=.	ε	ε
ns.	ε	**ns**
$\mathbf{n}_1\,\mathbf{sn}_2$.	ε	$\mathbf{n}_1\,\mathbf{s}$

La cohérence de l'union des tableaux est acquise puisque les ensembles d'entrées sont différents.

Ce dernier tableau précise totalement le sens de la touche [CE].

IV. CONCLUSIONS

Les méthodes que l'on vient d'exposer et de présenter sur un exemple permettent de décrire rapidement de nombreux cas simples. La principale

caractéristique est que la définition obtenue est largement indépendante d'une éventuelle implémentation. Ces méthodes reposent sur des techniques simples de description :

– définition précise de l'équivalence, correspondant à une notion intuitive souvent utilisée dans les manuels d'emploi;

– lisibilité des définitions : employant l'arsenal largement connu des notations syntaxiques, il est possible d'envisager de donner aux utilisateurs les définitions des SI. Les équivalences apparaissent comme des exemples paramétrés : faire « . . . » revient à faire « . . . ».

Une certaine classe de langages de SI est favorisée par ces méthodes : ce sont les langages tels que leur définition n'utilise qu'eux mêmes (ainsi que les notations syntaxiques usuelles).

La petite calculatrice du chapitre III est un exemple, L I S P en est un autre. La « circularité » de ces définitions, limitant le nombre de notions nécessaires, rend leur compréhension plus aisée.

D'un point de vue méthodologique, la construction des tableaux permet d'envisager successivement tous les cas. Ces mêmes tableaux peuvent être perçus comme un outil de communication entre concepteurs et implémenteurs.

REMERCIEMENT

Je remercie M. J.-F. Perrot de l'aide qu'il m'a apportée pour la rédaction de cet article.

BIBLIOGRAPHIE

1. J. ALLEN, *The Anatomy of LISP*, McGraw Hill, 1978.
2. B. N. BARNES *An Automata Theoretic Approach to Interactive Computer Graphics Commande Languages*, in Applied Computation Theory: Analysis, Design, Modeling, R. T. YEH, éd., Prentice-Hall, 1976, p. 565-581.
3. S. J. BOIES et M. F. SPIEGEL, *A Behavioral Analysis of Programming, The use of Interactive Debugging Facilities*, Microfiche AD-772/127.
4. S. H. CAMERON, D. EWING et M. LIVERIGHT, *DIALOG: a Conversational Programming System with a Graphical Orientation*, Com. A.C.M., vol. 10, n° 6, juin 1967, p. 349-357.
5. J. P. CRESTIN et C. QUEINNEC, *Towards Semantics for Graphical Interactions, SEILLAC II, Methodology of Interaction*, R. GUEDJ, éd., North Holland, 1979.
6. Y. CHU et E. R. CANNON, *Interactive High-Level Language Direct-Execution Microprocessor System*, I.E.E.E. Transactions on Software Engineering, vol. SE-2, n° 2, juin 1976, p. 126-134.
7. G. DEMARS, J. C. RAULT et G. RUGGIU, *Le langage et les systèmes APL*, Masson, Paris, 1974.
8. S. FEYOCK, *Transition Diagram-based CAI/HELP Systems*, Int. J. Man-Machine Studies, vol. 9, 1977, p. 399-413.
9. D. FRIED, *On the User's Point of View*, in [13], p. 11-21.

10. G. FRIESLAND, *Von der Programmerstellung zur Programmierumgebung – Zum Begriff des interaktiven Programmierens*, I.F.I., Université de Hambourg, mars 1976.
11. W. J. HANSEN, *User Engineering Principles for Interactive Systems*, A.F.I.P.S., proc., vol. 39, 1971, p. 523-532.
12. J. HARTMANIS et R. E. STEARNS, *Algebraic Structure Theory of sequential Machines*, Prentice-Hall, Englewood Cliffs, N.J., 1966.
13. H. J. HOFFMANN, *Programming by Selection*, International Computing Symposium 73, North Holland, 1974, p. 59-65.
14. M. KLERER et J. REINFELDS, *Interactive Systems for Experimental applied Maths*, Academic Press, N.Y., 1968.
15. I. KUPKA, *A Formal framework for Dialog Languages*, Bericht n° 2, décembre 1972, I.F.I., Université de Hambourg.
16. I. KUPKA, H. OBERQUELLE et N. WILSING, *An experimental Language for Conversational use*, Bericht n° 18, septembre 1975, I.F.I., Université de Hambourg.
17. I. KUPKA et N. WILSING, *A n APL-based Syntax for Dialog Languages*, APL congress 73, GJERLOV, HELMS et NIELSEN, éd., North Holland, 1974, p. 269-273.
18. I. KUPKA et N. WILSING, *Functions Describing Interactive Programming*, Proc. of International Computing Symposium 73, North Holland, 1974, p. 41-45.
19. J. MARTIN, *Design of Man-computer Dialog*, Prentice-Hall, Englewood Cliffs N. J., 1973.
20. J. G. MITCHELL, A. J. PERLIS et H. R. VAN ZOEREN, LC^2 *a language for Conversational Programming*, in: [13].
21. W. M. NEWMAN et R. F. SPROULL, *Principles of Interactive Computer Graphics*, McGraw-Hill, 1973.
22. E. I. ORGANICK et J. W. THOMAS, *Computer-Generated Semantics Displays*, Information Processing 74, North Holland, 1974, p. 898-902.
23. C. QUEINNEC, *Contribution à l'étude des systèmes interactifs*, Thèse de Docteur-ingénieur, Université de Paris-VI, 1978.
24. C. QUEINNEC, *Another LISP Formalization*, LISP bulletin (à paraître).
25. R. SCHILD, *Interactive Structured Programming*, International Computing Symposium 73, North Holland, 1974, p. 81-84.
26. W. SIBLEY, *The Engineering Assistant*, in: [13], p. 138-143.
27. S. C. SHAPIRO et S. C. KWASNY, *Interactive Consulting via Natural Language*, Com. A.C.M., vol. 18, n° 8, août 1975, p. 459-462.
28. R. SPENCE et M. APPERLEY, *The Interactive-Graphic Man-Computer Dialogue in Computer-Aided Circuit Design, I.E.E.E. transactions on circuits and systems*, vol. CAS-24, n° 2, février 1977, p. 49-61.
29. P. H. STARKE, *Abstract Automata*, North-Holland, 1972.
30. TEXAS INSTRUMENTS, SR 50 : *Manuel d'utilisation.*