An asynchronous parallel interpreter for arithmetic expressions and its evaluation

François Baccelli, Philippe Mussi

INRIA

CENTRE DE ROCQUENCOURT

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° 303

AN ASYNCHRONOUS PARALLEL INTERPRETER FOR ARITHMETIC EXPRESSIONS AND ITS EVALUATION

François BACCELLI
Philippe MUSSI

Mai 1984

F. BACCELLI

INRIA
Domaine de Voluceau
Rocquencourt
BP 105
78153 Le Chesnay Cedex
France

Ph. MUSSI

ENS St Cloud
2 avenue du Palais
92210 Saint Cloud
France

AN ASYNCHRONOUS PARALLEL INTERPRETER FOR ARITHMETIC EXPRESSIONS AND ITS EVALUATION

ABSTRACT

We define in an actor-like formalism a recursive parallel interpreter for arithmetic INFIX grammars. The resulting speed up and efficiency are then predicted via a probabilistic model based on stochastic grammars.

RESUME

Nous definissons, dans un langage d'acteurs, un algorithme parallèle récursif pour l'interprétation de grammaires arithmétiques infixées. L'accélération et l'efficacité obtenues sont étudiées au moyen d'un modèle de grammaire stochastique.

INTRODUCTION

The design of compilers (interpreters) for multiprocessing environment has been considered under several complementary apsects. One of them concerns the transformations of the sequencial structure Scanning-Parsing- Semantics into a pipe line of three (sequencial) procedures. ([BAE 77]).

Another set of problems consists of the parallelization of each of these three procedures (see [FIS 80] for parsing and for code generation and [BAC 82] for parsing). The present paper is concerned with the integration of there two complementary approaches in the case of the interpretation of an elementary INFIX arithmetic grammar involving only purely synthetized semantics. In the first part of the paper, we use an actor-like formalism for defining the general recursive structure of the interpreter. In the second one, we derive from this recursive structure a set of fixed point equations for predicting the average speed up resulting form the parallelization.

However both the algorithmic structure and the performance evaluation schema can be generalized to the parallelization of compilers (or interpreters) of other grammars provided their semantics remain purely synthetized.

PART 1

The parallelization we propose in the present paper is based on a "syntactic cutting" of the input string when in the former approaches of [MIC 78] and [FIS 80], the authorsconsidered a "regular interval cutting" and a "lexical cutting" respectively. The reason for this choice is that a syntactic cutting allows more parallelism than a lexical one (as proven in [BAC 82]) and however, does not require such extended automata as in [MIC 78] : the atomic tasks (actor's code) to be executed here will just consist of subprocedures of the sequential interpreter.

In the first section, we recall the definition and some properties of the syntactic decomposition (cutting) of the derivation trees (sentences) of our elementary arithmetic infix grammar. This decomposition determines from the parenthesis nesting of an arithmetic expression, a cover of its derivation tree by a set of subtrees with known syntactic properties.

The second section contains the definition of a recursive interpreter based on this decomposition. Roughly speaking, for a given arithmetic expression, this last will create one actor for each of the subtrees in its derivation tree decomposition. Each created actor will be in charge of local (i.e. concerning its own subtree only) parsing and semantic tasks which involve synchronization with the other actors.

SECTION 1

1.1. - THE ARITHMETIC INFIX GRAMMAR

1.1.1. - Notations

We shall use the following notation for a context free grammar a : $G = V_N, V_T, P, S$ and shall denote by L(a) the language generated by a.

1.1.2. - Arithmetic infix grammar

The terminals are underlined :

$V_T = \{\underline{+}, \underline{-}, \underline{*}, \underline{/}, \underline{**}, \underline{\surd}, \underline{(}, \underline{)}, \underline{id},\}$

$V_N = S, M_i \ (i = 1,7)$

P :

(1)	$S \rightarrow \underline{(}M_1\underline{)}$		
(2(j))	$M_1 \rightarrow M_j$	$j \geq 2$	
(3(j,k,$\overline{+}$))	$M_2 \rightarrow M_j \underline{+} M_k \mid M_j \underline{-} M_k$	$j \geq 2$	$k > 2$
(4(j,k,*))	$M_3 \rightarrow M_j \underline{*} M_k \mid M_j \underline{/} M_k$	$j \geq 3$	$k > 3$
(5(j,k))	$M_4 \rightarrow M_j \underline{**} M_k$	$j > 4$	$k \geq 4$
(6(j))	$M_5 \rightarrow \underline{\surd} M_j$	$j \geq 5$	
(7)	$M_6 \rightarrow \underline{(}M_1\underline{)}$		
(8)	$M_7 \rightarrow \underline{id}$		

G is unambiguous.

1.2. - PARENTHESIS SKELETON OF AN ARITHMETIC EXPRESSION

Let $w \in L(G)$, w may be written in a unique manner as :

$$w = d_1\gamma_1\ d_2\gamma_2 \cdots \gamma_{j-1}\ d_j\ \gamma_j \cdots \gamma_m\ d_{m+1}$$

where $d_i \in D \triangleq \{(,)\}$ and $\gamma_j \in (V_T - D)^*$.

Let p_j be the position of d_j in w, $1 \leq p_j \leq |w|$.

Let n_j be the nesting level of d_j in w, $1 \leq n_j \leq |w|$. We define δ_j as the triple :

$$\delta_j = (d_j, p_j, n_j) \in D \times \mathbb{N} \times \mathbb{N}$$

We call parenthesis skeleton of w the sequence

$$\mathcal{D}(w) \triangleq \{\delta_1, \ldots, \delta_{m+1}\}$$

We denote m+1 as $|\mathcal{D}(w)|$.

1.3. - DECOMPOSITION OF THE DERIVATION TREES

We define a decomposition which determines, for each derivation tree $\mathcal{A}$, a unique sequence of subtrees $\mathcal{A}_i$, i = 1, n($\mathcal{A}$) being all derivation trees in a known grammar $\hat{G}$ and such that the sequence $\mathcal{A}_i$, i = 1,n($\mathcal{A}$) determines completely $\mathcal{A}$.

1.3.1. - Definition

Let $W = \{S, M_6\} \subset V_N$. Let $\mathcal{A}$ be a derivation tree in G. Let $A_1, \ldots, A_{n(\mathcal{A})}$ be the sequence of all the nodes of $\mathcal{A}$ carrying a label which belongs to W ; their order being defined by a preorder traversal of $\mathcal{A}$.

We define the W-decomposition of $\mathcal{A}$ as the sequence $(\mathcal{A}_1, \ldots, \mathcal{A}_{n(\mathcal{A})})$ where $\mathcal{A}_i$, $i = 1, n(\mathcal{A})$, is the largest subtree of $\mathcal{A}$ rooted by A_i when considering all the nodes of $\mathcal{A}$ (but A_i) carrying labels in W as terminals.

1.3.2. - Example

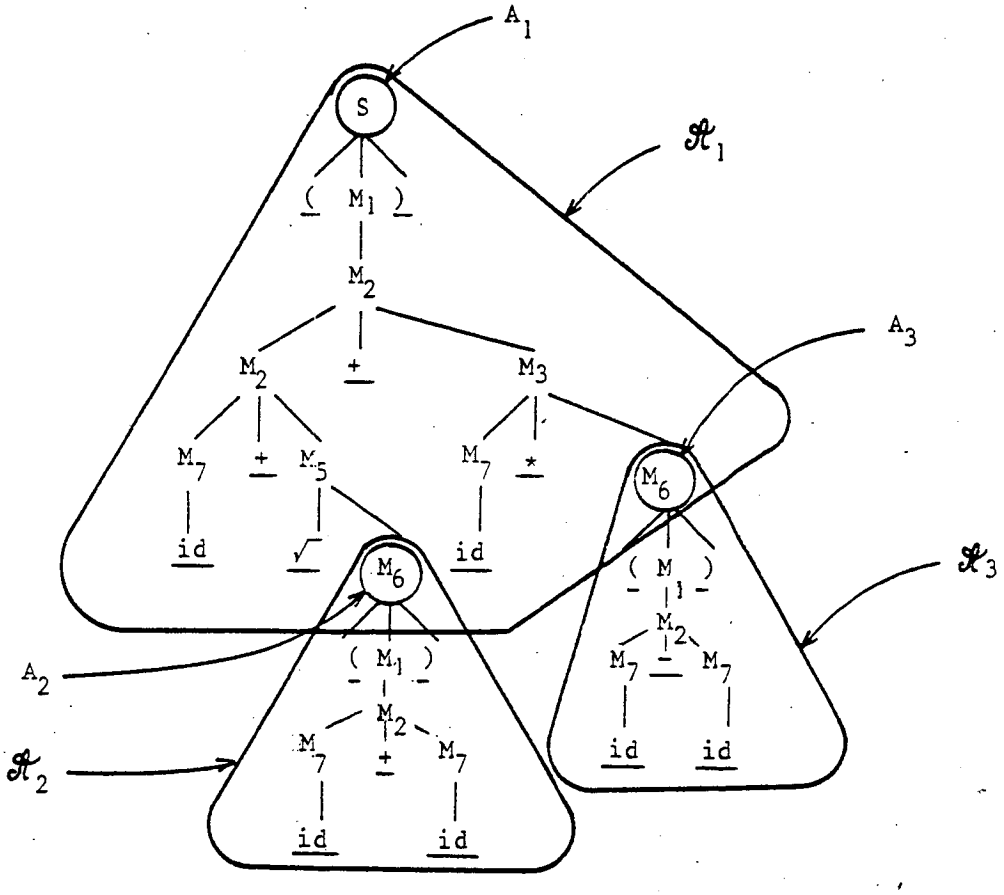


Figure 1

1.3.3. – Some syntactic properties

Let $\hat{G}$ be the following C.F. unambiguous grammar (the terminals are underlined) :

$$\begin{cases} \hat{V}_T = \{\underline{M_6}, \underline{+}, \underline{-}, \underline{*}, \underline{/}, \underline{**}, \underline{\surd}, \underline{(,)}, \underline{id}\} \\ \hat{V}_N = \{\hat{S}, M_j, j = 1,7, j \neq 6\} \\ \hat{P} = \{\hat{S} \to \underline{(}M_1\underline{)} \; ; \; M_1 \to M_j, \quad j = 1,7, j \neq 6 \; ; \; M_1 \to \underline{M_6} \; ; \end{cases}$$

$$M_2 \to M_j \underline{+} M_k,\ j \geq 2,\ k > 2,\ j \neq 6,\ k \neq 6 \; ;$$

$$M_2 \to \underline{M_6} \underline{+} M_k,\ k > 2,\ M_2 \to M_j \underline{+} \underline{M_6},\ j \geq 2 \; ;$$

$$M_3 \to M_j \underline{\surd} M_k,\ j \geq 3,\ k > 3,\ j \neq 6,\ k \neq 6 \; ;$$

$$M_3 \to \underline{M_6} \underline{/} M_k,\ k > 3,\ M_3 \to M_j \underline{*} \underline{M_6},\ j \geq 3 \; ;$$

$$M_4 \to M_j \underline{**} M_k,\ j > 4,\ k \geq 4,\ j \neq 6 \; ;$$

$$M_4 \to \underline{M_6} \underline{**} M_k,\ k \geq 4,\ M_4 \to M_j \underline{**} \underline{M_6},\ j > 4 \; ;$$

$$M_5 \to \underline{\surd} M_j,\ j = 5,7 \; ; \; M_5 \to \underline{\surd M_6} \; ; \; M_7 \to \underline{id}.\}$$

The following properties are satisfied :

i) For each $i = 1, n(\mathscr{A})$, $\mathscr{A}_i$ is a derivation tree in $\hat{G}$.

ii) $S \xrightarrow{\psi_1} \mathscr{B}_1 \xrightarrow{\psi_2} \mathscr{B}_2 \ldots \mathscr{B}_{n-1} \xrightarrow{\psi_n} \mathscr{B}_n = \mathscr{A}$

where $\mathscr{B}_{i-1} \xrightarrow{\psi_i}$ consists in replacing the leftmost leaf of the tree $\mathscr{B}_{i-1}$ belonging to W (that is A_i) by the subtree $\overset{A_i}{|}\!\!\underset{\Omega_i}{|}$ if $\mathscr{A}_i = \overset{\hat{S}}{|}\!\!\underset{\Omega_i}{|}$

Let w_i be the sentence of $\mathscr{A}_i$ considered as a derivation tree in $\hat{G}$. The two properties above show that the knowledge of the sequence w_i, $i = 1, n(\mathscr{A})$ is sufficient for building up $\mathscr{A}$: it is possible to determine $\mathscr{A}_i$ from w_i by a parsing in $\hat{G}$ (i) and then to build up $\mathscr{A}$ from the sequence $\mathscr{A}_i$, $i = 1, n(\mathscr{A})$ as indicated by (ii).

