stichting
mathematisch
centrum

DEPARTMENT OF COMPUTER SCIENCE IW 44/76 JANUARY

R. VAN VLIET

A PROBLEM IN COLLATERAL ELABORATION

2e boerhaavestraat 49 amsterdam

Printed at the Mathematical Centre, 49, 2e Boerhaavestraat, Amsterdam.

The Mathematical Centre, founded the 11-th of February 1946, is a non-profit institution aiming at the promotion of pure mathematics and its applications. It is sponsored by the Netherlands Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O), by the Municipality of Amsterdam, by the University of Amsterdam, by the Free University at Amsterdam, and by industries.

AMS(MOS) subject classification scheme (1970): 68A10

A problem in collateral elaboration

by

R. van Vliet

ABSTRACT

While studying the language ALGOL 68, we hit on the intriguing question: "which outputs may be generated by executing the statement print ((i:=1, i:=2, i:=3))". The units in the row display are elaborated collaterally, giving rise to 3 collateral actions. At a first glance one is tempted to think that any 3-tuple of the numbers 1, 2, 3 might be output. Looking closer, one easily sees that some 3-tuples (e.g. 3, 2, 1) will not occur in a reasonable implementation. If an implementation actually elaborates the units in the row display collaterally, only 16 of the 27 3-tuples may occur.

In sections 1 - 4 the reader is made familiar with the idea of decomposing actions. Three types of actions are considered: serial, collateral and simultaneous. The concept of independent actions is introduced. Using this concept it is pointed out to what extent collateral and simultaneous actions are different and to what extent they may be considered the same.

Finally in sections 5 and 6 a somewhat generalized version of the problem mentioned above is solved.

KEY WORDS & PHRASES: *Collateral actions, serial actions, simultaneous actions, result function of an action.*

1. SERIAL, COLLATERAL AND INSEPARABLE ACTIONS

From [1], section 2.1.4.2, we cite:

A) An action may be *inseparable, serial* or *collateral*. A serial or collateral action consists of one or more other actions, termed its *direct actions*. An inseparable action does not consist of other actions.

B) A *descendent action* of another action b is a direct action either of b, or of a descendent action of b.

C) The direct actions of a serial action s take place one after the other; i.e., the completion of a direct action of s is followed by the initiation of the next direct action, if any, of s.

D) The direct actions of a collateral action are merged in time; i.e., a collateral action consists of inseparable actions, taking place one after another, each of which is chosen from among those of its descendent inseparable actions which at that moment, are active (that is, initiated and not yet completed).

In this terminology an inseparable action may be regarded as either a collateral or a serial action having one direct action. From the next section of [1] we extract:

A serial action is initiated by initiating the first of its direct actions, and it is completed when the last of its direct actions is completed.

A collateral action is initiated by initiating all of its direct actions, and its is completed when all of its direct actions are completed.

Some terminology and notation:

A descendent action of an action a is termed a *descendant* of a.
An action is *executed* when it takes place.
The decomposition of an action into its direct actions will be denoted inbetween square brackets. *Is decomposed in* is denoted by "::".
For example, let an action p have two direct actions a and b; this is denoted:

Serial	p:: [a;b]
Collateral	p:: [a,b]
Simultaneous (sec3)	p:: [a/b]

The state of an action may be *uninitiated*, *initiated*, in *execution*, or *completed*.

Note that in this terminology *initiating an action* only means giving permission to execute it eventually. If an action a is not a descendant of a collateral action, it will be executed immediately after its initiation; if a is a descendant of a collateral action, then, after the initiation of a, other initiated descendants of that collaterial action may take place before a is executed.

This description does not cover all situations encountered in daily life. It emphasizes, however, the very important concept of decomposing actions into smaller and smaller ones, until we end up with some actions we wish to regard as inseparable. Moreover it gives a clear and suitable terminology, that we will use henceforth.

Whether or not a certain action is to be regarded as inseparable, depends on the problem under consideration. For instance, as far as the text is concerned [going to the next line] on a teletype may well be regarded as inseparable; but, while studying the fingers of the typist, [going to the next line] is a serial action: [depress return key; depress linefeed key].

2. THE RESULT OF AN ACTION

Let an action a take place in a system S described by a set of independent parameters Π. To each $\pi \in \Pi$ a set of values V_π corresponds. The state of the system can be denoted by a function $\phi : \Pi \to \cup V_\pi$ such that $\phi(\pi) \in V_\pi$. Let the set of all states be denoted by Φ.

