

Spring 5-1-1977

# All Argument Selection May Be Done First ; CU- CS-108-77

David B. Benson  
*University of Colorado Boulder*

Follow this and additional works at: [http://scholar.colorado.edu/csci\\_techreports](http://scholar.colorado.edu/csci_techreports)

---

## Recommended Citation

Benson, David B., "All Argument Selection May Be Done First ; CU-CS-108-77" (1977). *Computer Science Technical Reports*. 106.  
[http://scholar.colorado.edu/csci\\_techreports/106](http://scholar.colorado.edu/csci_techreports/106)

This Technical Report is brought to you for free and open access by Computer Science at CU Scholar. It has been accepted for inclusion in Computer Science Technical Reports by an authorized administrator of CU Scholar. For more information, please contact [cuscholaradmin@colorado.edu](mailto:cuscholaradmin@colorado.edu).

ALL ARGUMENT SELECTION

MAY BE DONE FIRST\*

David B. Benson\*\*

Department of Computer Science  
University of Colorado at Boulder  
Boulder, Colorado 80309

#CU-CS-108-77

May, 1977

\* Research supported in part by NSF grant MCS 73-03497A01.

\*\* Visiting from Washington State University, Pullman, Washington, 99164.



Running Head: Select Arguments First

Mailing Address: 6 Jun 77 - 20 Aug 77

Mathematics  
University of Denver  
Denver, Colorado 80210

20 Aug 77 onwards

Computer Science  
Washington State University  
Pullman, Washington 99164

List of Symbols:

$\mathbb{I}$ ,  $\mathbb{S}$ , etc. bold-face

$\langle \rangle$ , pointed brackets. There are no less-than or greater-than signs in the paper.

$\rightarrow$  arrow with open arrow-head.

$\dashrightarrow$  barred arrow.

$\circ$  open centered bullet.

ALGOL small capital letters.

$\mathbb{N}$  special letter N that looks like that.

$\tau$ ,  $\pi$ ,  $\delta$ ,  $\omega$  Greek lower case letters.

$\mathcal{X}$  Script capital X

$\Sigma$ ,  $\Omega$  Greek capital letters.



## ABSTRACT

There are many presentations of a function in terms of its 'primitive' components. These may involve the selection, permutation and duplication of partial results. We show that all argument selection and rearrangement may precede the calculation of the 'pure' function, wherein no argument rearrangement occurs.



## INTRODUCTION

View each function computation as requiring its arguments in a particular order, as is true of programming languages and computer hardware. Given a collection of 'primitive' functions, one duplicates, selects and permutes the partial results in order to compute any of the possible functions which may be built out of the primitive collection. In this note we show that all duplication, selection and permutation may be done first; thereafter the computation is pure, requiring no further rearrangement. This result is folklore to some but not all, and is certainly used in certain computational settings. The point here is to place this fact in its most general setting.

Perhaps the best intuition about this fact is in pipeline computer architectures. The arguments are selected from memory and placed on the proper combination of lines to be fed into the ALU. There is no further control of the partial results.

Here is a small example. Let  $\pi(i)$  denote projection to the  $i^{\text{th}}$  coordinate. Let the natural numbers denote the number and ordering of the arguments required for each function. A program to compute  $x+x$  may then be written as

$$P0: \quad 1 \xrightarrow{\langle \pi_0, \pi_0 \rangle} 2 \xrightarrow{+} 1$$

where  $\langle \pi_0, \pi_0 \rangle: 1 \rightarrow 2$  is the program statement to duplicate the argument and  $+: 2 \rightarrow 1$  is the program statement for addition.

To compute the pair of results  $\langle x+x, x+x \rangle$ , the following program uses only one addition.

$$P1: \quad 1 \xrightarrow{\langle \pi_0, \pi_0 \rangle} 2 \xrightarrow{+} 1 \xrightarrow{\langle \pi_0, \pi_0 \rangle} 2$$



P1 computes  $x+x$  and then duplicates the result. However, on a parallel processor, program P2 may result in less elapsed time.

$$P2: \quad 1 \xrightarrow{\langle \pi_0, \pi_0, \pi_0, \pi_0 \rangle} 4 \xrightarrow{(+,+)} 2$$

where  $(+,+)$  is two copies of  $+$  running in parallel. Program P2 is in the canonical form as all the projections occur before any of the "actual" computations.

One may view this fact in programming language terms. Consider the code segment

```

X := F(Y);
  :
  :
... G(X,Z)
  :
  :
IF X=0 THEN ...

