Feb 19 1975
John Backus

## Class Notes, I

### A variable-free functional programming system, FP

We construct an informal system FP comprising two kinds of elements: (1) objects $o \in \mathcal{O}$ and (2) functions $f \in F$. Every function $f \in F$ is a partial function mapping objects into objects:

$$f : \mathcal{O} \longrightarrow \mathcal{O}.$$

FP has a single operation [:] called application which maps a subset of $F \times \mathcal{O}$ into $\mathcal{O}$. Thus we write $f{:}x$ for the object [if it exists] obtained by applying the function $f$ to the object $x$.

The primitive objects of FP are called atoms. From any set of objects $o_1, \ldots, o_n$ a new object $\langle o_1, \ldots, o_n \rangle$ called a sequence may be constructed having $o_k$ as its $k^{th}$ member. The null sequence, $\phi$, has no members. Atoms and sequences are the only objects of FP.

FP has a number of primitive functions; and a number of functional forms each of which accepts one or more functions and/or objects and forms a new function from these parameters. We shall feel free to enlarge either the set of primitives [functions] or the set of [functional] forms at any time. For any given set of primitives and forms the functions of FP

Feb 14 1975
John Backus

## Class Notes I cont'd

are those which can be built with a finite number of forms from the primitives.

## Notation for describing functions

(1) We may write $f:g:h:x$ for $f:(g:(h:x))$ etc.

(2) We extend $\mathcal{O}$ to include $\perp$, "undefined", and introduce the equivalence

$$x \equiv \perp \implies \langle \ldots, x, \ldots \rangle \equiv \perp .$$

Thus an object now either has no occurrences of $\perp$ in it, or it "is" $\perp$.

(3) We use conditional expressions $p \rightarrow e;f$ to denote $e$ when $p$ is true and $f$ otherwise. Right associated parentheses may be omitted:

$$p_1 \rightarrow e_1 ; (p_2 \rightarrow e ; f) \equiv p_1 \rightarrow e_1 ; p_2 \rightarrow e ; f$$

## Functional forms

Let $x, x_1, \ldots, x_n, y_1, \ldots, y_n \in \mathcal{O}$. Let $n$ be an integer and an atom. Let $f, g, h, p, f_1, \ldots, f_n \in F$.

(1) Composition $f_1 \cdot f_2 \cdot \cdots \cdot f_n$

$$f_1 \cdot f_2 \cdot \cdots \cdot f_n : x \equiv f_1 : f_2 : \cdots : f_n : x$$

(2) Apply to all $\alpha f$

$$\alpha f : x \equiv x = \emptyset \rightarrow \emptyset ; \; x = \langle x_1, \ldots, x_n \rangle \rightarrow \langle f:x_1, \ldots, f:x_n \rangle ; \perp$$

(3) n functions $[f_1, \ldots, f_n]$

$$[f_1, \ldots, f_n] : x \equiv \langle f_1 : x, \ldots, f_n : x \rangle$$

(4) Condition $(p \rightarrow f ; g)$

$$(p \underset{\text{in FP}}{\rightarrow} f ; g) : x \equiv p:x = T \underset{\text{in metalanguage}}{\rightarrow} f:x \; ; \; p:x = F \rightarrow g:x \; ; \perp$$

Feb 19, 1975
John Backus

## Class Notes I, Cont'd

(5) <u>While</u> $\quad$ (while p f)

$$(\text{while } p\ f):x \equiv p:x = T \rightarrow (\text{while } p\ f):f:x\ ;$$
$$p:x = F \rightarrow x\ ;\ \perp$$

(6) <u>Power</u> $\quad$ (power f)

$$(\text{power } f):x \equiv x = \langle 0, y\rangle \rightarrow y\ ;$$
$$x = \langle n, y\rangle \wedge \overset{\&\ n \in \text{integer}}{\longrightarrow} (\text{power } f):\langle n-1, f:y\rangle ; \perp$$
$$= f:f:\cdots:f:x$$

(7) <u>Insert</u> $\quad$ (insert f)

$$(\text{insert } f):x \equiv x = \langle x_1\rangle \rightarrow x_1\ ;$$
$$x = \langle x_1, x_2, \ldots, x_n\rangle \rightarrow f:\langle x_1, (\text{insert } f):\langle x_2, \ldots, x_n\rangle\rangle ; \perp$$