The execution of a changes the state of S. We define:
The result function of a is a function $r_a : \Phi \to \Phi$ which maps the state of S before the execution of a on the state after the execution of a.

In general only a few parameters influence the changes caused by executing a. The set of these parameters is denoted by $d_a = \{\pi \in \Pi \mid$ there exist ϕ_1 and $\phi_2 \in \Phi$ such that $\phi_1(\rho) = \phi_2(\rho)$ for all $\rho \in \Pi \neq \pi$ and $r_a(\phi_1) \neq r_a(\phi_2)\}$

Likewise, only the values of some parameters can ever be affected by executing a. The set of these parameters is denoted by i_a.

$i_a = \{\pi \in \Pi \mid \text{there is a } \phi \in \Phi \text{ such that } \left(r_a(\phi)\right)(\pi) \neq \phi(\pi)\}$

As actions may be decomposed into inseparable descendants, we would like to express the result function of an action in the result functions of its inseparable descendants. Let f.g denote the product of two functions f and g, i.e., by f.g we mean the function obtained by first applying g and then applying f.

A serial action a:: $[a_1; a_2;\ldots;a_n]$ is executed by first executing a_1, then a_2,... then a_n. So the result of a serial action a::$[a_1;a_2;\ldots;a_n]$ is $r_{a_n}.r_{a_{n-1}}\ldots r_{a_1}$

The situation is more complicated if a is a collateral action. First, let a have only two inseparable direct actions a_1 and a_2. a may be executed by first executing a_1 and then a_2 - which has as result function $r_{a_2}.r_{a_1}$ -, or by first executing a_2 and then a_1 - which has as result function $r_{a_1}.r_{a_2}$.

So the result function of a is either $r_{a_2}.r_{a_1}$ or $r_{a_1}.r_{a_2}$. Generally these two functions are different. The result of a is then said to be *unpredictable* and a_1 and a_2 are termed *dependent* actions. $r_{a_2}.r_{a_1}$ and $r_{a_1}.r_{a_2}$ are termed *the possible result functions* of a, whereas the states of S after executing $[a_1;a_2]$ or $[a_2;a_1]$ are loosely indicated as "possible results" of a.

If $r_{a_2}.r_{a_1}$ is the same function as $r_{a_1}.r_{a_2}$, then the result function of a is $r_{a_2}.r_{a_1}$, and a_1 and a_2 are termed *independent* actions.

Without going into mathematical detail, we roughly indicate the conditions under which two inseparable actions a and b are expected to be independent. χ denotes the intersection of i_a and i_b. Two actions a and b are independent if:

A) $i_a \cap d_b = \emptyset$
$i_b \cap d_a = \emptyset$
and

B) $\left(r_a(\phi)\right)(\pi) = \left(r_b(\phi)\right)(\pi)$ for all $\phi \in \Phi$ and $\pi \in \chi$.

We make some evident generalizations:

Two actions a and b are independent, if each direct action of a is independent of all direct actions of b; otherwise a and b are dependent (if a or b is inseparable, that inseparable action is regarded as the only direct action).

An action has an unpredictable result, if either
a) one of its direct actions has an unpredictable result; or
b) that action is collateral, and two of its direct actions are dependent.

The order in which the inseparable descendants of a collateral action are executed, severely influences its result if some of its direct actions are dependent. Therefore, the direct actions of a collateral action are sometimes synchronized by the use of a semaphore (see [2]). Throughout this report we will assume no synchronization.

3. SIMULTANEOUS ACTIONS

As stated above, it is not possible to classify all actions as serial, collateral or inseparable. Consider for instance [playing the piano]. The left and right hand execute simultaneous actions. Thus [playing the piano] may be decomposed into two direct actions that must take place simultaneously. [1] however, only describes inseparable actions, that take place one after the other. So the only way to describe [playing the piano] is as an inseparable action. To cope with simultaneous actions, we extend our classification as follows.

An action may be simultaneous. A simultaneous action consists of one or more direct actions, that take place simultaneously. A simultaneous action is initiated by initiating all of its direct actions. It is completed when all of its direct actions are completed.

