Algebraic calculi of processes and net expressions

Georges Gonthier

INRIA

CENTRE
SOPHIA ANTIPOLIS

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France
Tél.:(3) 954 90 20

Rapports de Recherche

N° 367

ALGEBRAIC CALCULI OF PROCESSES AND NET EXPRESSIONS

Georges GONTHIER

Février 1985

Algebraic Calculi of Processes and Net Expressions

Georges Gonthier

CMA-ENSMP Sophia-Antipolis
06565 Valbonne CEDEX

ABSTRACT

It is shown that the Meije algebra of processes can be expressed in a flow algebra, the algebra of Net Expressions. We then use this fact to derive a normal form for the calculus.

RESUME

Nous montrons que l'algèbre de processus Meije peut être exprimée dans une "flow algebra", celle des expressions de Réseaux. Nous en déduisons une forme normale pour les termes de ce calcul.

Algebraic Calculi of Processes and Net Expressions

Georges Conthier

CMA-ENSMP Sophia-Antipolis
06565 Valbonne CEDEX

Introduction

Perhaps one of the most natural ways to model a parallel sysytem is to use a graphic approach, representing such a system by a communication network whose nodes are processes. It is the description of such a model –Flowgraphs– that motivated the first definition of algebras of processes –Flow algebras– by R.Milner [Mil1979a]. Dynamic operators, action and summation, were then added to this algebra, together with a direct semantic by the means of Plotkin-style operational rewrite rules. This, and a definition of equivalence through bisimulation, had become the algebraic calculus CCS [Mil1980a]. Finally, the interleaving constraints of CCS were solved by the introduction of synchrony and action simultaneity in SCCS [Mil1983a].

Many properties were gained through these improvements, but the original intuitive interpretation as Nets was lost. Our purpose here is to show how it may be restored. In order to do this, we will work on a SCCS sibling calculus, the Meije calculus [Bou1984a]. It has been shown that Meije is equivalent to the "computable" SCCS (see [Bou1984b, Sim1984a] for a precise comparison), indeed that it is in some way a "universal" calculus [Sim1984b]. Thus our precise goal will be to show that all Meije agents can be expressed as Flow expressions (we shall call them Net expressions) over an elementary set of generators in some semantic-preserving way.

Meije operators are very close to Net constructors, in fact they may be considered as a superset of these. Thus our method will be to gradually eliminate by structural transformations unwanted operators, concentrating their semantics in the generators, while remaining equivalent to the original expression.

In this way we shall assign an equivalent Net expression to each Meije agent; this will enable us to transfer properties of the former to the latter. In a last paragraph we derive a Normal Form for Meije agents from the Normal Form Theorem for Flow Expressions.

1. Definitions

1.1. Net Expressions

A *Net* is a synchronizing structure, whose *sites* (or *nodes*) are linked through *ports* to communication *channels*. An *Actor Net* is a Net whose sites are occupied by processes, or *actors*, taken from a set of *generators*.

In [Mil1979a] , R.Milner showed that such objects could be described by *Net Expressions* constructed from a free algebra. In order to do this we will give ourselves a set L of channel names, which is the disjoint union of a set **S** of port names, and of the set Λ of signal names. Let also X be a set of process variables. All sets are assumed infinite. We then define the operators of:

-(parallel) composition $/\!\!/$. Let $\Sigma_{/\!\!/}=\{/\!\!/\}$.

-finite relabelling ψ. Let $\Sigma_{\psi}=\{\psi:L\to L \mid P_{\psi}=\{s\in L \mid \psi s\neq s\} \text{ is finite}\}$.
$<s'_i/s_i>_{i\in I}$ will denote the relabelling ψ such that $\psi s=s'_i$ if $s=s_i$, $\psi s=s$ otherwise.

-signal restriction $\backslash\lambda$. Let $\Sigma_{\backslash}=\{\backslash\lambda \mid \lambda\in\Lambda\}$.