(8) <u>Constant</u> $\quad$ (constant y)

$$(\text{constant } y):x \equiv x = \perp \rightarrow \perp\ ;\ y$$

(9) <u>Element-by-element</u> $\quad$ (ee f)

$$(\text{ee } f):x \equiv x = \langle\langle x_1, \ldots, x_n\rangle, \langle y_1, \ldots, y_n\rangle\rangle \rightarrow$$
$$\langle f:\langle x_1, y_1\rangle, \ldots, f:\langle x_n, y_n\rangle\rangle\ ;\ \perp$$

(10) <u>Apply left</u> $\quad$ (apl f)

$$(\text{apl } f):x \equiv x = \langle x_1, \ldots, x_n\rangle \rightarrow \langle f:x_1, \ldots, x_n\rangle\ ;\ \perp$$

(11) <u>Apply right</u> $\quad$ (apr f)

$$(\text{apr } f):x \equiv x = \langle x_1, \ldots, x_n\rangle \rightarrow \langle x_1, \ldots, f:x_n\rangle\ ;\ \perp$$

Feb 1 1975
John Backus

# Class Notes I, cont'd

## Primitive functions

(1) Selectors $1, 2, \ldots$

Not to be confused with the integers 1,2,...!

$1 : x \equiv x = \langle x_1, \ldots, x_n \rangle \rightarrow x_1 ; \bot$

$2 : x \equiv x = \langle x_1, x_2, \ldots, x_n \rangle \;\&\; n \geq 2 \rightarrow x_2 ; \bot$

etc.

(2) Tail $tl$

$tl : x \equiv x = \langle x_1 \rangle \rightarrow \phi ; x = \langle x_1, \ldots, x_n \rangle \;\&\, n \geq 2 \rightarrow \langle x_2, \ldots, x_n \rangle ; \bot$

(3) Equal eq

$eq : x \equiv x = \langle y, y \rangle \rightarrow T ; x = \langle y, z \rangle \rightarrow F ; \bot$

(4) Null null

$null : x \equiv x = \phi \rightarrow T ; x \neq \bot \rightarrow F ; \bot$

(5) And and

$and : x \equiv x = \langle T, T \rangle \rightarrow T ; x = \langle y, z \rangle \;\&\; \{y, z\} \subseteq \{T, F\} \rightarrow F ; \bot$

(6) Not not

$not : x \equiv x = F \rightarrow T ; x = T \rightarrow F ; \bot$

(7) Atom atom

$atom : x \equiv x \in Atom \rightarrow T ; x \in Sequence \rightarrow F ; \bot$

(8) Rotate left rotl

$rotl : x \equiv x = \langle x_1 \rangle \rightarrow \langle x_1 \rangle ; x = \langle x_1, x_2, \ldots x_n \rangle \rightarrow \langle x_2, \ldots, x_n, x_1 \rangle ;$

incomplete definitions

(9) Rotate right $rotr : \langle x_1, \ldots x_n \rangle \Rightarrow \langle x_n, x_1, \ldots, x_{n-1} \rangle$

(10) Distribute from left $distl : \langle x, \langle y_1, \ldots, y_n \rangle \rangle = \langle \langle x, y_1 \rangle, \ldots, \langle x, y_n \rangle \rangle$

(11) Distribute from right $distr : \langle \langle x_1, \ldots, x_n \rangle, y \rangle = \langle \langle x_1, y \rangle, \ldots, \langle x_n, y \rangle \rangle$

(12) Separate sep

$sep : x \equiv x = \langle x_1 \rangle \rightarrow \langle x_1, \phi \rangle ; x = \langle x_1, \ldots, x_n \rangle \rightarrow \langle x_1, \langle x_2, \ldots, x_n \rangle \rangle ;$

(13) Join jn

$jn : x \equiv x = \langle x_1, \phi \rangle \rightarrow \langle x_1 \rangle ; x = \langle x_1, \langle x_2, \ldots, x_n \rangle \rangle \rightarrow \langle x_1, x_2, \ldots, x_n \rangle ; \bot$