1.3.4. - How to build subsentences from nesting

Let $w \in L(G)$ and $\mathcal{A}$ its derivation tree in G. For $n < r$, $\in \{1, \ldots \mathcal{D}|w|\}$ we denote as (d_n, d_r) (resp $[d_n, d_r]$) the substring $\gamma_n\ d_{n+1}\ \gamma_{n+1} \cdots d_{r-1}\ \gamma_{r-1}$ (resp. $d_n\ \gamma_n\ d_{n+1} \cdots d_{r-1}\ \gamma_{r-1} d_r$) of $w \equiv d_1\ \gamma_1\ d_2 \cdots \gamma_m\ d_{m+1}$.

Let ℓ_i (resp. r_i) be the leftmost (resp. rightmost) successor of A_i in $\mathcal{A}$, $i = 1, n(\mathcal{A})$. Since ℓ_i and $r_i \in D$, $\exists\, j_i, j_i' \in \{1, \ldots, |\mathcal{D}(w)|\}$ such that $\ell_i = d_{j_i}$, $r_i = d_{j_i'}$.

Let α_i be the substring of w rooted by A_i. We have $\alpha_i = [d_{j_i}, d_{j_i'}]$. Let $\mathcal{D}(\alpha_i)$ be the sequence :

$$\mathcal{D}(\alpha_i) = \{\delta_{j_i}, \delta_{j_i+1}, \ldots, \delta_{j_i'}\}$$

Let w_i be the sentence associated to $\mathcal{A}_i$ considered as a derivation tree in $\hat{G}$. w_i is determined by $\alpha_i, \mathcal{D}(\alpha_i)$ as follows :

Let $\ell_1, \ldots, \ell_{q_i}$ be the sub-sequence of the integers $\{j_i+1, j_i+2, \ldots, j_i'-1\}$ such that $n_{\ell_k} = n_{j_i} + 1$.

If it is empty : $w_j = \alpha_j$

If it is not : $w_j = \pi(\alpha_j)$

where :

$\pi(\alpha_j) \triangleq (d_{j_i}, d_{\ell_1})\ M_6(d_{\ell_2}, d_{\ell_3})\ M_6 \ldots M_6(d_{\ell_{q_i}}, d_{j_i'})$.

1.3.5. - Example

This is the continuation of 1.3.2 :

$\alpha_1 = (id + \sqrt{(id + id)} + id * (id-id))$

$\pi(\alpha_1) = (id + \sqrt{M_6} + id * M_6) = w_1$

$\alpha_2 = (id + id)$

$\alpha_2 = w_2$.

SECTION 2

2.1. - DESCRIPTION LANGUAGE : COMMUNICATING ACTORS [McQ 79]

In our model, an actor A consists of some code and one initialization port P_0. The code of an actor is an internal sequential program that executes when an initialization message arrives on the port P_0. (We shall only consider cases where an actor receives only one initialization message during its whole life).

The code of an actor of type A specifies :

1) What initialization message to wait for on port P_0 :

λ_{P_0} <arg : u, cont : (actor, port)>

means that A waits for the arrival on port P_0 of a message containing an argument of type u and a continuation couple (actor, port), for initializing its execution. We shall omit the argument if this one is not necessary.

2) What operations to perform on the data.

3) What new actors to create :
say $A_1, \ldots A_n$ of type A, $B_1, \ldots B_m$ of type B...

4) What new ports to create on A :
$P_1, \ldots, P_k$.

5) What synchronization messages to wait for on the created ports :

λ_P <arg : u>

means that A waits for a message containing an argument of type u to arrive on port P for continuing its execution.

6) What messages to send :

Initialization messages

$$[(B_1,P_0) \leftarrow \langle y,(A,P_1)\rangle]$$

means that A sends the initialization message $\langle y,(A,P_1)\rangle$ to the port P_0 of B_1.

Continuation messages :

$$[(X,Q) \leftarrow \langle y\rangle]$$

means that A sends the continuation message $\langle y\rangle$ to the port Q of X when assuming that the continuation couple in the initialization message of A was (X,Q).

2.2. - THE INTERPRETER

2.2.1. - Preliminary remarks

We shall define 3 different types of communicating actors : Scanning actors, Memory actors and Interpretation actors. The input of the Scanning actor consist of a well formed arithmetic expression (ie, we shall not consider error recovery problems). This actor is in charge of scanning the input text, creating one memory actor for each of the identifiers encountered in the expression, calculating the parenthesis nesting function and creating one interpretation actor which it initializes by the argument $(w,\mathcal{D}(w))$ (1.2). We shall not detail the code of this Scanning actor (the only non-trivial element of this actor, i.e. the code for the determination of the nesting function is detailed in [BAC 82] section 3.2.). The Memory actor related to the identifier id_j represents a memory location accessible by each of the created interpretation actors. Its code is as follows (denoting as X one interpretation actor and as π one of its ports) :

ACTOR M_j

λ_{P_o} < cont (X,Π) > ;

$[(X,\Pi) \leftarrow id_j]$;

END ACTOR

2.2.2. - The initialization argument of an interpretation actor

Consider the set $\Gamma = \{\alpha_i(w)$ for w varying in L(G) and i varying from 1 to $n(w)\}$. For $\alpha_i(w)$ in Γ, define $\mathcal{D}(\alpha_i(w))$ as in 1.3.4. The argument of the initialization message of an interpretation actor is a couple $(\gamma, \mathcal{D}(\gamma))$ where $\gamma \in \Gamma$. For such a couple say :

$$\begin{cases} d_j\ \gamma_j\ d_{j+1} \cdots \gamma_{j'-1}\ d_{j'} & d_i \in D,\ \gamma_i \in (V_T - D)^* \\ \delta_j\ \delta_{j+1} \cdots \delta_{j'} & \delta_i \in D \times \mathbb{N} \times \mathbb{N} \end{cases}$$

we shall use the following notations : $\ell_1, \ldots, \ell_p$ where $\ell_k \in \mathbb{N}$ will be :

- ε (the empty sequence) if $j' = j+1$,
- the subsequence of the $\{j, j+1, \ldots, j'\}$ with the property $n_k = n_j + 1$ otherwise

we shall also rewrite γ as :

* γ if $j' = j+1$

$$* \; d_j \; \beta_{\ell_1} \; d_{\ell_1} \; \beta_{\ell_2} \; d_{\ell_2} \; \beta_{\ell_3} \; \ldots \; \beta_{\ell_{2k-1}} \; d_{\ell_{2k-1}} \; \beta_{\ell_{2k}} \; d_{2k} \; \beta_{\ell_{2k+1}} \; \ldots$$

$$d_{\ell_{2p-1}} \; \beta_{\ell_{2p}} \; d_{\ell_{2p}} \; \beta_{\ell_{2p+1}} \; d_j$$

where for $1 \leq k \leq p$, $\beta_{\ell_{2k}} \in V_T^*$ and β_{ℓ_1}, $\beta_{\ell_{2k+1}} \in (V_T - D)^*$ otherwise.

2.2.3 The interpretation actor

Roughly speaking this actor creates one interpretation actor for each subtree (in the decomposition of the derivation tree) but the first. Our actor is in charge of parsing the first subtree and evaluating it as soon as the relevant information is communicated by the other actors. These local parsing and semantic tasks are detailed in APPENDIX 1

ACTOR A

λ_{P_0} < arg : $(\gamma, \mathscr{D}(\gamma))$; Cont (X,π)> ;

IF j'> j + 1 THEN

FOR k ← 1 TO p

$\gamma_k \leftarrow d_{\ell_{2k-1}} \; \beta_{\ell_{2k}} \; d_{\ell_{2k+1}}$;

CREATE ACTOR A_k;

CREATE PORT P_k ;

[$(A_k, P_0) \Longleftarrow (\gamma_k, \mathscr{D}(\gamma_k))$];

END FOR

$\hat{\gamma} \leftarrow d_j\ \beta_{\ell_1}\ M_6\ \beta_{\ell_3}\ M_6 \ \ldots\ \beta_{\ell_{2-1}}\ M_6\ \beta_{\ell_{2+1}} \ \ldots$

$\beta_{\ell_{p-1}}\ M_6\ \beta_{\ell_{p+1}}\ d_{j'}$;

PARSE $\hat{\gamma}$ IN $\hat{G}$;

FOR k ← 1 TO p

EXECUTE $\hat{\gamma}$ as long as $M_{6,k}$ is not needed ;

$\lambda_{P'_k}$ <arg : $x_k \in \mathbb{R}$> ;

END FOR ;

ELSE PARSE γ IN $\hat{G}$;

EXECUTE γ ;

x ← RESULT ;

[(X,π) <= (x)] ;

END ACTOR

Assume the actor SCANNER sends to A the following initialization message :

$$[(w, \mathcal{D}(w))\ ;\ (\text{SCANNER},\ Q\ )]\quad ,\ w \in L(G)$$

Then, the properties given in 1.3 show that the real number sent back by A to the port Q of the "SCANNER" is the value of the arithmetic expression w.