-recursive definition $\mathbf{where}_{X'}$. Let $\Sigma_{\mathbf{where}}=\{\mathbf{where}_{X'} \mid X'\subset X \text{ is finite}\}$. In expressions we use the syntax $(t \ \mathbf{where}\ \{x_i=t_i\}_{x_i\in X'})$.

Net expressions are then terms of the free Σ_0-algebra $\mathbf{A}_{\Sigma_0}(X)$, where $\Sigma_0=\Sigma_{/\!\!/}\cup\Sigma_{\psi}\cup\Sigma_{\backslash}\cup\Sigma_{\mathbf{where}}$.

The operator $\backslash\lambda$ (resp. ψ, $\mathbf{where}_{X'}$) binds signal λ (resp. signals of $P_{\psi}\cap\Lambda$, variables of X'). Terms are equal if identical up to change of bound signal or variable. As a consequence, when we apply a substitution σ to a term t (which we shall denote by $t[\sigma]$), we change bound signals or variables in order to avoid captures.

Our operator set is in fact parametrized by **S**, Λ and X. In some cases we shall make use of this, and, when there is some ambiguity, we will indicate in parentheses the sort used (e.g. $\Sigma(S')$). Note that if a relabelling and its extension by the identity to a larger sort are identified, $\mathbf{S}'\subset\mathbf{S}$ implies $\Sigma_{\mathbf{S}'}\subset\Sigma_{\mathbf{S}}$.

Finally, if **C** is a generator set, then an actor net on **C** is described by a term of $\mathbf{A}_{\Sigma_0}(\mathbf{C})$.

1.2. The Meije Calculus

Here the expressions are meant to describe transition systems labelled by a set **M** of actions. Following Milner's idea[Mil1983a], we shall assume **M** is a commutative monoid, the product of two actions having their joint effect, the unity being an action with no outside effect. As we intend to work with communication structures, having only communication actions is quite natural. Then every action has an inverse, the action of "receiving" it; thus **M** is an abelian group. Since we want to model independant channels, we shall in fact work with $\mathbf{M}=\mathbf{Z}<L>$, the free group generated by L. The operation will be noted as a product, the inverse of s s^-, and, if $u\in\mathbf{M}$ and $s\in L$ then $u(s)$ is the valuation of s in

u, i.e. $u=s^{u(s)}u'$ with $u'\in\mathbf{Z}<L-\{s\}>$.

Meije [Bou1984b] can then be constructed from a superset of the Net operators, adding:

-action u :. Let $\Sigma_{:}=\{u:\mid u\in\mathbf{M}\}$.

-ticking $u*$. Let $\Sigma_{*}=\{u*\mid u\in\mathbf{M}\}$.

-(finite) renaming φ. Let $\Sigma_{\varphi}=\{\varphi:\mathbf{M}\to\mathbf{M}$ a morphism $\mid P_{\varphi}$ is finite$\}$. A relabelling will be identified with the morphism it induces, so $\Sigma_{\psi}\subset\Sigma_{\varphi}$.

The only generator used will be inaction $\mathbf{0}$, thus a Meije term will be a term of $\mathbf{A}_{\Sigma_1}(\{\mathbf{0}\}\cup X)$, where $\Sigma_1=\Sigma_0\cup\Sigma_{:}\cup\Sigma_{*}\cup\Sigma_{\varphi}$. A Meije agent is a Meije term closed for variables and signals.

1.3. Semantics

We will now give our Meije terms their transition-system semantics by the following operational rewrite rules:

$$\frac{p\xrightarrow{u}p'}{p/\!/q\xrightarrow{u}p'/\!/q}\ (\text{Rpar1}),\qquad \frac{q\xrightarrow{u}q'}{p/\!/q\xrightarrow{u}p/\!/q'}\ (\text{Rpar2}),\qquad \frac{p\xrightarrow{u}p',\ q\xrightarrow{u}q'}{p/\!/q\xrightarrow{uu}p'/\!/q'}\ (\text{Rpar12})$$

We thus have an asynchronous composition.