```

where the second and third occurrences of  $X$  contain the value of  $F(Y)$ . The second and third lines of code use argument selection functions to obtain the value of  $X$  each time it is used. The result here shows that indeed the code segment may be rewritten as

```

Comment  X := F(Y);  Endcomment
  :
  :
... G(F(Y),Z)
  :
  :
IF F(Y)=0 THEN ...

```

with the function recomputed at each use. Obviously these considerations are closely related to the call-by-name mechanism of ALGOL 60.

To place the result in its proper theoretical setting requires the use of algebraic theories. The plan is: After introducing the notation and defining algebraic theories, concrete free theories are defined. We show that indeed concrete free theories are algebraic theories and note that the arrows (morphisms) of a concrete free theory are almost in the desired form. It is then easy to obtain the result. Then via standard algebraic methods, every arrow of an arbitrary algebraic theory has a representation in a concrete free theory, proving the theorem. Some discourse in aid of intuition is interwoven.

#### NOTATION

We are working with categories and a modest familiarity is assumed. We compose arrows (morphisms) of a category in arrow-order: The composition of  $f:a \rightarrow b$  with  $g:b \rightarrow c$  is denoted  $f \circ g$  or  $fg$ . A sequence of  $n$  arguments,  $\langle x_0, \dots, x_{n-1} \rangle$  is sometimes denoted just by the natural number  $n$ . Let  $\mathbb{N}$  denote the set of natural numbers.

$\underline{\mathbb{N}}$  is the category with object set  $\mathbb{N}$  and as arrows from  $m$  to  $n$ , the functions  $f:m \rightarrow n$ . Let  $\pi^{\text{op}}(i):1 \rightarrow n$  denote the function which has  $i$  as value. That is,  $\pi^{\text{op}}(i):0 \rightarrow i$ . Then every function  $f:m \rightarrow n$  in  $\underline{\mathbb{N}}$  is isomorphic to some list  $\langle \pi^{\text{op}}(i_0), \dots, \pi^{\text{op}}(i_{m-1}) \rangle : m \rightarrow n$ . The individual  $\pi^{\text{op}}$  specify the function value for each individual argument. We fail to distinguish  $f:m \rightarrow n$  from its isomorphic representation as a list.

$\underline{\mathbb{N}}^{\text{op}}$  is the category  $\underline{\mathbb{N}}$  with all the arrows reversed. In  $\underline{\mathbb{N}}^{\text{op}}$  we have arrows  $\pi(i):n \rightarrow 1$  which are the reversals of the  $\pi^{\text{op}}(i)$  available

in  $\mathbb{N}$ . The  $\pi(i)$  in  $\mathbb{N}^{op}$  are called projections. The reason for this name follows: Let  $F:\mathbb{N}^{op} \rightarrow \underline{\text{set}}$  be a functor such that  $F(1)=A$  and all categorical products are preserved. Then  $F(n)=A^n$  for all  $n \in \mathbb{N}$ , and  $F(\pi(i):n \rightarrow 1):A^n \rightarrow A^1$  is the projection function from  $A^n$  to  $A^1 \cong A$  along the  $i^{\text{th}}$  coordinate. Thus  $\mathbb{N}^{op}$  may be viewed as the collection of all possible program statements for argument selection, replication and permutation. Every arrow in  $\mathbb{N}^{op}$  is a list of projections and is called a base arrow. We sometimes use the notation  $\langle x_{i_0}, x_{i_1}, \dots, x_{i_{n-1}} \rangle$  for  $\langle \pi(i_0), \pi(i_1), \dots, \pi(i_{n-1}) \rangle$ .

### ALGEBRAIC THEORIES

An algebraic theory is a category  $\mathbb{T}$  such that  $\mathbb{N}$  is the object set of  $\mathbb{T}$ ,  $\mathbb{N}^{op}$  is a subcategory of  $\mathbb{T}$ , and the projections of  $\mathbb{N}^{op}$  are projections in  $\mathbb{T}$ . Specifically, this last statement means that every arrow  $f:m \rightarrow n$  in  $\mathbb{T}$  is isomorphic to a list  $\langle f_1, \dots, f_n \rangle:m \rightarrow n$  such that  $f \circ \pi(i) = f_i$ .