2.2.4. - Example

Assume that the actor SCANNER sends to A the message :

$$\left\{\begin{array}{l} [(w, \mathcal{D}(w)),\ (\text{SCANNER},\ Q\ )] \\ \text{where :} \\ w = ((id_1 + \sqrt{}(id_2+id_3) + id_4 * (id_5-id_6)) \\ \quad * (\sqrt{}(id_7+id_8) + \sqrt{}(id_9+id_{10}) + \sqrt{}(id_{11}+id_{12}))) \end{array}\right.$$

The actor A creates two new actors, say A_1 and A_2, in charge of interpreting $(\gamma_1, \mathcal{D}(\gamma_1))$ and $(\gamma_2, \mathcal{D}(\gamma_2))$ respectively, with :

$$\left\{\begin{array}{l} \gamma_1 \triangleq (id_1 + \sqrt{}(id_2+id_3) + id_4 * (id_5-id_6) \\ \gamma_2 \triangleq (\sqrt{}(id_7+id_8) + \sqrt{}(id_9+id_{10}) + \sqrt{}(id_{11}+id_{12})) \end{array}\right.$$

Then A builds $\hat{\gamma} = (M_{6,1} * M_{6,2})$ and parses $\hat{\gamma}$ in $\hat{G}$ as indicated in the appendix. From this, the following semantic tasks are generated :

λ_{P_1} <arg : x_1> ;
λ_{P_2} <arg : x_2> ;
$z_1 \leftarrow x_1 * x_2$;
RESULT $\leftarrow z_1$;

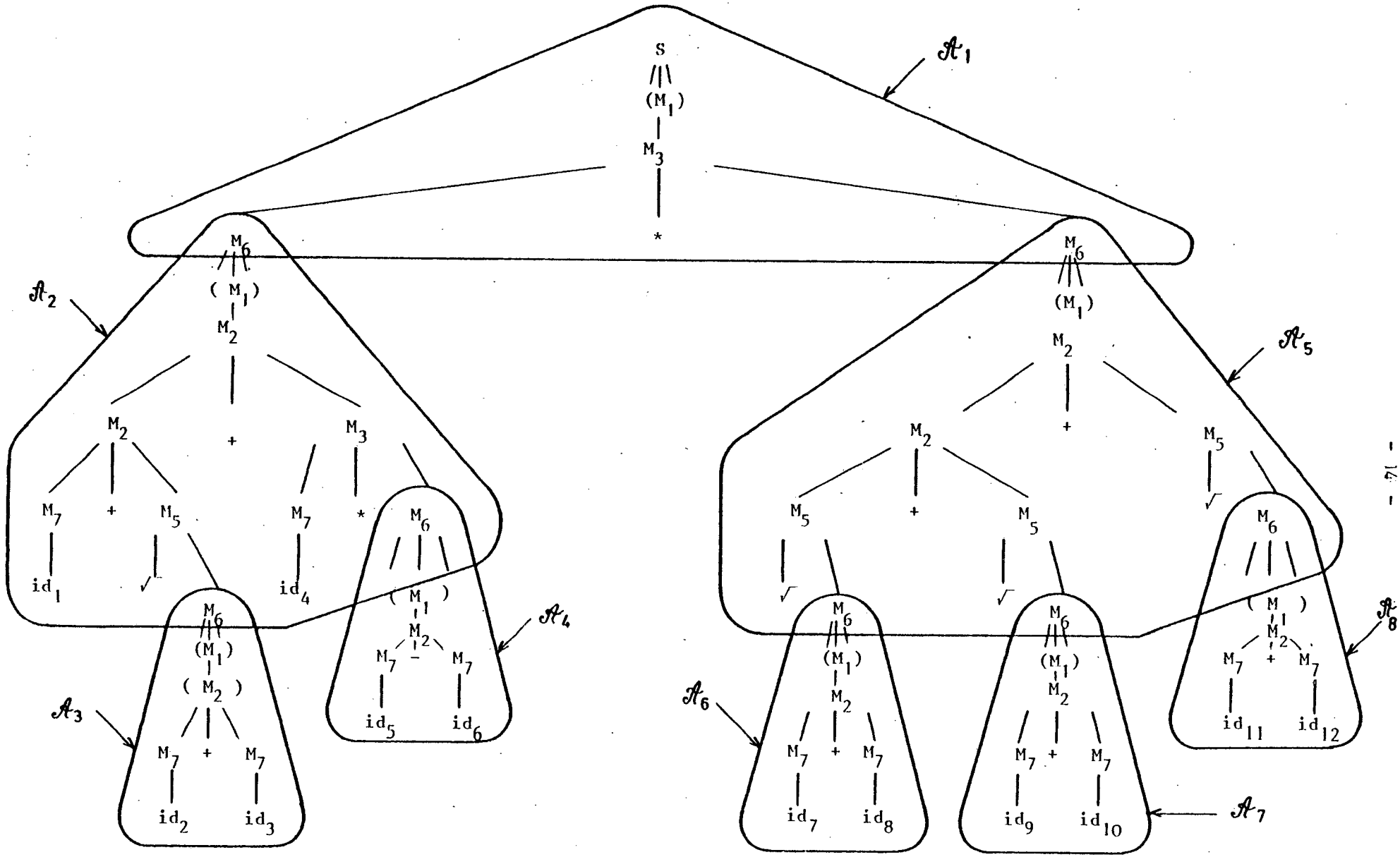
Once initialized by the message $<(\gamma_1, \mathcal{J}(\gamma_1)), (A,P_1)>$, A_1 creates two new actors say $A_{1,1}$, $A_{1,2}$ in charge of interpreting :

$$\gamma_{1,1} \triangleq (id_2 + id_3)$$
$$\gamma_{1,2} \triangleq (id_5 - id_6)$$

respectively. Then it builds $\hat{\gamma}_1 = (id_1 + \sqrt{M_6} + id_4 * M_6)$ parses $\hat{\gamma}_1$ in $\hat{G}$ and generates the following semantic tasks :

[$(M_1, p_o) \leftarrow <A, R_1>$] ;
λ_{R_1} < arg : id_1 > ;
λ_{P_1} < arg : x_1 > ;
z_2 SQRT (x_1) ;
$z_3 \leftarrow id_1 + z_2$;
[$(M_4, P_o) \leftarrow <A, R_4>$] ;
λ_{R_4} < arg : id_4 > ;
λ_{P_2} < arg : x_2 > ;
$z_4 \leftarrow id_4 * x_2$;
$z_3 \leftarrow z_1 + z_3$;
RESULT $\leftarrow z_3$;

We give in figure 2 the decomposition of the derivation tree of w and in figure 3 the creation and synchronization diagram of the actors created by the generator in order to interpret w.



We denote as w_i the sentence of $\mathcal{A}_i$ considered as a derivation tree in $\hat{G}$.

Figure 2

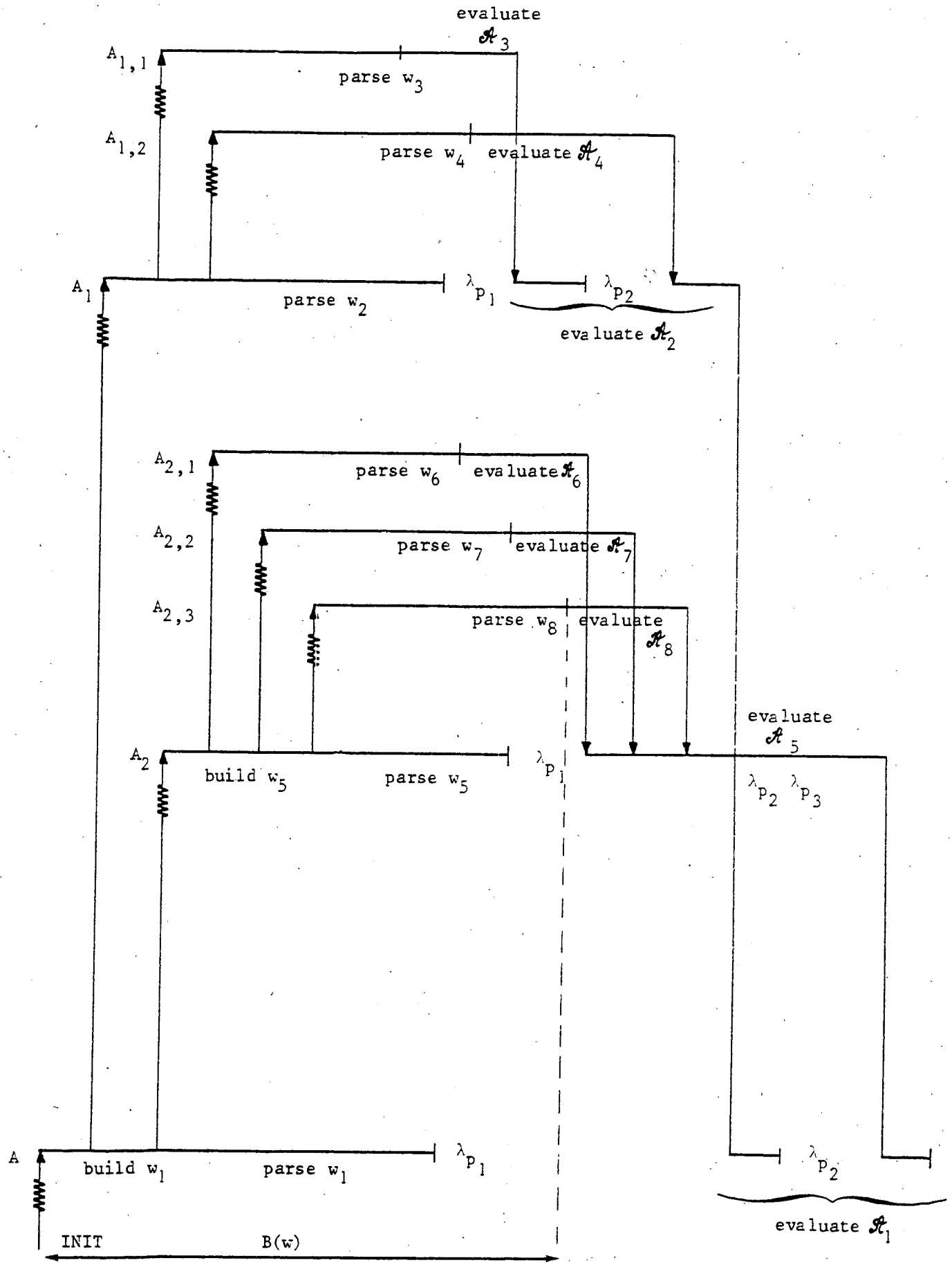

evaluate $\mathscr{A}_3$
$A_{1,1}$
parse w_3
$A_{1,2}$
parse w_4
evaluate $\mathscr{A}_4$
A_1
parse w_2
λ_{p_1}
λ_{p_2}
evaluate $\mathscr{A}_2$
$A_{2,1}$
parse w_6
evaluate $\mathscr{A}_6$
$A_{2,2}$
parse w_7
evaluate $\mathscr{A}_7$
$A_{2,3}$
parse w_8
evaluate $\mathscr{A}_8$
evaluate $\mathscr{A}_5$
A_2
build w_5
parse w_5
λ_{p_1}
λ_{p_2} λ_{p_3}
A
build w_1
parse w_1
λ_{p_1}
λ_{p_2}
evaluate $\mathscr{A}_1$
INIT
B(w)
I(w)

Figure 3

PART II

SECTION 1

1.1- INTRODUCTION

In order to get quantitative informations on the execution time of the interpreter defined in the previous section, we shall first specify the execution time of elementary parsing and semantic tasks to be processed by an interpretation actor in function of its input. From this we derive recursive equations for the interpretation time of an arithmetic expression. We are then faced with the following difficulty (which was already pointed out in [BAC 82] for parsing) : the speed up and efficiency very much depends upon the input sentence : a "balanced" sentence will allow a high speed up when for certain non-balanced cases, one could get a larger execution time than with the sequential interpreter. To overcome this difficulty, we propose to average this speed up over all possible sentences. The most natural probabilistic schema to be used in C.F. grammars is a stochastic grammar model [GON 78] . Roughly speaking, this consists of attributing weights (obtained by frequency measurements) to the multiple productions of each given nonterminal and of applying the rewrite rules, following these frequencies and independently of the context .