$$u:p\xrightarrow{u}p\ (\text{Ract}),\qquad \frac{p\xrightarrow{u}p',\ u(\lambda)=0}{p\backslash\lambda\xrightarrow{u}p'\backslash\lambda}\ (\text{Rres})$$

$$\frac{p\xrightarrow{v}p'}{u*p\xrightarrow{uv}u*p'}\ (\text{Rtic}),\qquad \frac{p\xrightarrow{v}p'}{\varphi p\xrightarrow{\varphi u}\varphi p'}\ (\text{Rren})$$

and finally

$$\frac{t[q_i/\!/x_i]_{x_i\in X'}\xrightarrow{u}p'}{(t\ \textbf{where}\ \{x_i=t_i\}_{x_i\in X'})\xrightarrow{u}p'}\ (\textit{Rrec})\ \text{with}\ q_i=t_i[(x_j\ \textbf{where}\ \{x_k=t_k\}_{x_k\in X'})/\!/x_j]_{x_j\in X'}$$

for recursion.

Note then that if generators of **C** are given an **Z<S>**-labelled transition system semantics, then these rules may also be used to give such a semantic to Nets built on these generators.

The semantic obtained in this way for Meije is too refined; we thus need to introduce an equivalence, the strong equivalence $\sim$, defined as the least refined symetric relation R verifying

$$\forall p,q,u,p',\ p\xrightarrow{u}p'\text{ and }pRq\ \Rightarrow\ \exists q',\ q\xrightarrow{u}q'\text{ and }p'Rq'$$

(thus it is Park's bisimulation).

We extend it to non-closed terms by

$$t\simeq t'\iff\forall\sigma\text{ closed for }t\text{ and }t',\quad t[\sigma]\sim t'[\sigma]$$

(thus a term denotes an operator on transition systems).

Now a transition system t can be expressed in a set of transition systems **P** iff there exists some $p \in \mathbf{P}$ such that $t \sim p$; this notion is readily extended to sets of transition systems, objects having such a semantic, and operators on transition systems. For example let the clock h_u on $u \in \mathbf{M}$ be the system defined by $h_u \xrightarrow{u} h_u$, and the unbounded clock h_u^+ by $h_u^+ \xrightarrow{u^k} h_u^+$ if $k>0$. Then both can be expressed in Meije, for

$$h_u \sim (x \textbf{ where } x = u : x) \quad \text{and} \quad h_u^+ \sim (x \textbf{ where } x = (u : \mathbf{0} /\!\!/ x))$$

In fact it has been shown that any computable transition system (on a finitely generated monoid), can be expressed in Meije [Bou1984b]. Thus any computable Net, with the above semantics, can be expressed in Meije. What we now need to show is the converse, that is that any Meije agent can be expressed by Net Expressions over a simple set of generators (basically clocks).

2. Expressing Meije in Net Expressions

2.1. Operator substitution

Technically what we want to show is that $\mathbf{A}_{\Sigma_1}(\{\mathbf{0}\})$ can be expressed in $\mathbf{A}_{\Sigma_0}(\mathbf{C})$. We shall proceed in several steps, showing each time that an $\mathbf{A}_\Sigma(\mathbf{C})$ can be expressed in some $\mathbf{A}_{\Sigma'}(\mathbf{C}')$, with $\Sigma' \subset \Sigma$ and $\mathbf{C} \subset \mathbf{C}'$, i.e. we remove operators and add constants.

In order to show each step, we will define structural transformations, replacing operators of $\Sigma \cup \mathbf{C}$ by contexts of the new algebra, in which we will place the operator's transformed arguments (while taking care of bindings). Let us immediately note that such a transformation ϑ will be compatible with substitution, i.e. $\vartheta(t[p_i/x_i]) = \vartheta(t)[\vartheta(p_i)/x_i]$.