There is a difficulty in this definition, worth spending some words on. Let p be a simultaneous action, p:: [a/b]. a and b are both serial,
a:: $[a_1;a_2;\ldots;a_n]$, b:: $[b_1;b_2;\ldots;b_m]$.
Initiating p will initiate a and b, which in turn initiate a_1 and b_1 simultaneously. a_1 and b_1 are descendants of a serial action, so they will immediately be executed. It is not clear that a_1 and b_1 are completed at the same instant of time; nor is it clear that a_n and b_m are; nor is it clear that this was the desired effect. Again synchronization of a and b may play an important role here.

Expressing the result function of a simultaneous action p, having two inseparable descendants a and b in the result functions of a and b is complicated, because we lack the tools to describe the simultaneous application of two functions.

We will assume that the following restrictions hold:

a) if a and b are independent $r_p = r_b \cdot r_a$

b) $i_p \subset i_a \cup i_b$

c) $d_p \subset d_a \cup d_b$

4. EQUIVALENCE OF SIMULTANEOUS AND COLLATERAL ACTIONS IN VAGUE COSYSTEMS

DEFINITION. A cosystem C has two elements: a system S_c (i.e., a set of parameters Π, and for each $\pi \in \Pi$ a set of possible values V_π, and a set of actions A_c).

The states in the system can again be denoted by functions $\phi : \Pi \rightarrow \cup V_\pi$, the set of all states is indicated by Φ. The elements of A_c are denoted as functions $r : \Phi \rightarrow \Phi$.

An action is said to be in the cosystem C if it can be decomposed into elements of A_c.

A cosystem C is said to be fixed if all elements of A_c are inseparable; otherwise C is termed vague.

In many applications it is possible to specify a set of actions A_c, such, that all actions in a system S_c that are considered, are composed of elements of A_c. In most cases however it is uncertain whether or not all elements of A_c are inseparable.

In vague cosystems it is difficult to determine whether or not two actions are dependent. This is due to the fact that independence of two actions was ultimately based on the independence of their inseparable descendants. But in vague cosystems we don't know the inseparable descendants of an action. This is remedied if the following condition holds (which it does usually):

For each action $a \in A_c$ and each inseparable descendant h of a $d_h \subset d_a$ and $i_h \subset i_a$. It is now easily seen that two actions a and b are independent if the conditions specified in section 2 hold.

Two arbitrary actions p and q in such a cosystem are independent if each descendent action of p belonging to A_c is independent of each descendent action of q belonging to A_c. We are tempted to replace the notion of "inseparability" by the notion "belongs to A_c".

Let us now consider the actions

p:: [a/b]
q:: [a,b]

where a and b are dependent elements of A_c. For the simultaneous action p we have to accept any result not violating the restrictions of section 3.

If the cosystem is fixed we know that the only possible results of the collateral action q are $r_b \cdot r_a$ and $r_a \cdot r_b$. But if the cosystem is vague, the only thing we know is that all results will fit the restrictions in section 3.

<u>EXAMPLE.</u> C is a cosystem.
The system S_c has three parameters x, y, z.
The set $V_x = V_y = V_z = \{\rho \mid \rho \text{ is real}, \rho > 0\}$
$A_c = \{a, b\}$ where

$$r_a = (x,y,z) \rightarrow (x,x*y,z) \text{ and}$$

$$r_b = (x,y,z) \rightarrow (x,y+z,z).$$

For the result function of the simultaneous action p:: [a/b] we clearly have to accept all functions that leave the values of x and z unchanged.

If the cosystem is fixed the possible result functions for the collateral action q:: [a,b] are

$$r_b \cdot r_a = (x,y,z) \rightarrow (x,x*y+z,z) \text{ and}$$

$$r_a \cdot r_b = (x,y,z) \rightarrow (x,x*y+x*z,z).$$

If the cosystem is vague however we cannot exclude any of the result functions of p as a possible result function of q. Suppose for instance the result function $r_q = (x,y,z) \rightarrow (x, \frac{x*y}{1+x*y*z}, z)$ would occur. As we don't know the inseparable descendants of a we cannot exclude the splitting a:: [a_1;a_2] where

$$r_{a_1} = (x,y,z) \rightarrow (x, \frac{1}{x*y}, z)$$

$$r_{a_2} = (x,y,z) \rightarrow (x, \frac{1}{y}, z)$$

as a possible decomposition of a.

Executing the descendants of q in the order a_1; b; a_2 would yield the above result function.

It is this equivalence of collateral and simultaneous actions in vague cosystems, that makes us think of collaterality as a kind of parallellism.