From this probabilistic model, we get the distribution function of the executing times of both parallel parsing and semantic phases as solutions of tractable fixed point equations. From this, we determine then average measures of speed up of both phases in comparison with their sequential version. We also derive an equation for the number of computing ressources which are needed. Lastly, we derive a fixed point equation for the distribution of an upperbound of the interpretation time which in turn provides lower bounds for the speed up and the efficiency of the parallel interpreter as a whole.

It is worth mentioning that the performance prediction of such asynchronous processes has received little attention in the modelling literature and that the analytic model presented in this second part (which contains as a particular case the performance evaluation of recursive Fork-Join structures) can be used within more general contexts.

1.2. - THE BRANCHING PROCESS MODEL

We shall sort the elements of V as follows : $S, M_1, \ldots, M_7, \pm, \mp, \overline{+}, /, **, \surd, (,), id$. Consider now the following real postive numbers, representing the probabilities of each of the possible productions related the non-terminals with multiple productions :

$M_1 : (p^1_j)$, $j = 2,7$, $\sum_j p^1_j = 1$

$M_2 : (p^2_{j,k,r})$ $\quad j = 2,7$, $\sum_{j,k,r} p^2_{j,k,r} = 1$. (We shall denote $P^2_{j,k,8} + P^2_{j,k,9}$ as $P^2_{j,k}$)

$k = 3,7$

$r = 8,9$ (+ and - respectively)

$M_3 : (p^3_{j,k,r})$ $\quad j = 3,7$ $\sum_{j,k,r} p^3_{j,k,r} = 1$. (We shall denote $P^3_{j,k,10} + P^3_{j,k,11}$ as $P^3_{j,k}$)

$k = 4,7$

$r = 10,11$ (* and / respectively)

$M_4 : (P^4_{j,k})$ $\quad j = 5,7$ $\quad k = 4,7$ $\quad \sum_{j,k} P^4_{j,k} = 1$

$M_5 : (P^5_j), j = 5,7$, $\sum P^5_j = 1$

The grammar G and the set of probabilities above define a multitype branching process [HAR 63] with $|V| = 17$ types. This process is completely caracterized by the following generating functions determined from the production rules of the 8 non-terminals(Denoting as $s_{j-1}, |s_{j-1}| \leq 1$, the complex variable related to the j-th element of V) :

$$\begin{cases} f_o(\vec{s}) = s_1 s_{14} s_{15} \\ f_1(\vec{s}) = \sum_{j=2,7} p^1_j s_j \\ f_2(\vec{s}) = \sum_{j=3,7} \sum_{k=3,7} \sum_{r=9,10} p^2_{j,k,r} s_j s_k s_r \\ \vdots \\ f_{16}(\vec{s}) = 1 \end{cases}$$

Let $(n_0, \ldots, n_{|V|-1})$ be the total number of types $S, M_1, \ldots, M_7, \pm, \mp, \ldots, \underline{id}$ respectively, in a sample path of our branching process. When the Frobenius condition is satisfied, the functional equation :

$$(1) \qquad W^i(\vec{s}) = s_i \, f_i(W^0(\vec{s}), \ldots, W^{16}(\vec{s})) \quad , \; i = 0,16$$

has a unique solution $(W^0(\vec{s}), \ldots W^{16}(\vec{s}))$ where the $W^i(s)$'s are generating functions of non defective random variables on $(\mathbb{N})^{17}$. Furthermore, the generating function $E[s_0^{n_0} \; s_1^{n_1} \ldots s_{16}^{n_{16}}]$ is given by $W^0(s)$. (see [HAR 63]).

1.3. DISTRIBUTION FUNCTION OF THE NUMBER OF CREATED ACTORS

The total number of interpretation actors created for interpreting a sample arithmetic expression (ie the sentence produced by a sample path of our branching process) is $N = n_6 + 1$, so that the generating function of $N, y_6(z)$, is equal to $zw_6(z)$ when denoting as $w_i(z)$ the function $W^i(1,1,1,1,1,Z,1,\ldots,1)$, $|z| \leq 1$. For determining $y_6(z)$, one can reduce (1) to the following functional equation with unknowns $(y_2(z), \ldots, y_6(z))$:

$$(2) \qquad \begin{cases} y_6(z) = z(P_7^1 + \sum_{j=2}^{6} P_j^1(z)) \\ y_5(z) = P_7^5 + \sum_{j=5}^{6} P_{j,7}^4 \; y_j(z) + \sum_{k=4}^{6} P_{7,k}^4 \; y_k(z) + \sum_{j=5}^{6} \sum_{k=4}^{6} P_{j,k}^i \; y_j(z).y_k(z) \\ y_i(z) = P_{7,7}^i + \sum_{j=i}^{6} P_{j,7}^i \; y_j(z) + \sum_{k=i+1}^{6} P_{7,k}^i \; y_k(z) + \sum_{j=1}^{6} \sum_{k=i+1}^{6} P_{j,k}^i \; y_j(z).\, y_k(z) \quad i=2,4 \end{cases}$$

The successive eliminations of $y_5(z)$, $y_4(z)\ldots$, $y_2(z)$ in (2) yield for $y_6(z)$ a polynomial functional equation of degree 4.

EXAMPLE

Assume the derivations in non-terminals are equiprobable in the following sense : $\rho \in [0,1]$ and

$$(3)\quad \begin{cases} p_7^1 = (1-\rho)\ ,\ p_j^1 = \rho/5\ ,\ j = 2,6 \\ p_7^5 = (1-\rho)\ ,\ p_j^5 = \rho/2\ ,\ j = 5,6 \\ p_{jk}^4 = q_j^4\, r_k^4\ ,\ q_7^4 = r_7^4 = 1-\rho\ ,\ q_j^4 = \rho/2\ ,\ j = 5,6 \\ \qquad\qquad r_k^4 = \rho/3\ ,\ j = 4,6 \\ p_{jk}^i = q_j^i\, r_k^i\ ,\ q_7^i = r_7^i = 1-\rho\ ,\ q_j^i = \rho/(7-i),\ j = i,6 \\ \qquad\qquad r_k^i = \rho/(6-i),\ j = i+1,\ 6 \end{cases}$$

$\rho \in [0,1]$ is a "tuning" parameter (we shall see later on that there exists a bijective mapping $\rho \leftrightarrow E[L]$ where $E[L]$ is the average length of the arithmetic expression so that scaling with respect to ρ is equivalent to scaling with respect to this average length). In this case, we get by elimination :

$$(4)\quad y_6(z)^4\, A_4(\rho) + y_o(z)^3\, (A_3(\rho) + zB_3(\rho)) + y_6(z)^2\, ((A_2(\rho) + zB(\rho)) + y_6(z)\ (A_1(\rho) + zB_1(\rho)) + z\, B_o(z) = 0$$

where the A_i's (B_i's) are known polynomials in the variable ρ given in Appendix 2. Differenciating equation(4) , one gets the following expression :

$$E[N] = \frac{\prod_{i=2}^{5} (\rho - i)}{120 - 178\rho + 45\rho^2 - 23\rho^3}$$

The polynomial $P(\rho) \triangleq 120 - 178\rho + 45\rho^2 - 23\rho^3$ has two complex roots and one real positive root $\rho_{max} = 0.7641\ldots$ which is the maximum value of ρ for which the Frobenius condition is satisfied (Appendix 3).

1.4. EXECUTION TIME OF THE PARSING PHASE

1.4.1 Assumptions and notations

We shall assume communication times between actors to be zero. For $\hat{\gamma} \in L(\hat{G})$, we shall denote as $a(\hat{\gamma})$ the time required for parsing $\hat{\gamma}$ in $\hat{G}$ (It is shown in [BAC 82] that this time is equal to $b + cn_{\theta}(\hat{\gamma})$ where b and c are positive integers and $n_{\theta}(\hat{\gamma})$ is the total number of operators $(+,-,\ldots,\sqrt{\ })$ in $\hat{\gamma}$). Let $\bar{\gamma} \in \Gamma$ (see 2.2.2 for this notation). We assume that once initialized, an actor requires $n_6(\hat{\gamma})d$ units of time to create its $n_6(\hat{\gamma})$ son actors, where d is a positive integer (one actor is created each d units of time from initialization). Lastly, we shall denote as $B(\gamma)$ the parsing time of γ, that is the time between the initialization of an interpretation actor by the message $(\gamma, \mathcal{D}(\gamma))$ and the end of the latest parsing task among all the actors created for this input (see fig. 3).

1.4.2 A fixed point equation for the distribution departing time

Using the notations of 2.2.3., we get the following recursion :

$$(5) \qquad B(\gamma) = n_6(\hat{\gamma})d + \max\left(a(\hat{\gamma}), \max_{j=1,n_6(\hat{\gamma})} (B(\gamma_j) - d(n_6(\hat{\gamma})-j))\right)$$

so that for $w \in L(G)$, $B(w)$ is determined from the couples $(n_6(w_i))$ $i=1, n(w)$ (see 1.3.3 for these notations). Using classical independence properties in branching processes, we get from (5) the following fixed point equation for $\pi(n) \stackrel{def}{=} \text{Prob}\,[B(w) \leq n,\ w \in L(G)]$:

$$(6) \qquad \Pi(n) = \sum_{k \leq M} \alpha(n,k) \prod_{i=1}^{k} \Pi(n - d.i)$$

where $\alpha(n,k)$ denotes $\text{Prob}\,[b+cn_{\theta}(\hat{\gamma}) \leq n - dk,\ n_6(\hat{\gamma}) = k, \gamma \in L(\hat{G})]$
Let $n_6(w)$ ($n_{\theta}(w)$) denote the total number of M_6 (operators) in the derivation tree of w. From (5), we get the upperbound :

$$B(w) \leq dn_6(w) + b(n_6(w) + 1) + cn_{\hat{e}}(w)$$

proving hence that π is the law of a proper random variable whenever the Frobenius condition is satisfied by the branching process related to G.