As before, consider a functor  $F:\mathbb{T} \rightarrow \underline{\text{set}}$  such that  $F(1)=A$  and categorical products are preserved. Each arrow  $f:m \rightarrow n$  transforms to a function  $F(f):A^m \rightarrow A^n$ , so the arrow  $f:m \rightarrow n$  may be viewed as a highly abstract program statement for computing an  $n$ -list of results from  $m$  arguments. For additional information on algebraic theories, consult [1,2,3,4,5,6,7].

We construct a particular algebraic theory, the concrete free theory, taken with minor modifications from [1]. The basic idea is that lists of trees with variables are the arrows of the concrete free theory. The arrows here will point from the variables at the leaves to the root, while ADJ runs them the other way.

An operator domain is a function  $\Omega: \mathbb{N} \rightarrow \text{sets}$ . Each  $\Omega(n)$  is to be thought of as the set of  $n$ -ary operators. To view  $\Omega$  as a ranked alphabet  $\langle \Sigma, n \rangle$ , let  $\Sigma = \bigcup_{n \in \mathbb{N}} \Omega(n)$  and let  $r: \Sigma \rightarrow \mathbb{N}$  be the relation such that  $\langle \omega, n \rangle \in r$  iff  $\omega \in \Omega(n)$ . Let  $x: \mathbb{N} \rightarrow X$  be an infinite sequence of "variables",  $x_0, x_1, x_2, \dots$  such that  $X \cap \Omega(n) = \emptyset$  for all  $n \in \mathbb{N}$ . For each  $n$ , let  $X(n) = \{x_0, x_1, \dots, x_{n-1}\}$ . We now define trees over the operation domain  $\Omega$  and variables  $x$  using the standard linear notation for trees.

Let  $T(n)$  be the smallest set such that

- (i)  $X(n) \subseteq T(n)$
- (ii) if  $\sigma \in \Omega(p)$  and  $\langle t_0, \dots, t_{p-1} \rangle \in (T(n))^p$   
then  $\sigma(t_0, \dots, t_{p-1}) \in T(n)$ .

Note that as  $(T(n))^0 = \{\langle \rangle\}$ ,  $\sigma(\langle \rangle) \in T(n)$  for all  $\sigma \in \Omega(0)$ .

DEFINITION. The concrete free theory over  $\Omega$  and  $x$  is the category  $\Omega C$  with  $\mathbb{N}$  as the set of objects and arrows  $t: m \rightarrow n$  where  $t$  is an  $n$ -list of trees in  $T(m)$ ,  $t = \langle t_0, t_1, \dots, t_{n-1} \rangle$  with each  $t_i \in T(m)$ . The composition in  $\Omega C$  is the substitution of trees for variables: For  $t = \langle t_0, \dots, t_{m-1} \rangle: k \rightarrow m$  and  $t' = \langle t'_0, \dots, t'_{n-1} \rangle: m \rightarrow n$ ,

$$t \circ t' = \langle t''_0, \dots, t''_{n-1} \rangle: k \rightarrow n$$

where each  $t''_i$  is the result of simultaneously replacing each occurrence of  $x_j$  in  $t'_i$  by  $t_j$  for all  $j \in m$ . The identities of  $\Omega C$  are  $\text{id}_n = \langle x_0, x_1, \dots, x_{n-1} \rangle$ .

As the composition in a concrete free theory is associative, [1], and the identities act as categorical identities, e.g.,  $\langle t_0, \dots, t_{n-1} \rangle \circ \langle x_0, \dots, x_{n-1} \rangle = \langle t_0, \dots, t_{n-1} \rangle$ , a concrete free theory is a

category. There is a map

$$\alpha \longmapsto \alpha(\tilde{x}_0, \dots, \tilde{x}_{n-1}), \quad \alpha \in \Omega(n),$$

imbedding each operation of  $\Omega$  into  $\Omega\mathcal{C}$  as a tree of height one.

To show that  $\Omega\mathcal{C}$  is an algebraic theory, it suffices to demonstrate a faithful functor  $F: \tilde{N}^{\text{op}} \rightarrow \Omega\mathcal{C}$  which is bijective on objects and preserves products. Product preservation guarantees that the projections of  $\tilde{N}^{\text{op}}$  act as projections in  $\Omega\mathcal{C}$ . To this end, let  $F$  be the identity map on objects. Recall that an arrow of  $\tilde{N}$  is a function  $f: n \rightarrow m$ . Corresponding to  $f$  is a sequence of variables  $\langle x_{f(0)}, x_{f(1)}, \dots, x_{f(n-1)} \rangle$ . Define  $F(f^{\text{op}}: m \rightarrow n) = \langle x_{f(0)}, x_{f(1)}, \dots, x_{f(n-1)} \rangle: m \rightarrow n$ . All that we have done is change the notation from functions on finite cardinals (equivalently, lists of selection functions  $\pi^{\text{op}}$ ) to base arrows written as lists of variables.  $F: \tilde{N}^{\text{op}} \rightarrow \Omega\mathcal{C}$  is faithful since each function  $f: n \rightarrow m$  determines a unique list of variables.