2.2. Eliminating Action

Let $\mathbf{C}_2 = \{v : h_u, h_u \mid u, v \in \mathbf{Z}<\mathbf{S}>\} \cup \{\mathbf{0}\}$ be the set of possibly actionned clocks, with their obvious semantics; let $\Sigma_2 = \Sigma_1 - \Sigma_:$. Then

Proposition : $\mathbf{A}_{\Sigma_1}(\{\mathbf{0}\})$ can be expressed in $\mathbf{A}_{\Sigma_2}(\{\mathbf{C}_2\})$

proof :

Let us define by structural induction $\vartheta_2 : \mathbf{A}_{\Sigma_1}(\{\mathbf{0}\} \cup X) \to \mathbf{A}_{\Sigma_2}(\{\mathbf{C}_2\} \cup X)$:

$\vartheta_2(u : t) = (u : h_\alpha \text{-} /\!\!/ \alpha * \vartheta_2(t)) \backslash \alpha$ where α is not free in $u : t$,

and $\vartheta_2(\mathrm{op}(\vec{t_i})) = \mathrm{op}(\overrightarrow{\vartheta_2(t_i)})$ for other operators "op" ($\vartheta_2(x) = x$ for variables).

Here and in the other constructions a generator sorted on L is considered to be a notation for a generator sorted on $\mathbf{S}$ (e.g. $h_u \equiv \psi h_{u'}$, with $\psi u' = u$ and $u' \in \mathbf{Z}<\mathbf{S}>$).

Then $\vartheta_2(u : x) \simeq u : x$ and $\vartheta_2(u : t) = \vartheta_2(u : x)[\vartheta_2(t)/x]$,
and since $\simeq$ is a congruence, by structural induction,

$\vartheta_2(t) \simeq t$ for all t,

and if t is closed, then $\vartheta_2(t) \sim t$, Q.E.D.

2.3. Eliminating Ticking

This is the central part of our proof. Let $\Sigma_3 = \Sigma_2 - \Sigma_*$; we first show:

Lemma : if $\mathbf{S}' \subset \mathbf{S}$, $s \in S - S'$, then $s * \mathbf{A}_{\Sigma_3(\mathbf{S})}(\mathbf{C}_2(\mathbf{S}'))$ can be expressed in $\mathbf{A}_{\Sigma_3(\mathbf{S})}(\mathbf{C}_2(\mathbf{S}))$.

proof :

We define ϑ_3 by :

$\vartheta_3(v : h_u) = s.v : h_{s.u}$, $\qquad \vartheta_3(h_u) = h_{s.u}$

$\vartheta_3(t_1 /\!\!/ t_2) = (h_{s\alpha} \text{-} /\!\!/ (h_{s\alpha} \text{-} 2 /\!\!/ <\alpha/s> \vartheta_3(t_1)) /\!\!/ <\alpha/s> \vartheta_3(t_2)) \backslash \alpha$

(where α is not free in $t_1 /\!\!/ t_2$)

$\vartheta_3(u * t) = <s.u/s> \vartheta_3(t)$

$$\vartheta_3(\mathrm{op}(\vec{t_i})) = \mathrm{op}(\overrightarrow{\vartheta_3(t_i)}) \quad \text{otherwise}$$

Now we will show, using Park induction, that $s *t \sim \vartheta_3(t)$ for all $t \in \mathbf{A}_{\Sigma_2(S')}(\mathbf{C}_2(\mathbf{S}'))$;
thus let $R = \{<s *t, \vartheta_3(t)>, <\vartheta_3(t), s *t> \mid t \in \mathbf{A}_{\Sigma_2(S')}(\mathbf{C}_2(\mathbf{S}'))\}$.

then if $s *t \xrightarrow{u} t'$, then it is by (Rtic), so $u = s.u'$, $t' = s *t''$ and $t \xrightarrow{u'} t''$, therefore $u'(s) = 0$.