1.4.3 Calculation of the function $\alpha(n,k)$

Let $K(x,y)$, $|x| \leq 1$, $|y| \leq 1$ be the joint generating function $E[x^{n_\theta(\hat{\gamma})} y^{n_6(\hat{\gamma})}, \hat{\gamma} \in L(\hat{G})]$ in the multitype branching process defined by $\hat{G}$ and the probabilities (3.2). The generating function

$$\begin{cases} A(x,y) \triangleq \sum_{n\geq 0} \sum_{k\geq 0} \alpha(n,k)\, x^n y^k \\ |x| < 1, \quad |y| \leq 1 \end{cases}$$

is related to $K(x,y)$ by :

$$A(x,y) = \frac{x^b}{1-x} K(x^c, x^d y)$$

Thus, it is sufficient to determine $K(x,y)$ to get the $\alpha(n,k)$'s. From (1) rewritten in $\hat{G}$, we obtain :

$$(7) \qquad K(x,y) = \sum_{i=1}^{7} p_i^1 g_i(x,y)$$

$$(8) \begin{cases} g_6(x,y) = y \\ g_5(x,y) = (x \sum_{j=6}^{7} p_j^5 g_j(x,y)) / (1 - x\, p_5^5) \\ g_4(x,y) = (x \sum_{j=5}^{7} \sum_{k=5}^{7} p_{j,k}^4 g_j(x,y) g_k(x,y) / (1 - x(\sum_{j=5}^{7} p_{j,4}^4 g_j(x,y)) \\ g_i(x,y) = (x \sum_{j=i+1}^{7} \sum_{k=i+1}^{7} p_{j,k}^i g_j(x,y) g_k(x,y)) / (1 - x (\sum_{j=i+1}^{7} p_{i,j}^i g_j(x,y))) \end{cases}$$

for $i = 3,2$

1.4.4 Example (continuation of 1.3)

From (8) we get :

$$K(x,y) = \frac{B(x,\rho) + y\,A(x,\rho)}{D(x,\rho) - y\,C(x,\rho)}$$

where

$$A(x,\rho) = 24\rho - x\,(10\rho^2(1-\rho))$$

$$B(x,\rho) = 120(1-\rho) - x.2\rho\,(1-\rho)\,(29-11\rho) + x^2\rho^2(1-\rho)^2$$

$$C(x,\rho) = x\,\rho^2(36 - x\rho\,(1-\rho))$$

$$D(x,\rho) = 120 - x\rho\,(154 - 94\rho) + x^2\rho^2(\rho-1)\,(2\rho-13)$$

Since $D(x,\rho) > C(x,\rho)$ for $\rho \in [0,1]$, $x \in [0,1]$, we get the following expression for $\alpha(n,k)$:

$$\alpha(n,k) = \begin{cases} [x^n]\ \left(\frac{x^b}{1-x} \cdot \frac{B(x^c)}{D(x^c)}\right) & \text{if } k = 0 \\ [x^n]\ \left(\frac{x^{b+dk}\,C^{n-1}(x^c)}{(1-x)D^{n+1}(x^c)}\right)\,(B(x^c)C(x^c) + A(x^c)D(x^c)) & \text{otherwise} \end{cases}$$

where $[x^n](\beta(x))$ denotes the coefficient of x^n in the expansion of $\beta(x)$.

1.5 EXECUTION TIME OF THE SEMANTIC PHASE

1.5.1 Assumptions and notations

We shall assume that both the execution of an elementary arithmetic operation and the emission of a message require one unit of time. Hence the number of time units for an actors to get the value of an identifier is 2.

The aim of the present section is to analyze the semantic phase alone. For this, we shall assume that the semantic phases of all the created actors begin simultaneously (at time B(w)). Under this assumption, for $\gamma \in \Gamma$ and $p \in \mathbb{N}$, let $S_p(\gamma)$ be the number of time units needed for completing all the semantic work related to γ assuming that this work is delayed for p-units of time. (see fig.4).

1.5.2 Recursive equations for the interpretation time

For $\gamma \in \Gamma$, let $\hat{\gamma}$ be defined as in (2.2.3) and $\hat{\mathcal{A}}$ be the reduced derivation tree of $\hat{\gamma}$ in $\hat{G}$ (see appendix I). Let ρ be a node of $\hat{\mathcal{A}}$ and n be a positive integer. $\phi(n,\rho) \in \mathbb{N}$ will denote the number of time units needed for the interpretation actor to complete all the semantic tasks related to the subtree rooted by ρ (ie to execute the arithmetic operation related to node ρ if ρ is an operator or to dispose of ρ's value if $\rho \in \{id, M_6\}$, when assuming that the interpretation actor has completed all the semantic tasks to be executed before those in the subtree rooted by ρ in (n-1) units of time. From the sorting of semantic tasks described in appendix 1, we get the following recursion for ϕ where r and ℓ denote the right and left sons of ρ in $\hat{\mathcal{A}}$ (if any) :

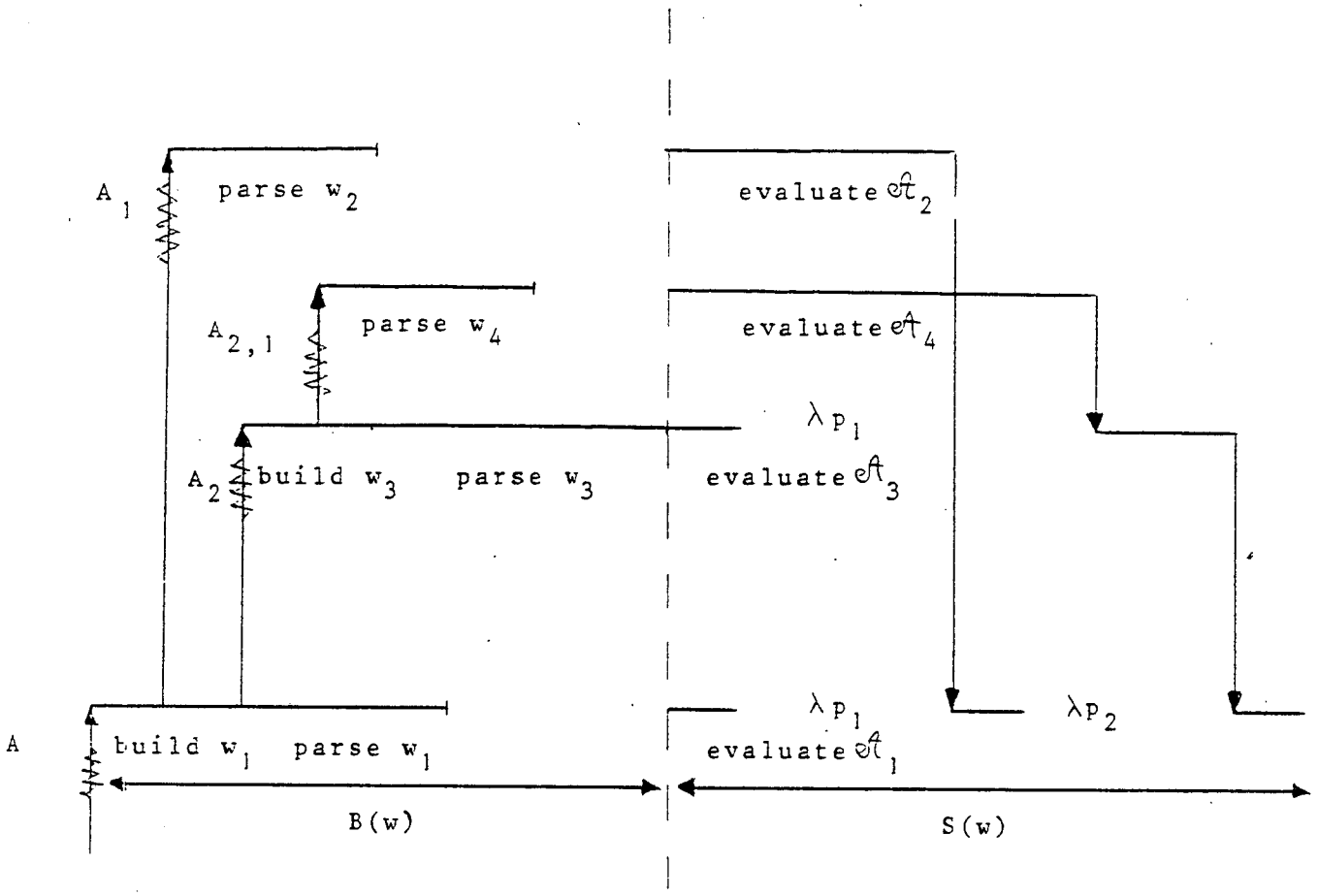
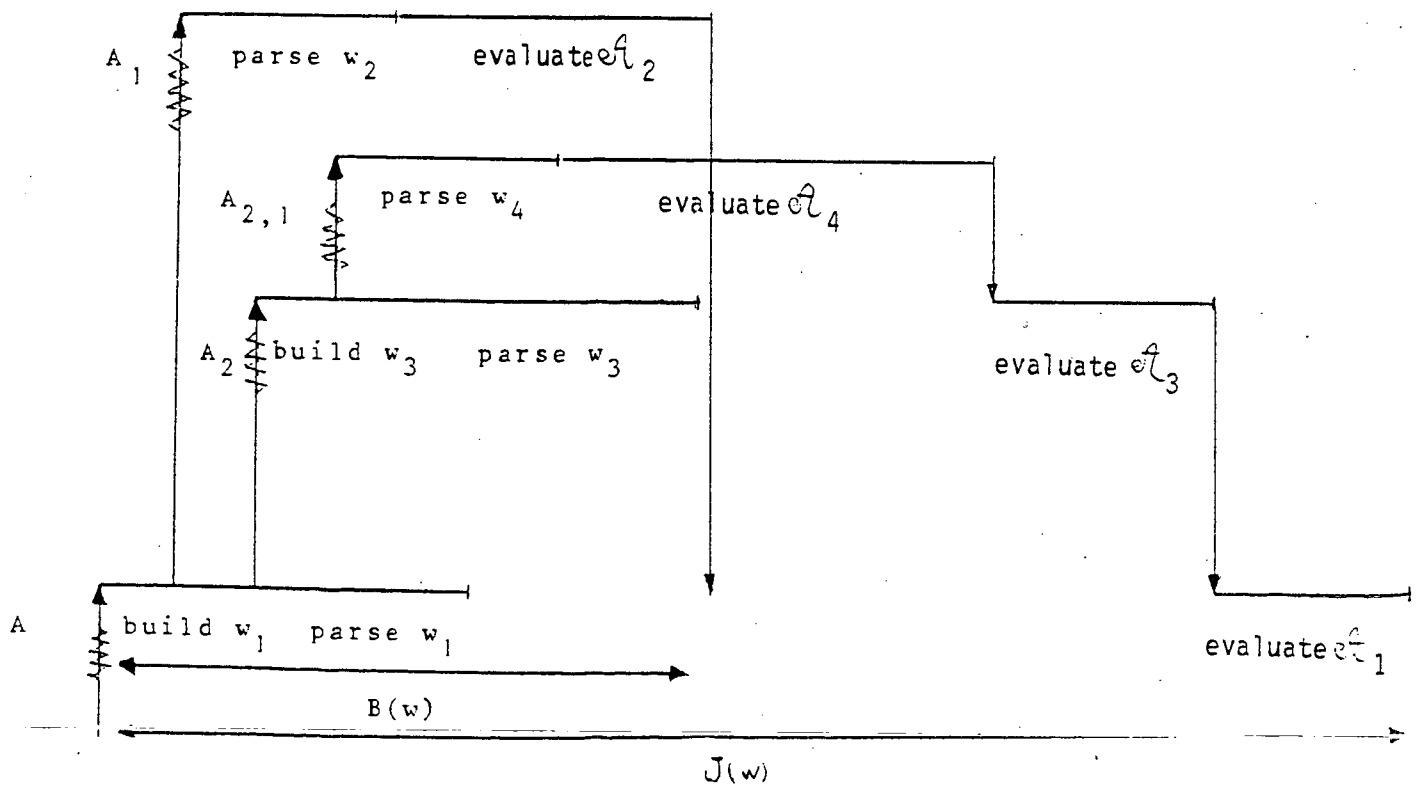

A 1
parse w 2
evaluate 𝒜 2
A 2,1
parse w 4
evaluate 𝒜 4
λ p 1
A 2
build w 3
parse w 3
evaluate 𝒜 3
λ p 1
λ p 2
A
build w 1
parse w 1
evaluate 𝒜 1
B(w)
S(w)