5. THE ALGOL 68 STATEMENT print ((i:+1, i:=2, i:=3)).

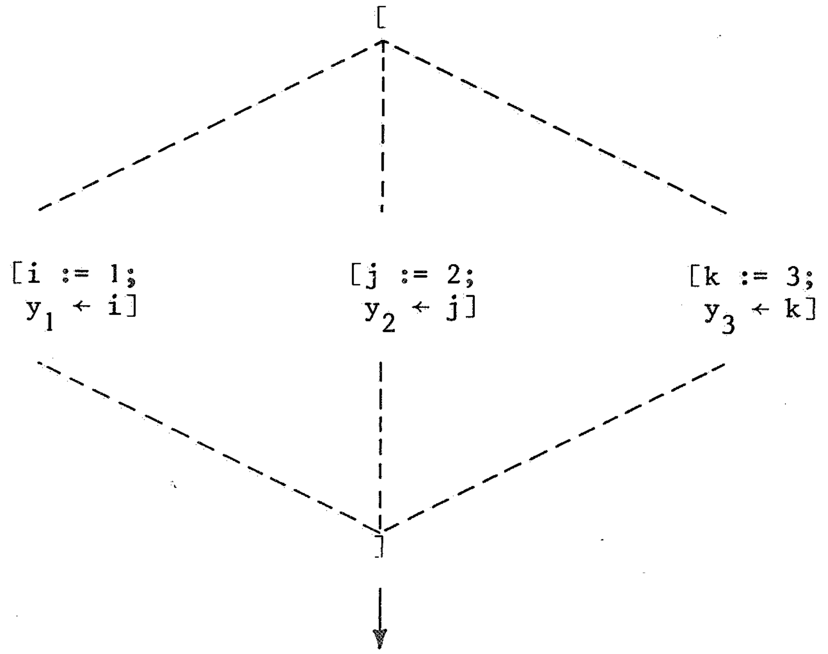
We will now study the problem what are the possible outputs of the simple ALGOL 68 program:

```
'begin'
        'int' i;
        print((i:=1, i:=2, i:=3))
'end'
```

As a first step we will consider the program

```
'begin'
        'int' i,j,k;
        print((i:=1, j:=2, k:=3))
'end'
```

Roughly following [1], the elaboration of the unit print ((i:=1, j:=2, k:=3)) can be viewed as a graph of actions as in fig. 1.



assignation to i is denoted by "i :="
dereferencing i is denoted by "i ←"

The splitting into three paths corresponds to the collateral elaboration of the three units in the rowdisplay (i:=1, j:=2, k:=3).

From fig. 1 we see that the statement print ((i:=1, j:=2, k:=3)) gives rise to a serial action, having a collateral direct action p:: [[i:=1; y_1 ← i], [j:=1; y_2 ← j], [k:=3; y_3 ← k]]. [1] only specifies a vague system, so we cannot determine the state of the system after the execution of this action.

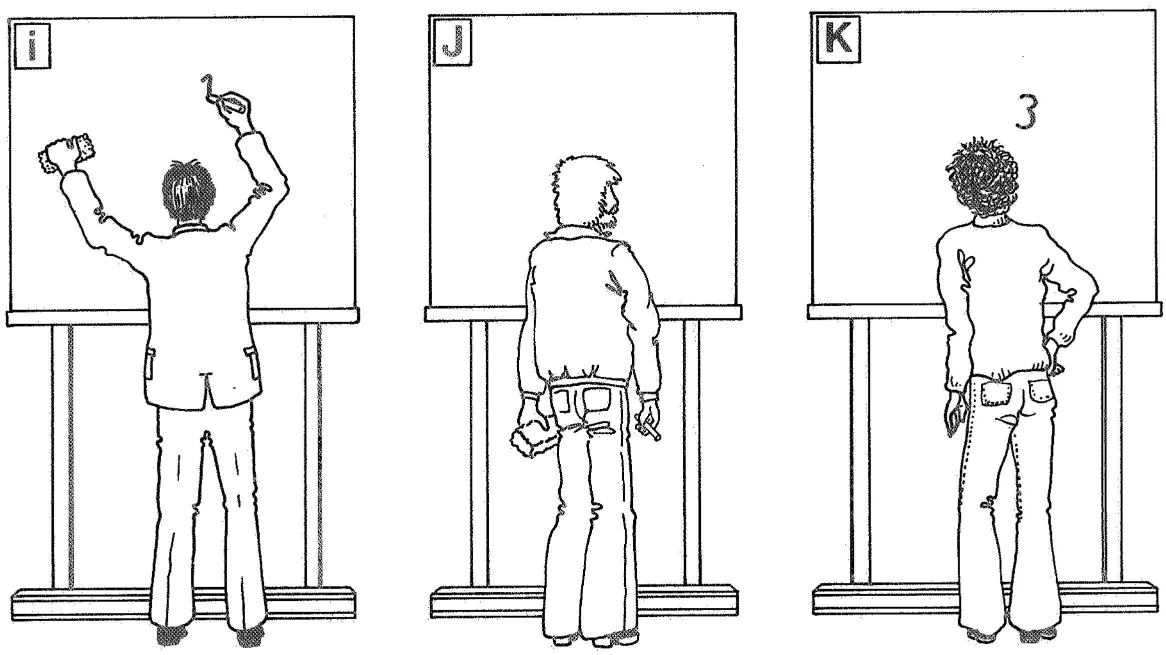
In a reasonable implementation the direct actions of p will of course be independent, thus {y_1=1, y_2=2, y_3=3} after the execution of p. We can achieve this by assuming the following types of inseparable actions: assignations [i:=1] [j:=2] [k:=3]; dereferences [y_1 ← i] [y_2 ← j] [y_3 ← k].*