Now all we need is to show by induction on the proof of $t \xrightarrow{u} t'$ that $\vartheta_3(t) \xrightarrow{s.u} \vartheta_3(t')$. For most rules the proof is straightforward, let us deal with (Rrec), which is a little delicate:

if $(t \textbf{ where } \{x_i = t_i\}_{x_i \in X'}) \xrightarrow{u} t'$, then it is by (Rres), so, using previous notations

$t[q_i/x_i]_{x_i \in X'} \xrightarrow{u} t'$, so by induction

$\vartheta_3(t[q_i/x_i]_{x_i \in X'}) \xrightarrow{s.u} \vartheta_3(t')$

now

$$\vartheta_3(t[q_i/x_i]_{x_i \in X'}) = \vartheta_3(t)[\vartheta_3(q_i)/x_i]_{x_i \in X'}$$

and

$$\vartheta_3(q_i) = \vartheta_3(t_i)[(x_j \textbf{ where } \{x_k = \vartheta_3(t_k)\}_{x_k \in X'})/x_j]_{x_j \in X'}$$

so by (Rrec)

$$(\vartheta_3(t) \textbf{ where } \{x_i = \vartheta_3(t_i)\}_{x_i \in X'}) \xrightarrow{s.u} \vartheta_3(t')$$

that is,

$$\vartheta_3\left[(t \textbf{ where } \{x_i = t_i\}_{x_i \in X'})\right] \xrightarrow{s.u} \vartheta_3(t'),$$

Q.E.D.

The symetrical case ($\vartheta_3(t) \xrightarrow{u} t'$ implies $s *t \xrightarrow{u} s *t''$ and $t' = \vartheta_3(t'')$) is proven in the same way (and left to the reader).

Now we can deduce

Proposition: $\mathbf{A}_{\Sigma_2}(\mathbf{C}_2)$ can be expressed in $\mathbf{A}_{\Sigma_3}(\mathbf{C}_2)$

proof:

Let $t \in \mathbf{A}_{\Sigma_2}(\mathbf{C}_2)$, then since t is finite, we actually have $t \in \mathbf{A}_{\Sigma_2(S')}(\mathbf{C}_2(\mathbf{S}'))$, with finite $\mathbf{S}'$; now let $s \in \mathbf{S} - \mathbf{S}'$, since $s *t$ can be expressed in $\mathbf{A}_{\Sigma_3(S)}(\mathbf{C}_2(\mathbf{S}))$, then so can $\vartheta'_3(t) = (h_\alpha \text{-} /\!\!/ <\alpha/s> s *t) \backslash \alpha$, (α is not free in t)
and $t \sim \vartheta'_3(t)$, Q.E.D.

2.4. Reducing Renaming to Relabelling

Let $C_4=C_2\cup\{s*h_u^+ \mid s\in S,\ u\in Z<S>\}$, be given its obvious semantics.

Proposition: $A_{\Sigma_3}(C_2)$ can be expressed in $A_{\Sigma_0}(C_4)$

proof:

In the wake of Milner[Mil1983a], we define ϑ_4 by

$$\vartheta_4(\varphi t)=\left(\mathop{/\!/}_{s\in N_\varphi}(h_\alpha\text{-}/\!/\alpha*h_{\beta_s}\text{-}.\varphi s/\!/\alpha*h_{\beta_s}\text{-})\backslash\alpha/\!/\psi\vartheta_4(t)\right)\backslash\{\beta_s\}_{s\in N_\varphi}$$

where $N_\varphi=\{s\in L \mid \varphi s\notin L\}$, α, β_s, are all distinct and not free in φt, and ψ is the relabelling defined by $\psi s=\beta_s$ if $s\in N_\varphi$, $\psi s=\varphi s$ otherwise;
ϑ_4 is the identity on other operators.

Now $\vartheta_4(\varphi x)\simeq\varphi x$, and ϑ_4 conserves bindings, therefore by a staightforward adaptation of Milner's proof that $\simeq$ is a congruence, we have $\vartheta_4(t)\simeq t$ for all t, Q.E.D.

Now from the last three paragraphs we can immediately deduce:

Theorem:Any Meije agent can be expressed as a Net Expression on clocks
more precisely, $A_{\Sigma_1}(\{0\})$ can be expressed in $A_{\Sigma_0}(C_4)$.