Fig 4

Fig 5


A 1
parse w 2
evaluate 𝒜 2
A 2,1
parse w 4
evaluate 𝒜 4
A 2
build w 3
parse w 3
evaluate 𝒜 3
A
build w 1
parse w 1
evaluate 𝒜 1
B(w)
J(w)

$$(11)\quad \begin{cases} \psi(n,\rho) = \begin{cases} 2 & \text{if } \rho = \mathrm{id} \\ 1 + \psi(n,r) & \text{if } \rho = \surd \\ 1 + \psi(n,\ell) + \psi(n+\psi(n,\ell),r) & \text{if } \rho \in \{+,-,*,/,**\} \\ (S(\gamma_j) - n) \vee 0 & \text{if } \rho = M_{6,j}. \end{cases}^{+} \\ S(\gamma) = 1 + \psi(1, \mathrm{root}\ (\hat{\gamma})). \end{cases}$$

Using now in (11) the independence properties in the branching process model, we get the fixed point equation (12) hereafter for $\nu(k) = \mathrm{Prob}\ [S(w) = k,\ w \in L(G)]$, $k \in \mathbb{N}$:

$$\nu(k+1) = \sum_{j=2}^{7} p_j^1\ q_j(0,k)\ ,\ \nu(0) = 0$$

where $q_j(n,m)$ denotes Prob $[S(\gamma) = m \mid \mathrm{root}\ (\hat{\alpha})$ is of type j and the evaluation of $\hat{\gamma}$ begins at $B(w) + n+1\ ]$. We get the following recursion : $\forall n,m \in \mathbb{N}$

$$(12)\quad \begin{cases} q_7(n,m) = 1_{(m=2)}\ ; \\ q_6(n,m) = 1_{(m>0)}\nu(n+m) + 1_{(m=0)}\sum_{s=0}^{n} \nu(s)\ ; \\ q_5(n,m) = 1_{(m>0)}\left[\sum_{j=5,7} p_j^5\ q_j(n,m-1)\right]\ ; \\ q_4(n,m) = 1_{(m>0)}\left[\sum_{j=5,7}\ \sum_{k=4,7}\ \sum_{s=0,m-1} p_{j,k}^4\ q_j(n,s)\ q_k(n+s,m-s-1)\right] \\ q_i(n,m) = 1_{(m>0)}\left[\sum_{j=i,7}\ \sum_{k=i+1,7}\ \sum_{s=0,m-1} p_{j,k}^i\ q_j(n,s)\ q_k(n+s,m-s-1)\right] : \\ \quad i=2,3 \end{cases}$$

Notice that when the Frobenius condition is satisfied by G, ν is the distribution function of a proper random variable on $\mathbb{N}$. This is proven by the following inequality (obtained from (11) and $(S(\gamma_i)-n)\vee 0 \le S(\gamma_i)$) :

$$S(w) \le 1 + n_6(w) + 2.n_7(w) + n_\theta(w)$$

$^{+}$For $a \in \mathbb{Z}$, $b \in \mathbb{Z}$, $a \vee b$ denotes $\max(a,b)$.

1.5.4 Example (continuation)

We get the following five first probabilities :

$$\nu(0) = \nu(1) = \nu(2) = 0$$

$$\nu(3) = 1 - \rho$$

$$\nu(4) = 2\ (1-\rho)\ \rho/5$$

$$\nu(5) = (1-\rho)\ \rho^2(36\rho^2+19\rho-139)/300$$

...

1.5.5 A fixed point equation for the distribution of an upper bound of the interpretation time

In this section, we consider that for $\gamma \in \Gamma$, the semantic work related to each actor begins only when both the parsing of $\hat{\gamma}$ is completed and the data to be provided by its son actors are all available.

This modification (exemplified in fig 5) clearly provides an overestimate for the interpretation time $I(\gamma)$ (see fig.3).
Let $J(\gamma)$ be this overestimate. One gets :

$$(13) \quad J(\gamma) = \left[\max_{i=1,n_6(\hat{\gamma})} (d_i + J(\hat{\gamma}_i)) \ v\ (dn_6(\hat{\gamma}) + a(\hat{\gamma})) \right] + g(\hat{\gamma})$$

Where $g(\hat{\gamma}) = n_8(\hat{\gamma}) + 2n_7(\hat{\gamma}) + 1$ denotes the number of time units needed for executing the semantic of $\hat{\gamma}$ in the case considered in this section.

Let $\eta(n)$ $n \in \mathbb{N}$, denote Prob $[J(w) \leq n,\ w \in L(G)]$. We get from (13):

$$(14) \quad \eta(n) = \sum_{k\geq 0,\ell\geq 0} \ \sum_{j=0}^{n-b-c\ell-d-k} \beta(\ell,k,j) \cdot \prod_{i=1}^{k} \eta(n-j-d_i)$$

where $\sum_{j=0}^{m}$ is zero if $m < 0$ and

Where $\beta(\ell,k,j) = \text{Prob}\,[n_6(\hat{\gamma}) = k,\ n_\theta(\hat{\gamma}) = \ell, 2n_7(\hat{\gamma}) = j - (1+\ell),$ $\hat{\gamma} \in L(\hat{G})\,]$. The proof of the finiteness of $J(w)$ (under the Frobenius condition for G) is similar to those of 1.4.2 and 1.5.3. Notice furthermore that

$$B(x,y,z) = \sum_{k\geq 0,\ell\geq 0,j\geq 0} \beta(\ell,k,j)x^{\ell}y^{k}z^{j}$$

is related to $K(x,y,z) \triangleq E\,[x^{n_\theta(\hat{\gamma})},\ y^{n_6(\hat{\gamma})},z^{n_7(\hat{\gamma})},\hat{\gamma} \in L(\hat{G})\,]$ by :

$$B(x,y,z) = z\,K(xz,y,z^2)$$

and that $K(x,y,z)$ is obtained exactly as in (7) and (8) in which the first line of (8) is replaced by $g_7(x,y,z) = z$.

1.5.6 Example (continuation)

$$(15)\qquad K(x,y,z) = \frac{z^2N_2(x,y) + z\,N_1(x,y) + N_o(x,y)}{z^2D_2(x,y) + z\,D_1(x,y) + D_o(x,y)}$$

Where

$$D_2(x,y) = x^2\,[2\rho^2(1-\rho)^2]$$

$$D_1(x,y) = x^2\,[\rho^2(1-\rho)\ (y\rho+11)\,] - x\,[94\rho(1-\rho)]$$

$$D_0(x,y) = x\,[-12\rho\ (3y\rho+5)] + 120$$

$$N_2(x,y) = x^2[\,\rho^2(1-\rho)^2] - x[22\rho\ (1-\rho)^2\,]$$

$$N_1(x,y) = n\,[-\rho(1-\rho)\ (10\rho y+36)] + 120(1-\rho)$$

$$N_0(x,y) = 24\rho y.$$

1.6 SPEED UP AND EFFICIENCY

From the results in [BAC 82] and those of the previous sections, we get integess α,β,γ and δ such that the sequential interpretation (based on the same precedence function method) time of $w \in L(G)$ is given by :

(16) $\Sigma(w) = \alpha + \beta\, n_6(w) + \gamma\, n_7(w) + \delta\, n_\theta(w)$

Hence $E[\Sigma(w), w \in L(G)]$ is obtained from $E[n_6(w), w \in L(G)]$, $E[n_7(w), w \in L(G)]$ and $E[n_\theta(w), w \in L(G)]$ which are in turn derived from (1) using standard techniques. In our example for instance :

$$E[\Sigma(w)] = \alpha + \frac{\beta \prod_{i=2}^{5}(\rho - i) + \gamma(1-\rho)(\rho^3+21\rho^2+14\rho+120) + \delta(\rho^2-\rho+8)12\rho}{120-178\rho+45\rho^2-23\rho^3}$$

Similarly, denoting as L(w) the number of symbols in w (its length), we get

$$L(w) = 2 + 2M_6 + M_7 + M_\theta$$

so that

$$E[L] = 2 + \frac{2 \cdot \prod_{i=2}^{5}(\rho - i) + (1-\rho)(\rho^3+21\rho^2+14\rho+120) + (\rho^2-\rho+8)12\rho}{120 - 178\rho + 45\rho^2 - 23\rho^3}$$

Proving hence that $E[L]$ is an increasing function of ρ for $\rho < \rho_{max}$. (see figure 6).

We define now the speed up of the parallel interpreter as

(17) $$s \triangleq \frac{E[\Sigma(w), w \in L(G)]}{E[I(w), w \in L(G)]}$$

and its efficiency as

(18) $$e \triangleq \frac{s}{E[N(w), w \in L(G)]}$$

Let $\bar{\pi}, \bar{\nu}, \bar{\eta}$ be the respective first moments (which we know to exist under the Frobenius condition) of the distributions π, ν and η (given by equations (6), (12) and (14)) . Since $J(w) \leq I(w)$ and $J(w) \leq B(w) + S(w)$, we get the lowerbound :

$$s \geq \frac{E[\Sigma(w), w \in L(G)]}{\bar{\eta} \wedge (\bar{\pi}+\bar{\nu})}$$

Providing also a lowerbound for e.

Some of the computations of the present section have been made using the macsyma programm for symbolic calculation.

APPENDIX I

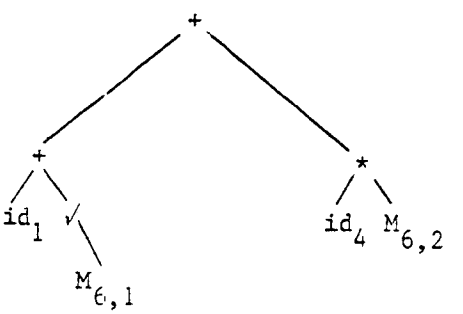
Parsing

Consider for $\hat{\gamma} \in L(\hat{G})$, the reduced derivation tree of $\hat{\gamma}$, build as follows from $\hat{\mathcal{A}}$, the derivation tree of $\hat{\gamma}$ in $\hat{G}$:

-Replace in $\hat{\mathcal{A}}$ the subtree " $\hat{S}$ (M_j) " by " M_j ".

-For each M_j, j=2,5 in $\hat{\mathcal{A}}$, delete the son being an operator (ie $\in \sigma \triangleq \{+,-,*,/,**,\surd\}$) and replace M_j by this operator.

-For each M_7 in $\hat{\mathcal{A}}$, delete its son and replace M_7 by id. For instance, the direct derivation tree of $\hat{\gamma}_1 = (id_1 + \surd M_{6,1} + id_4 * M_{6,2})$ in example 2.3.4 is :



For $\theta \in \sigma$, let assoc (θ) be + 1 if $\theta \in \{**,\surd\}$, -1 otherwise and prio (θ) be 1 if $\theta \in \{+,-\}$, 2 if $\theta \in \{*,/\}$, 3 if $\theta = **$ and 4 if $\theta = \surd$. If $(\theta_1,\ldots,\theta_q)$ is the sequence of the operation in γ(in our example, q=4, θ_1=+,θ_2=$\surd$,θ_3=+,θ_4=*) let prec (θ_i) $\triangleq$ i * assoc (θ_i) + 2*(1+q)* prio (θ_i), for i=1,q. This function allows to calculate the reduced derivation tree of $\hat{\gamma}$ as follows (see [Fis 80] for a proof) :

For $1 \leq i \leq q$ let SR(i) $\triangleq$ $\{\theta_j, i<j\leq q$ such that $\nexists$ k $i<k<j$ satisfying prec (θ_k) < prec (θ_i)$\}$ (the largest set of consecutive operators on the right of θ_i with a precedence greater than θ_i's).