The collateral action p now has three direct actions, each being a series of two inseparable actions: an assignation followed by a dereference.

* Our definition of a dereference differs of that given in [1]. This is a consequence of not going into the details of elaborating a row display.

Further we assume that the parameters of the system are (i,j,k,y_1,y_2,y_3). It is then easy to see that the direct actions of p are independent.

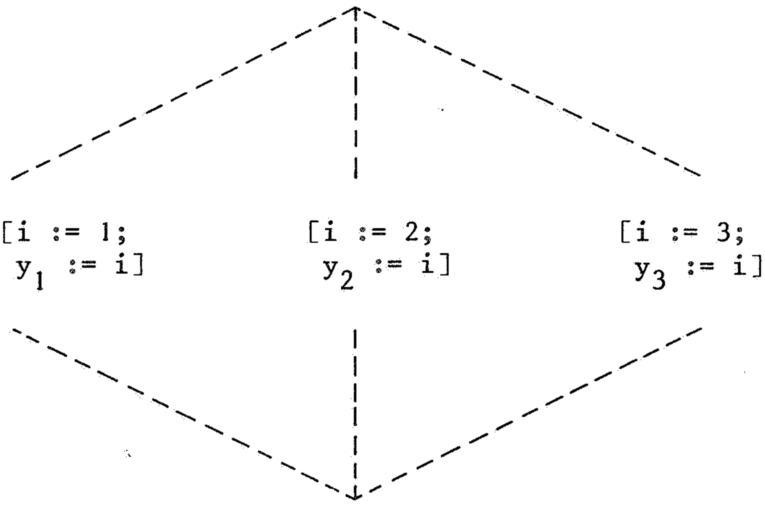
As an illustration of the actions that take place, consider the following analogous problem. Three boys are given the numbers 1, 2, and 3. Boy number 1 is given the task [clean blackboard i and write down your number on it; read the number written on blackboard i]. Boy number 2 has a same task using blackboard j and boy number 3 has the same task using blackboard k. (See fig. 2).



Which numbers will be read by the boys? Evidently an assignation corresponds to [cleaning the board and writing a number on it], and dereferencing corresponds to [reading a number].

Let us return to the question about the possible outputs of print ((i:=1, i:=2, i:=3)). As above we have to make some assumptions about our cosystem, in order to make it fixed.

Assume the system has 4 parameters (i,y_1,y_2,y_3). Inseparable actions are assignations [i:=n] (n=1,2,3), dereferences $[y_n \leftarrow i]$ (n=1,2,3). In fig. 3 we have decomposed the action print ((i:=1, i:=2, i:=3)).



It is a serial action having only one direct action p.
p:: [[i:=1; $y_1 \leftarrow$ i], [i:=2; $y_2 \leftarrow$ i], [i:=3; $y_3 \leftarrow$ i]]. Denoting by a_n the assignation [i:=n] (n=1,2,3) and by d_n the dereference [$y_n \leftarrow$ i] (n=1,2,3), we find: p:: [[a_1;d_1], [a_2;d_2], [a_3;d_3]]. A possible output is denoted by the values of y_1,y_2,y_3 after the execution of p. The assignations a_1, a_2 and a_3 are dependent. They all have a different effect on i. So the result of p is unpredictable.