PROPOSITION.  $F: \tilde{N}^{\text{op}} \rightarrow \Omega\mathcal{C}$  preserves products.

Proof: Let

$$m \xleftarrow{\delta_m = \langle \pi_0, \dots, \pi_{m-1} \rangle} m+n \xrightarrow{\delta_n = \langle \pi_m, \dots, \pi_{m+n-1} \rangle} n$$

be a product in  $\tilde{N}^{\text{op}}$ . Then  $F(\delta_m) = \langle x_0, \dots, x_{m-1} \rangle: m+n \rightarrow m$  and  $F(\delta_n) = \langle x_{m+1}, \dots, x_{m+n-1} \rangle: m+n \rightarrow n$ . Let  $t = \langle t_0, \dots, t_{m-1} \rangle: k \rightarrow m$  and  $t' = \langle t'_0, \dots, t'_{n-1} \rangle: k \rightarrow n$  be any pair of arrows in  $\Omega\mathcal{C}$  over  $k$  variables.

The arrow

$$t'' = \langle t_0, t_1, \dots, t_{m-1}, t'_0, t'_1, \dots, t'_{n-1} \rangle: k \rightarrow m+n$$

clearly satisfies  $t'' \circ F(\delta_m) = t$  and  $t'' \circ F(\delta_n) = t'$  and is the unique arrow with these properties.

The conclusion is that  $\Omega C$  is an algebraic theory.

SELECTING ARGUMENTS FIRST

Each  $n$ -list of trees in  $\Omega C$  is almost in the desired form. The argument selection occurs at the frontiers of the trees as described by the particular  $x_j$  at the leaves. The interiors of the trees specify the computation to be performed. However, we want the argument selection to be represented as a first step, to be followed by a "pure" representation of trees.

DEFINITION. The  $x$ -frontier of an arrow  $t:m \rightarrow 1$  in  $\Omega C$  is defined by:

$$\begin{aligned} \text{fr}_x(x_j) &= x_j, \quad j \in \mathbb{N} \\ \text{fr}_x(\sigma(t_0, \dots, t_{n-1})) &= \text{fr}_x(t_0) \cdots \text{fr}_x(t_{n-1}) \\ \text{fr}_x(\sigma()) &= \lambda, \text{ the null string} \end{aligned}$$

DEFINITION. An arrow  $t:n \rightarrow 1$  in  $\Omega C$  is pure if  $\text{fr}_x(t) = x_0 x_1 \cdots x_{n-1}$ .

A pure arrow is a tree in the traditional sense: No argument selection is implied at the leaves of the tree. Consider an arrow  $t:k \rightarrow 1$  with  $x$ -frontier  $x_{j_0} x_{j_1} \cdots x_{j_{m-1}}$ . Since the variables are just place-holders, there is a base arrow  $\langle \pi(j_0), \dots, \pi(j_{m-1}) \rangle : k \rightarrow m$  and a pure tree  $t':m \rightarrow 1$  such that  $t = \langle \pi(j_0), \dots, \pi(j_{m-1}) \rangle \circ t'$ . In variable notation, the base arrow is  $\langle x_{j_0}, \dots, x_{j_{m-1}} \rangle$ .

The above gives the decomposition for arrows of the form  $k \rightarrow 1$ . To extend to arbitrary arrows of  $\Omega C$  requires additional apparatus. Define the parallel combination of arrows  $t:m \rightarrow n$  and  $t':m' \rightarrow n'$  as

$$(t, t') = \langle \delta_m \circ t, \delta_{m'} \circ t' \rangle : m+m' \rightarrow n+n'$$

where  $\delta_m = \langle \pi(0), \dots, \pi(m-1) \rangle : m+m' \rightarrow m$

and  $\delta_{m'} = \langle \pi(m), \dots, \pi(m+m'-1) \rangle : m+m' \rightarrow m'$ .