3. Normal Forms

3.1. On Minimal Generators

In the preceding transformations, as the reader might have noticed, we took care not to modify the equational structure of our terms; in particular, we added no recursive definitions. If we allow ourselves such additions, then we can express generators of C_4 as Net Expressions over an even more elementary set of generators, finite deterministic terms:

Let $C_5=\{u:v:\mathbf{0},v:\mathbf{0},\mathbf{0} \mid u,v\in Z<S>\}$, then

Proposition: C_4 can be expressed in $A_{\Sigma_0}(C_5)$, thus any Meije agent can be expressed as a Net over finite generators.

proof:

One may verify that, if $s\in S$, $u,v,w\in Z<S>$ and $u(s)=v(s)=w(s)=0$, we have:

$$(u:)v*h_w^+ \sim$$
$$((u:)v.\alpha:\mathbf{0}/\!\!/<\alpha/s>x)\backslash\alpha \text{ where } \{x=(s^-.w.\alpha:\mathbf{0}/\!\!/s^-.w:v.\alpha:\mathbf{0}/\!\!/<\alpha/s>x)\backslash\alpha\}$$

Now the result follows from $u:h_v\sim u:v*h_1^+$ and $h_u\sim u*h_1^+$.

3.2. On Minimal Algebras

The generators of C_5 may be described by Meije terms, more precisely by terms of $A_{\Sigma_\cdot}(\{\mathbf{0}\})$. Thus the last result implies

Corollary: Meije agents can be expressed in $A_{\Sigma_0}(A_{\Sigma_\cdot}(\{\mathbf{0}\}))$

As a consequence, Meije can be expressed in $A_{\Sigma_0\cup\Sigma_\cdot}(\{\mathbf{0}\})$; thus morphisms and ticking are non-primitive operators! This result, while natural for morphisms (see [Mil1983a]), seems surprising for ticking. In fact, what we have shown is that ticking is not essential to the *computing* power of the calculus; it still remains primitive for the *expressive* power (for building operators), as we have:

Proposition: Ticking cannot be reduced to an expression of other operators, that is, for any $u\neq 1$, there is no expression $E\in A_{\Sigma_1-\Sigma_\cdot}(\{\mathbf{0},x\})$ such that $E\simeq u*x$.

proof:

Let us assume such an E exists. Now

-Either all the occurences of x are guarded in E (or guarded after a finite "unwinding" – see [Mil1983a]); since

$u*x[1:\mathbf{0}/x] \xrightarrow{u}$, we have

$E[1:\mathbf{0}/x] \xrightarrow{u}$, but since x is guarded in E,

$E[\mathbf{0}/x] \xrightarrow{u}$, absurd since $u*\mathbf{0}\sim\mathbf{0}$.

-Or there is some unguarded occurence of x in E. Then by straightforward induction on the depth of that occurence, we show $E[1:\mathbf{0}/x] \xrightarrow{1}$, which contradicts $E[1:\mathbf{0}/x] \sim_u *1:\mathbf{0}$.

3.3. Normalizing Recursive Equations

The corollary in the last paragraph gives a more precise result than just a reduction of the operator set; it also shows that the *syntax* can be restricted, so that we will now have not a single free algebra, but a *stack* of algebras, which can be termed a *normal form*. In a second step towards such a form, we show how to regroup equations.

Proposition: Meije can be expressed in $\Sigma_{\mathbf{where}}(A_{\Sigma_0-\Sigma_{\mathbf{where}}}(A_{\Sigma}(\{\mathbf{0}\})\cup X))$, i.e. is equivalent to a single equation system, a term of the form $(t_0 \mathbf{\ where\ } \{x_i=t_i\}_{i>0})$ where there are no recursions in the t_i's.

Proof:

We will actually show that for all finite S'⊂S, that $A_{\Sigma_0(S')}(A_{\Sigma(S')}(\{\mathbf{0}\}))$ can be expressed in $\Sigma_{\mathbf{where}}(S)(A_{(\Sigma_0-\Sigma_{\mathbf{where}})(S)}(A_{\Sigma(S)}(\{\mathbf{0}\})\cup X))$; the finiteness of terms will thus imply our stated result.