The problem of the boys and the blackboards can be changed accordingly by assuming that all boys work on one and the same blackboard i. Of course they have to work collaterally, so no two boys can [clean and write] or [read] simultaneously, nor can one of the boys be [cleaning and writing] while one of the others is [reading].

The result of p depends on the order in which the assignations and dereferences take place, for example,

[a_1; d_1; a_2; d_2; a_3; d_3] will yield 1,2,3 as output.

[a_1; a_2; d_1; d_2; a_3; d_3] will yield 2,2,3 as output.

Which outputs of p are possible is a combinatorial problem, that is solved in the next section.

6. POSSIBLE OUTPUTS OF print ((i:=1, i:=2,...,i:=n)).

Under similar assumptions as in section 5, and using the same notation, we may decompose the action print ((i:=1, i:=2,...,i:=n)) into a series of actions. This series has one collateral direct action p:: $[[a_1; d_1],...,[a_n; d_n]]$. The system is described by the parameters $(i, y_1, y_2,..., y_n)$, and the possible outputs of p^n will be indicated by the values of $y_1, y_2 ..., y_n$ after the execution of p^n.

First we observe that the inseparable descendants of p^n will not be mixed arbitrarily. An action d_k is not initiated before the corresponding action a_k has been completed. So only permutation in which a_k precedes d_k for all k $(1 \le k \le n)$ will occur.

Secondly, not all possible permutations yield different outputs. For instance, the orders

$$[a_1; d_1; a_2; d_2;...a_i; d_i;...a_j; d_j;...a_n; d_n]$$

and

$$[a_1; d_1; a_2; d_2;...a_j; d_j;...a_i; d_i;...a_n; d_n]$$

will give rise to the same output.

As an illustration consider p^2:: $[[a_1;d_1], [a_2;d_2]]$.
It has 4 inseparable descendants, giving 24 permutations. Only 6 permutations can occur, and there are only 3 possible outputs: {1,1} {1,2} {2,2}.

Before continuing we introduce some sets:

s^n denotes the set:

$s^n = \{L \mid L$ is an n-tuple $(\ell_1, \ell_2,..., \ell_n)$; ℓ_i is an integral number $1 \le \ell_i \le n\}$.

As the results of p^n are indicated by n integral numbers $\{\ell_1, \ell_2, ... \ell_n\}$ for which the above relation holds, the results of p^n are elements of s^n.

s_w^n is a subset of s^n: $s_w^n = \{L \mid L \in s^n$; L is not a possible result of $p^n\}$.

G^n denotes the set $G^n = \{G \mid G$ is a directed graph on n numbered points $q_1, q_2,..., q_n$; to a given point in that graph at most one arrow is pointing; G contains no trivial cycles}.

G_0^n is a subset of G^n:G_0^n = {G | G $\in$ G^n; G contains at least one cycle}.

s'^n_w is the complement of s^n_w in s^n.

G'^n_0 is the complement of G_0^n in G^n.

Finding the possible outputs of p^n (in other words finding the elements of s'^n_w) will be done in two steps.

1. A one-to-one mapping is defined, that maps s^n on G^n
2. The following theorem is proved: $L \in s'^n_w$ *if and only if* $f(L) \in G'^n_0$

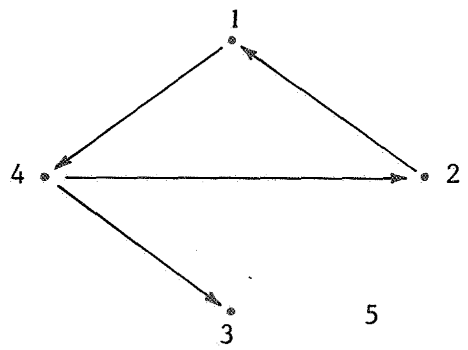
1. Let L be an element of s^n. The mapping f maps L on an element G of G^n as follows:

A. $q_1,q_2,\ldots,q_n$ are n numbered points.

B. 'For' i 'from' 1 'to' n 'do' (connect $q_{(\ell_i)}$ with q_i in the direction from $q_{(\ell_i)}$ to q_i)

C. Remove all trivial cycles.

From step b and c it is clear that G belongs to G^n. Fig. 4 shows f (2,4,4,1,5).