-If SR(i) = $\emptyset$, the right son of θ_i is the first terminal on its right in $\hat{\gamma}$.

-If SR(i) $\neq \emptyset$ the right son of θ_i is the θ_j in SR(i) with the smallest precedence, say $\theta_{R(i)}$.

A symmetrical relation exists between a set SL(i) and the left son of θ_i. In our example prec (θ_i) = 9, prec (θ_2) = 42, prec (θ_3) = 7, prec (θ_4) = 16 which yields the direct derivation tree here above.

2 Semantic

The evaluation task related to node θ_j of the reduced derivation tree is generated as follows :

IF LEFTSON (RIGHTSON) θ_i is id_j (id_k) THEN [$(M_j, P_0) \longleftarrow$ <A,R_j>] ([$(M_R, P_0) \longleftarrow$ <A,R_k>]) ; λ_{R_i} < arg : id_j> (λ_{R_k} <arg : id_k>);

ELSEIF LEFTSON (RIGHTSON) θ_i is $M_{6,j}(M_6, k)$ THEN λ_{P_j} < arg : $x_j \in \mathbb{R}$ > (λ_{P_k} <arg : $x_k \in \mathbb{R}$ >) ; ℓ_i denotes x_j(r_i denotes x_k).

ELSE LEFTSON (RIGHTSON) θ_i is $\theta_j(\theta_k)$ THEN ℓ_i denotes z_i (r_i denotes z_k).

ENDIF ;

$z_i = \ell_i \; \theta_i \; r_i$;

In order to sort these tasks, we shall use the following order, departing from the root : evaluate θ_i's left subtree (if any) ; evaluate θ_i's right subtree (if any) ; evaluate note θ_i. This order can be determined during the parsing phase from the calculation of SR and SL : The rank of the operator with minimal precedence function will be zero and, if the rank of θ_i is known

-The rank of the operator $\theta_{R(i)}$ if any will be θ_i's rank plus one.

-The rank of the operator $\theta_{L(i)}$ if any will be θ_i's rank plus |SR(i)| plus one.

The proposed sorting corresponds to a decreasing rank order.

APPENDIX 2

Let $Q(\rho) = \rho^3-15\rho^2+26\rho-24$. We have :

$A_4(\rho) = 36\rho^6(\rho^2-\rho+36)$

$A_3(\rho) = 12\rho^4(\rho^5-22\rho^4+122\rho^3-911\rho^2+1422\rho-1224)$

$B_3(\rho) = 72\rho^5(5\rho^2-5\rho+12)$

$A_2(\rho) = \rho^2\ Q(\rho)\ \ (\rho^5-40\rho^4+257\rho^3-1874\rho^2+2808\rho-2304)$

$B_2(\rho) = 12\rho^3(13\rho^5-232\rho^4+677\rho^3-1394\rho^2+1224\rho-576)$

$A_1(\rho) = -Q(\rho)^2.\ (2\rho^4-15\rho^3+107\rho^2-154\rho+120)$

$B_1(\rho) = 2\rho\ Q(\rho)\ (11\rho^5-224\rho^4+703\rho^3-1498\rho^2+1152\rho-288)$

$B_0(\rho) = Q(\rho)^2\ (\rho^3-23\rho^2+58\rho-120)\ (\rho-1)$

APPENDIX 3

In the example 3.2.2, the Frobenius condition is satisfied if and only if $\rho \in (0,\rho_{max})$.

To check this, it is enough to verify that the greatest positive characteristic root of the matrix $m_{ij} = \partial f_i/\partial s_j(1,\ldots,1)$ $i = 6$, $j = 1,6$ is less than one, or equivalently that the characteristic polynomial

$$C(x) = x(x-\frac{\rho}{5})\ (x-\frac{\rho}{4})\ (x-\frac{\rho}{3})\ (x-\frac{\rho}{2}) - \frac{\rho}{5}\ (x+\frac{\rho}{4})\ (x+\frac{\rho}{3})\ (x+\frac{\rho}{2})$$

has no positive root greater than 1 whenever $\rho<\rho_{max}$. One checks numerically that for $\rho=\rho_{max}^-$ this condition is satisfied whence it is not satisfied for $\rho=\rho_{max}^+$. To complete the proof, remark that each of the $m_{ij}(\rho)$ is an increasing function of ρ.

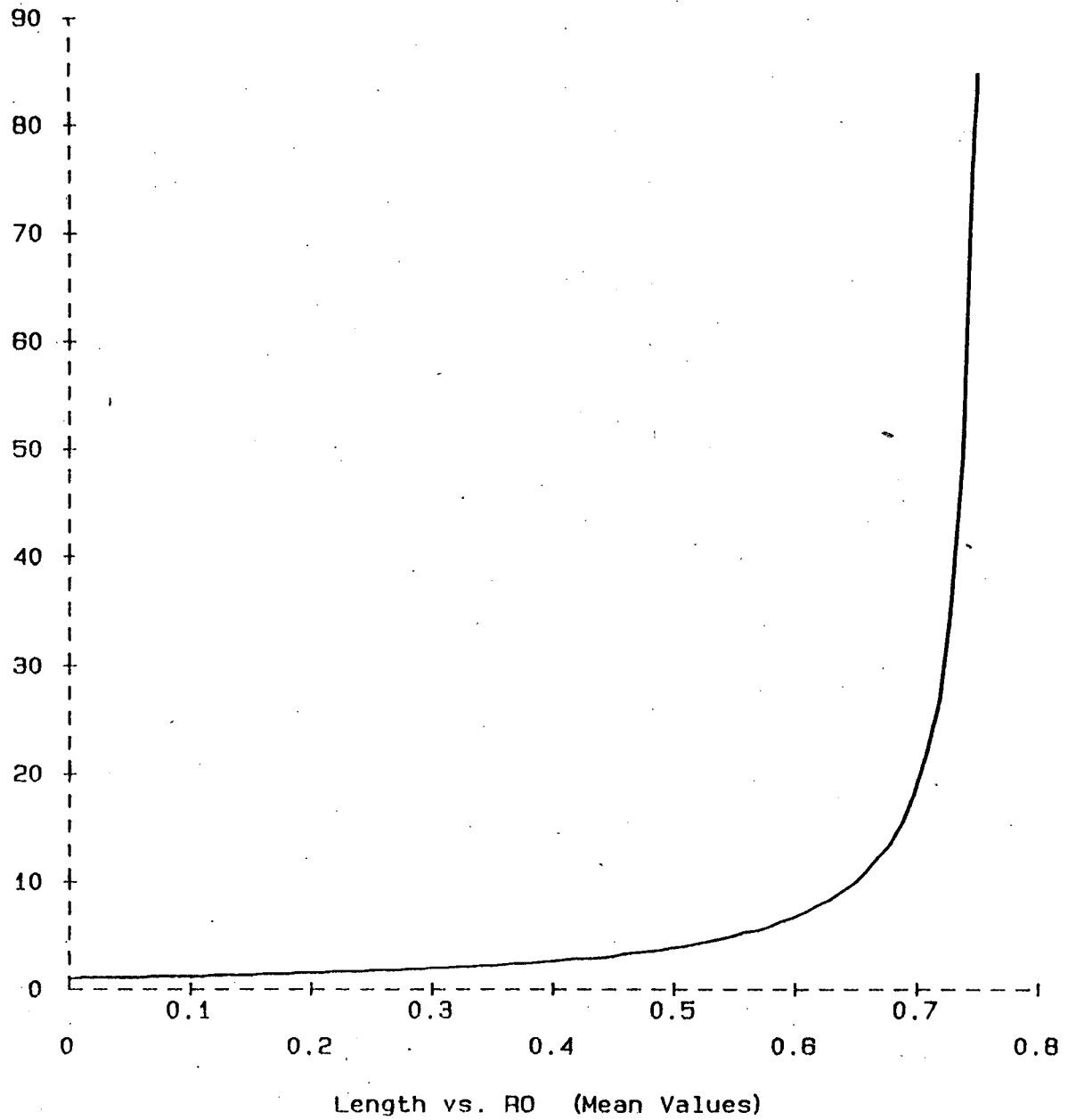

90
80
70
60
50
40
30
20
10
0
0.1
0.3
0.5
0.7
0
0.2
0.4
0.6
0.8
Length vs. RO (Mean Values)

Fig. 6

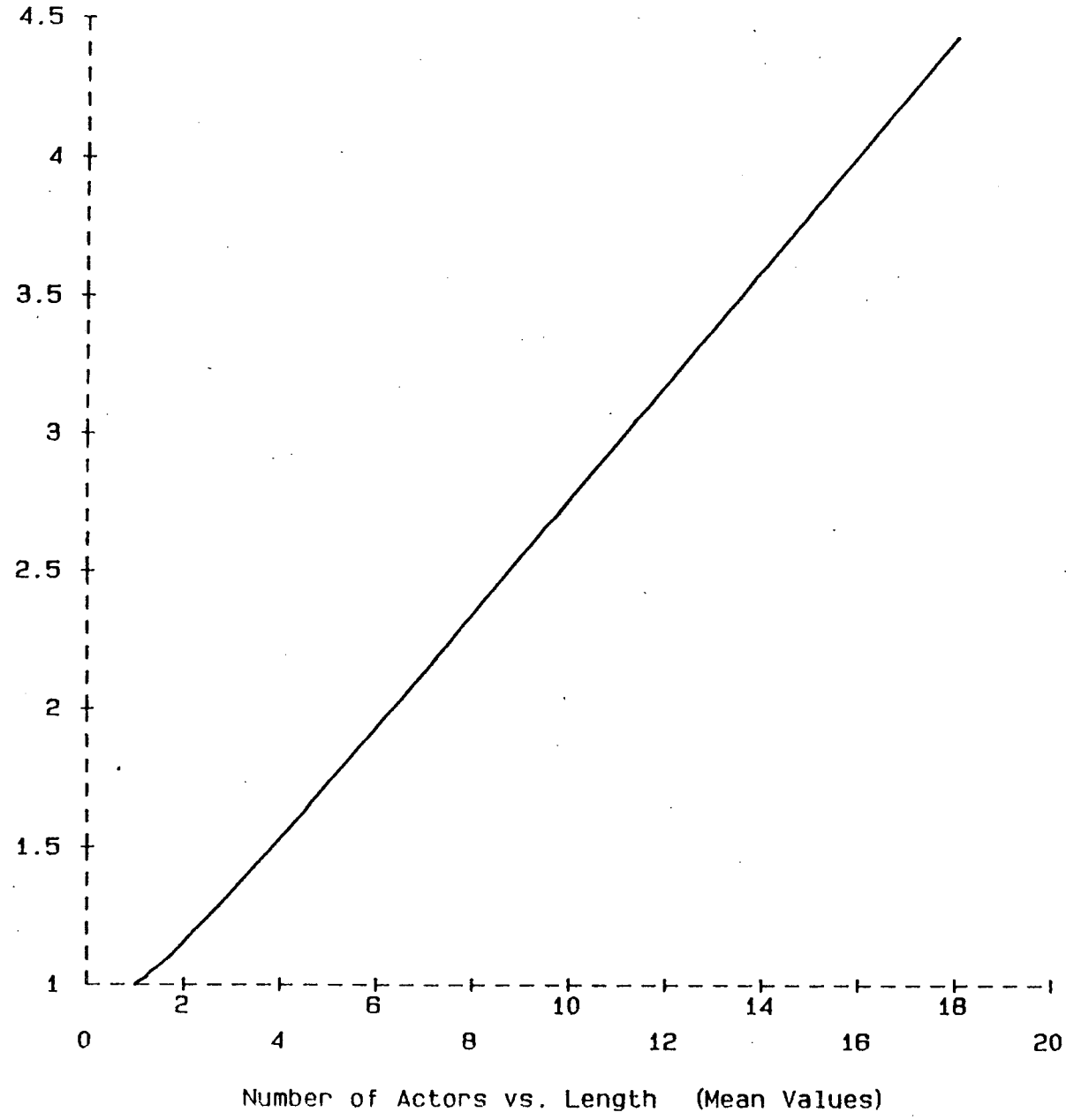

4.5
4
3.5
3
2.5
2
1.5
1
0
2
4
6
8
10
12
14
16
18
20
Number of Actors vs. Length (Mean Values)