Now let $p \in A_{\Sigma_0(S')}(A_{\Sigma(S')}(\{\mathbf{0}\}))$, we define structurally the regrouping $\mu(p)$ of p by

$$\mu(p^1 /\!\!/ p^2) = \left| t_0^1 /\!\!/ t_0^2 \mathbf{\ where\ } \{x_j^i = t_j^i\}_{\substack{i\in[1,2] \\ j\in[1,n_i]}} \right|$$

$$\mu(p^0 \mathbf{\ where\ } \{x_0^i = p^i\}_{i\in[1,n]}) = \left| x_0^0 \mathbf{\ where\ } \{x_j^i = t_j^i\}_{\substack{i\in[1,n] \\ j\in[1,n_i]}} \right|$$

where x_j^i are all distinct, and $\mu(p^i)=(t_0^i \mathbf{\ where\ } \{x_j^i=t_j^i\}_{j\in[1,n_i]})$, up to change of bound variables.

$$\mu(\mathrm{op}(p))=(\mathrm{op}(t[\psi x_i/x_i]_{i\in[1,n]}) \mathbf{\ where\ } \{x_i=\psi(t_i[\psi x_j/x_j]_{j\in[1,n]})\}_{i\in[1,n]})$$

where ψ is a relabelling that maps the signals binded by op onto a subset of S−S', and vice-versa (i.e. $\psi=\psi^{-1}$), and $\mu(p)=(t \mathbf{\ where\ } \{x_i=t_i\}_{i\in[1,n]})$

$$\mu(x)=(x \mathbf{\ where\ } \phi), \quad \mu(\mathbf{0})=(\mathbf{0} \mathbf{\ where\ } \phi)$$

It is readily shown that for closed t, $\mu(t)\sim t$, and it has the announced form.

3.4. A Normal Form for Meije

Theorem: Meije can be expressed in $\Sigma_{\mathbf{where}}(\mathbf{A}_{\Sigma\backslash}(\mathbf{A}_{\Sigma/}(\Sigma_{\psi}(X \cup \mathbf{A}_{\Sigma:}(\{\mathbf{0}\})))))$, that is every Meije agent is equivalent to a term $(t_0 \textbf{ where } \{x_i = t_i\})$, where for all i, $t_i \equiv (/\!\!/ \psi_{ij} k_{ij}) \backslash \Gamma_i$, and k_{ij} is either a variable or a term $u_{ij1} : \cdots : \mathbf{0}$. So any Meije agent is described by a normalized set of equations.

proof:

The Laws of Flow are valid in $\mathbf{A}_{\Sigma_0 - \Sigma_{\mathbf{where}}}(X \cup \mathbf{A}_{\Sigma:}(\{\mathbf{0}\}))$, therefore the present theorem is an immediate consequence of the preceding paragraphs and the Normal Form theorem for Flow Algebras.

Acknowledgements

Many thanks to Gérard Boudol, who provided invaluable guidance in the elaboration of this note.

Bibliography

Bou1984a. G. Boudol and D. Austry, "Algèbre de Processus et Synchronisation," *T.C.S.* **30**(1) (1984).

Bou1984b. G. Boudol, "Notes on Algebraic Calculi of Processes," in *Proceedings of the Advanced Course "Logics and Models for Verification and Specifications of Concurrent Systems"*, ed. K.R. Apt, Springer-Verlag (1984).

Mil1979a. R. Milner, "Flowgraphs and Flow Algebras," *J. ACM* **26** (1979).

Mil1980a. R. Milner, *A Calculus of Communicating Systems*, Springer-Verlag (1980).

Mil1983a. R. Milner, "Calculi for Synchrony and Asynchrony," *T.C.S.* **25**(3) (1983).

Sim1984a. R. de Simone, "On Meije and SCCS: Infinite Sum Operators vs Non-Guarded Definitions," *T.C.S.* **30**(1), Note (1984).

Sim1984b. R. de Simone, *Calculabilité et expressivité dans l'algèbre de processus parallèles Meije*, Jussieu Paris 7 (1984). Thèse de 3 ème cycle

Imprimé en France

par

l'Institut National de Recherche en Informatique et en Automatique