That this mapping is one-to-one can be seen by constructing its inverse. If G is element of G^n then $f^{-1}(G)$ is found as follows:

'for' i 'from' 1 'to' n 'do'

('if' an arrow points to q_i

'then' follow it backwards, arriving at q_j $(1 \le j \le n, j \ne i)$;

$\ell_i = j$.

'else $\ell_i = i$.

'fi')

'od'

2. The theorem is proved using two lemmas.

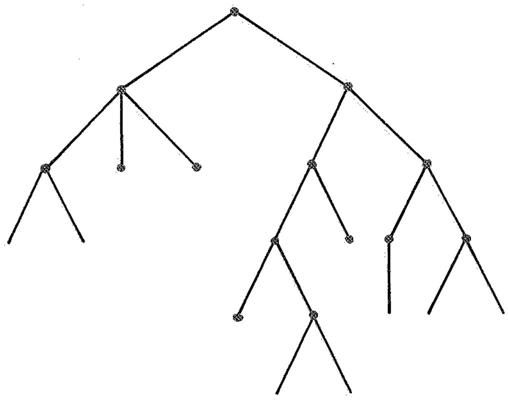
LEMMA. *If* $G \in G_0^n$ *then* $f^{-1}(G) \in s_w^n$

PROOF. G belongs to G_0^n so it has a cycle. Renumbering the points $q_1,\dots,q_n$ of G only corresponds to renumbering the direct actions of the collateral action p^n. So, without loss of generality, we may assume a cycle in G over the first k points $q_1,\dots,q_k$ ($k > 1$, as we excluded trivial cycles).

We now try to find an order in which the descendants of p^n could have been executed to yield $L = f^{-1}(G)$. As q_1 is connected to q_2, ℓ_2 must have the value 1. Similarly ℓ_3 must have the value $2,\dots,\ell_1$ must have the value k. For a while we discard all action a_i and d_i with $i > k$. The only way to get k different values in $\ell_1,\dots,\ell_k$ after execution of p^n is to have the k assignations $a_1,\dots,a_k$ and k dereferences $d_1,\dots,d_k$ in alternating order. The relation "must immediately precede" will be denoted by an arrow. ℓ_k has the value k-1, thus $a_{(k-1)} \rightarrow d_k$; similarly $a_{(k-2)} \rightarrow d_{(k-1)};\dots a_1 \rightarrow d_2$; $a_k \rightarrow d_1$. These requirements on the order of $a_1,\dots,a_k$ and $d_1,\dots,d_k$ must be combined with the requirement that a_i precedes d_i for all i. Clearly, no such order exists. So $L \in s_w^n$. □

LEMMA 2B. *If* $f(L) \in G'^n_0$ *then there is at least one way to arrange the inseparable descendants of* p^n *such, that* L *is its output.*

PROOF. $G = f(L)$ is a directed graph, not containing cycles. So it can be decomposed in a number of trees, t1,t2,...,tm. Let t be such a a tree. The nodes of t are grouped in levels; the level of a node equals the number of arrows in the path from the root to that node. So the root has level 0, under the root is level 1, then level 2, etc. (See fig. 5).

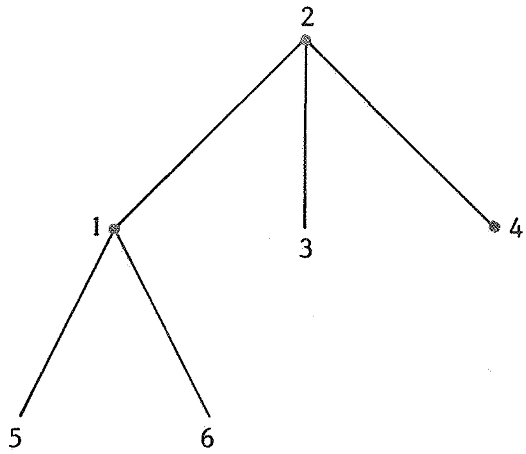


Nodes may further be divided into two groups: leaves and branches. Leaves are nodes, that have no descendent nodes, i.e., have no arrows starting from them. Branches do have descendent nodes. All nodes to which an arrow is pointing from a given node q, are termed *daughters* of q, and q is termed their *mother*.