For further details, see [3,p.110] and [5,p.191]. The parallel combination of  $t:m \rightarrow n$  and  $t':m' \rightarrow n'$  may be visualized as the computations of  $t$  and  $t'$  proceeding in parallel on disjoint sets of variables. Recall the example in the introduction.

Rewriting [5,(2.5.16)], we have

$$\langle b_0 \circ t_0, b_1 \circ t_1 \rangle = \langle b_0, b_1 \rangle \circ (t_0, t_1)$$

for  $b_i:k_i \rightarrow m_i$ ,  $t_i:m_i \rightarrow n_i$ ,  $i=0,1$ . Now any arrow  $t = \langle t_0, \dots, t_{n-1} \rangle : m \rightarrow n$  in  $\Omega C$  has a decomposition  $\langle b_0 t'_0, \dots, b_{n-1} t'_{n-1} \rangle$  where each  $b_i$  is a base arrow and each  $t'_i$  is pure. Applying the above result on parallel combinations, which obviously extends to any number of components,

$$t = \langle b_0, \dots, b_{n-1} \rangle \circ (t'_0, \dots, t'_{n-1})$$

demonstrating that all argument selection may precede the parallel combination of pure trees.

It now remains to extend this result to an arbitrary algebraic theory. In [1] it is proved that each concrete free theory is a free algebraic theory generated by an operator domain. It is a general result of universal algebra that every algebraic theory is an epimorphic image of a free theory. If  $\underline{T}$  is an arbitrary algebraic theory with a production preserving epifunctor  $\Omega C \rightarrow \underline{T}$ , then each arrow of  $\underline{T}$  is an equivalence class of arrows in  $\Omega C$ . Pick any representative,  $t$ , of the equivalence class and apply the above argument to obtain the decomposition. As the equivalence relation induced by the epifunctor is a congruence for composition, the decomposition exists in  $\underline{T}$ . That is, using the notation from before,

$$[t]:m \rightarrow n = [\langle b_0, \dots, b_{n-1} \rangle] \circ [(t'_0, \dots, t'_{n-1})]$$

and as parallel combinations and lists commute with the congruence, we have proved the

THEOREM. Let  $\Omega C$  be a concrete free theory and  $\mathcal{T}$  an algebraic theory equipped with a product-preserving epifunctor  $\Omega C \rightarrow \mathcal{T}$ . For each arrow  $[t]:m \rightarrow n$  of  $\mathcal{T}$  there are base arrows  $b_i$  and pure arrows  $t_i$  in  $\Omega C$  such that

$$[t] = \langle [b_0], \dots, [b_{n-1}] \rangle \circ ([t_0], \dots, [t_{n-1}]).$$

### CONCLUSION

It is difficult to conceive of a model of computation which cannot be cast into the form of an algebraic theory. Even models for interation [5] and recursion [2] fit the mold. Therefore the theorem gives a canonical form for computations, irrespective of the model. This canonical form is implicitly used in studies of code generation, [8], to guarantee expression trees rather than dags, and the form may be of use in the study of parallel processing. It is also of value in the study of nondeterministic programs, as will be demonstrated in a subsequent paper. Further study of the consequences of the form in terms of [2,5] may be enlightening.



REFERENCES

- [1] ADJ {J. A. Goguen, J. W. Thatcher, E. G. Wagner, J. B. Wright},  
An introduction to categories, algebraic theories and algebras,  
IBM Research, RC 5369, 1975.
- [2] —, Rational Algebraic theories and fixed-point solutions,  
Proc. 17<sup>th</sup> Symp. Foundations Comp. Sci., IEEE, 1976, 147-158.
- [3] D. B. Benson, Semantic preserving translations, Math. Sys. Theory  
8:2 (1974), 105-126.
- [4] S. Eilenberg and J. B. Wright, Automata in general algebras,  
Inform. and Control 11:4 (1967).
- [5] C. C. Elgot, Monadic computation and iterative algebraic theories,  
in Logic Colloquium '73 (Rose and Sheperdson, Eds.), American  
Elsevier Publ. Co., New York, 1975, 175-230.
- [6] F. W. Lawvere, Functorial semantics of algebraic theories,  
Proc. Nat. Acad. Sci. 50:5 (1963), 869-872.
- [7] B. Pareigis, Categories and functors, Academic Press, New York,  
1970.
- [8] R. Sethi and J. D. Ullman, The generation of optimal code for  
arithmetic expressions, J. Assoc. Comp. Mach. 17:4 (1970),  
715-728.