Fig. 7

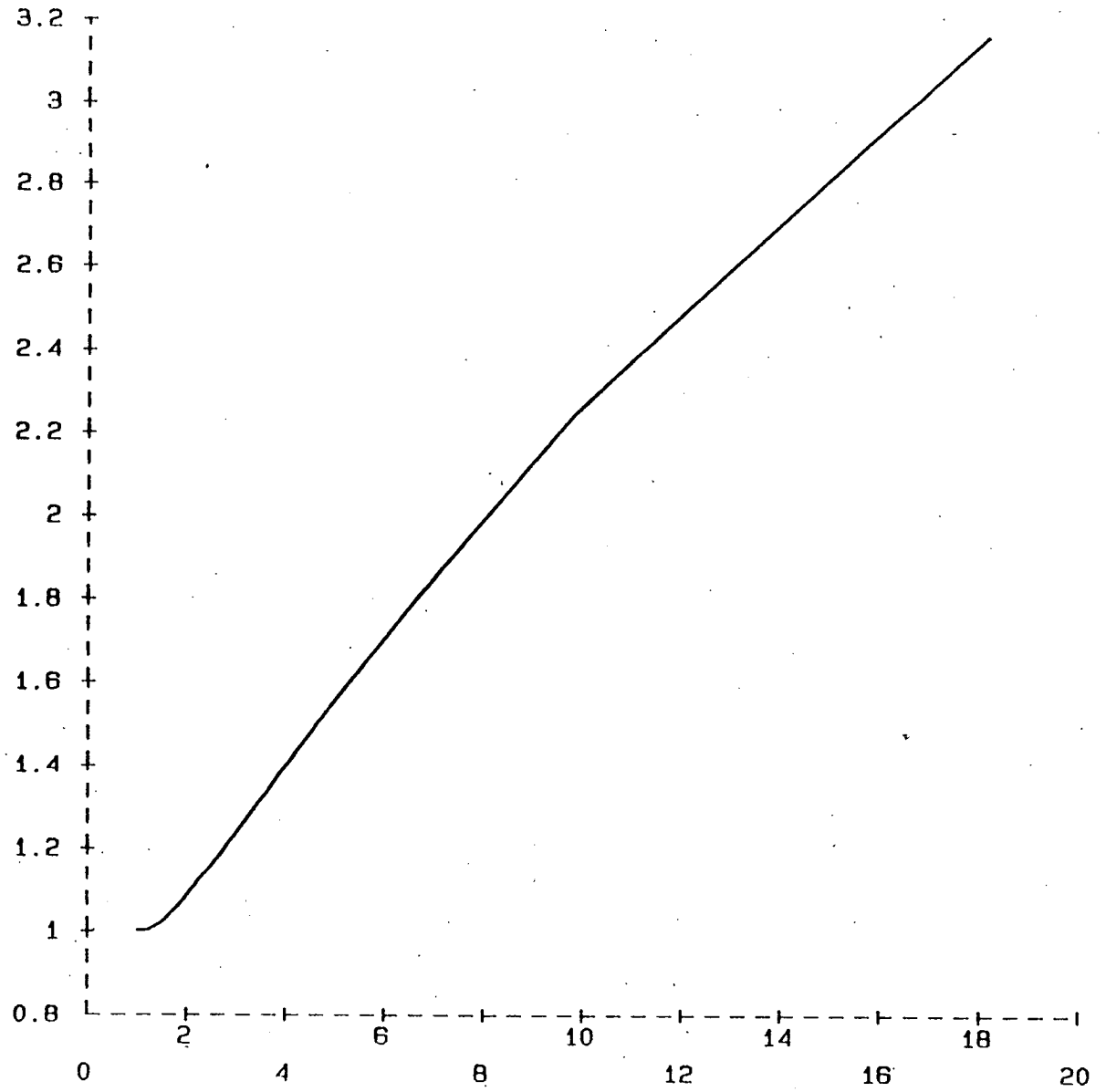

3.2
3
2.8
2.6
2.4
2.2
2
1.8
1.6
1.4
1.2
1
0.8
0
2
4
6
8
10
12
14
16
18
20
speed up vs. Length (Mean Values)

Fig. 8

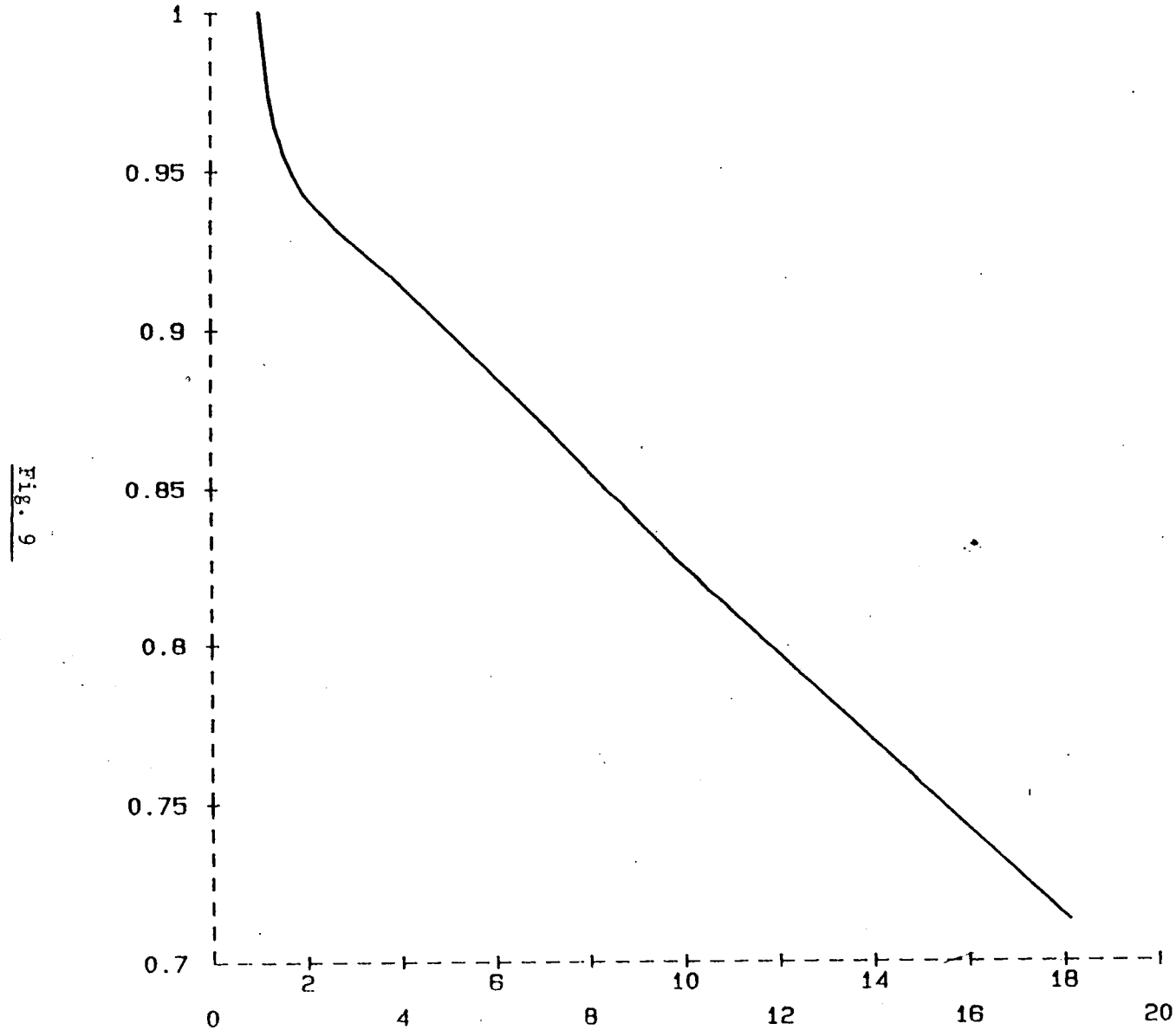

1
0.95
0.9
0.85
0.8
0.75
0.7
0
2
4
6
8
10
12
14
16
18
20
Efficiency vs. Length (Mean Values)

Fig. 9

BIBLIOGRAPHY

[BAC 82] BACCELLI F., FLEURY T., "On parsing arithmetic expressions in a multiprocessing environment", Acta Informatica 17, 287-310, 1982.

[BAE 77] J.L. BAER, C.S. ELLIS : "Model, design, and evaluation of a compiler for a parallel processing environment", IEEE soft. Eng. Vol. SE.3, N° 6, November 1977.

[BAN 79] J.P. BANATRE, J.P. ROUTEAU, L. TRILLING : "An event driven compiling technique", Communications of the ACM, January 1979 Vol. 22, N° 1.

[FIS 80] C.N. FISHER " On parsing and compiling arithmetic expressions on vector computers", ACM Trans. Program. Lang. and Syst. Vol. 2 n°2, p.203-24, April 1980.

[GON 78] R.C. GONZALES, M.G. THOMASON : "Syntactic pattern recognition", Addison Wesley, 1978.

[GOO 49] I.J. GOOD : "The number of individuals in the cascade process", Proc. Camb., Phil. Soc. 45, 1949, 360-363.

[GRI 71] D. GRIES : "Compiler construction for digital computers", 1971, John Wiley & sons, New York.

[HAR 63] T.E. HARRIS : "The theory of branching processes", 1963, Springer, Berlin.

[HOP 69] J.E. HOPCROFT et J.D. ULLMAN : "Formal languages and their relation to automata", ADDISON-WESLEY 1969.

[MIC 78] M. MICKUNAS and E. SCHELL : "Parallel compilation in a multiprocessor environment" In Proc. ACM 1978, Annu. Conf., Dec. 1978, pp. 241-246.

[McQ 79] David B. Mac QUEEN, "Models for distributed computing" Rapport de Recherche IRIA N° 351, Avril 1979.

Imprimé en France

par

l'Institut National de Recherche en Informatique et en Automatique