Let the levels in t be numbered from 0 to k. We will now indicate an order in which the assignations and dereferences corresponding to the nodes of t, may be grouped, to yield L.

1. If k=0 then step 8 is taken.
2. Set a counter i to k.
3. Insert the assignations corresponding to the leaves of level i.
4. Let a pointer p point to the leftmost node of level q.
5. Let the mother of the node pointed to by p be N. Insert the assignation corresponding to N followed by the dereferences corresponding to the daughters of N. Make p point to the rightmost daughter of N.
6. If the node pointed to by p has a righthand neighbour, then make p point to that neighbour, and retake step 5.
7. i is decreased by 1. If i > 0 then step 3 is retaken.
8. If the root of t is a leave, then insert the assignation corresponding to it.
9. Insert the dereference corresponding to the root.

Fig. 6 illustrates this algorithm.



To this tree corresponds the order

[a5;a6; a1;d5;d6; a3;a4; a2;d1;d3;d4; d2]

That this is a correct way of ordering the actions corresponding to the nodes of t can be seen by observing:

A. While treating level i, all assignations corresponding to level i-1 are inserted. Furthermore all dereferences corresponding to nodes in level i are inserted. As the levels are treated in reverse order, an assignation corresponding to a branch always precedes the corresponding dereference.

B. If a node is a leave, then in step 3 special care is taken, to ensure that the assignation corresponding to it precedes its corresponding dereference.

C. Let, in t, q_i be a daughter of q_j. Then a_j precedes d_i, and thus (assuming that no other assignation is put inbetween these actions) ℓ_i has the value j. Hence in f(L) q_j points at q_i, which is indeed the case.

D. Let now q_i be such, that it is not a daughter of any other node, i.e. it is the root of t. The only nodes in f(L) to which no other nodes are pointing, are nodes q_j, such that ℓ_j has the value j. This is indeed guaranteed by the way in which the algorithm treats the root.

We arrange the inseparable descendants of p^n such, that the actions corresponding to a specific tree of f(L) are clustered together and have the designed order. Each cluster ends with a dereference, whereas the successor starts with an assignation. This implies that clusters cannot influence one another, so any permutation of clusters will yield L as output. □

CORROLARY. *The number of possible outputs of* p^n *equals* $(n+1)^{n-1}$

PROOF. Add a special point q_0 to $f(L)$.
When q_0 is connected to all roots of the component trees of $f(L)$ then we achieve a tree t_L. On the other hand, given a tree of $n+1$ numbered nodes $q_0, q_1, \ldots, q_n$, we can erase q_0 and its connections. The resulting graph is element of G'^n_0. So there is a one-to-one correspondence between the elements of G'^n_0 - and hence the possible outputs of p^n - and the trees on $n+1$ numbered points. According to [3] the number of such trees equals $(n+1)^{n-1}$. □

7. GENERALIZATION

To conclude this document we think about the following more general question.

Given a collateral action $p :: [a^1, a^2, \ldots, a^n]$.

Each action a^i can be decomposed into a series of inseparable descendants $a^i :: [a^i_1; a^i_2; \ldots; a^i_{k_i}]$ what is the maximum number of possible results of p.

Of course, this number is maximal if each allowed way of merging the actions a^i_j yields a different result. So the question simplifies to counting all allowed ways of merging the actions a^i_j.

From combinatorics we know that there are $(k_1 + k_2 + \ldots + k_n)!$ ways of merging the actions a^i_j. However there is only one way in which the descendants of an action a^i can be grouped. We thus divide by
$k_1! * k_2! \ldots * k_n!$
The maximum number of possible results of p equals $\dfrac{(\Sigma^n_{i=1} k_i)!}{\Pi^n_{i=1} k_i!}$

The answer to the question above is general and simple. The difficulty in finding the actual number of different possible results of collateral actions is to distinguish which of the allowed ways of merging the descendants yield identical results. This fully depends on the actions under consideration, and cannot be solved generally.

REFERENCES

[1] VAN WIJNGAARDEN, A., et al. (eds.), *Revised Report on the Algorithmic Language ALGOL 68*, Mei 1975.

[2] DIJKSTRA, E.W., *Co-operating Sequential Process*, in: F. Genugs (ed.), Programming Languages, Academic Press, London, 1968.

[3] HARARY, F. & E.M. PALMER, *The graphical enumeration*, Academic Press, New York and London, 1968.