Feb 14, 1973
John Backus

## Class Notes I, cont'd

(14) Identity $id$

$$id : x \equiv x$$

(15) Length $ln$

$$ln : x \equiv x = \langle x_1, \ldots, x_n \rangle \rightarrow n;\ x = \emptyset \rightarrow 0;\ \perp$$

(16) Transpose $trans$

$$trans : x \equiv x = \langle x_1, \ldots, x_n \rangle\ \&\ x_i = \langle x_{i1}, \ldots, x_{im} \rangle,\ i = 1, \ldots, n$$
$$\rightarrow \langle y_1, \ldots, y_m \rangle;\ \perp$$
$$\text{where } y_j = \langle x_{1j}, \ldots, x_{nj} \rangle,\ j = 1, \ldots, m$$

(17) Decrease $decrease$

$$\text{decrease} : x \equiv x \in \text{Integer}\ \&\ x > 0 \rightarrow x - 1;\ \perp$$

(18) Increase

$$\text{increase} : x \equiv x \in \text{Integer} \rightarrow x + 1;\ \perp$$

(19) Integer

$$\text{integer} : x \equiv x \in \text{Integer}\ \&\ x \geq 0 \rightarrow T;\ x \neq \perp \rightarrow F;\ \perp$$

(20) Arithmetic functions $+\ -\ \times\ \div$

$$+ : \langle x\ y \rangle = x + y \quad \text{when } x, y \in \text{number}$$

etc

Feb 19, 1975
John Backus

Class Notes, I cont'd

## Definition of new functions in FP

A new function can be defined in FP by writing a definition [of f]:

$$f = \mathcal{F}(f_1, \ldots, f_n)$$

where each $f_i$ is either (a) a primitive or (b) a symbol for which there is another definition or (c) f itself, and $\mathcal{F}$ is either a functional form of FP or a composition of such forms. [E.g., the forms $\alpha f$, $[f_1, f_2]$, and (ee f) can be composed to yeild the form $\alpha[f, (ee\ g)]$, which has the parameters f and g.].

The rule for evaluating an application in which defined functions occur is simply this: whenever it is required to evaluate $f{:}x$ and f is defined as $f = \mathcal{F}(f_1, \ldots, f_n)$, evaluate instead $\mathcal{F}(f_1, \ldots, f_n){:}x$. [We shall not be concerned at present about questions of existence or uniqueness.]

FP has no provision for defining new functional forms.

Feb 19 1975
John Backus

## Examples of defined functions

(1) Definition: $\text{last} = (\text{null}\cdot\text{tl} \rightarrow 1;\ \text{last}\cdot\text{tl})$

A consequence of this definition is the following:

$$(1.1) \qquad \text{last}:x \equiv x = \langle x_1, \ldots, x_n \rangle \rightarrow x_n\ ;\ \perp$$

To prove this, first prove:

$$(1.2) \qquad \text{null}\cdot\text{tl}:x \equiv x = \langle x_1 \rangle \rightarrow T\ ;\ x = \langle x_1, \ldots, x_n \rangle \rightarrow F\ ;\ \perp$$

Then

$\text{last}:x \equiv (\text{null}\cdot\text{tl} \rightarrow 1;\ \text{last}\cdot\text{tl}):x$ by def of last

$\equiv \text{null}\cdot\text{tl}:x = T \rightarrow 1:x\ ;\ \text{null}\cdot\text{tl}:x = F \rightarrow \text{last}\cdot\text{tl}:x\ ;\ \perp$ by def of $(p \rightarrow f;g)$

$\equiv x = \langle x_1 \rangle \rightarrow 1:x\ ;\ x = \langle x_1, \ldots, x_n \rangle \rightarrow \text{last}\cdot\text{tl}:x\ ;\ \perp$ by (1.2)

$\equiv x = \langle x_1 \rangle \rightarrow x_1\ ;\ x = \langle x_1, \ldots, x_n \rangle \rightarrow \text{last}:\langle x_2 \ldots x_n \rangle\ ;\ \perp$ by def of 1 and tl and $\cdot$

Thus, for $n>1$ $\text{last}:\langle x_1, \ldots, x_n \rangle = \text{last}:\langle x_2, \ldots, x_n \rangle = \text{last}:\langle x_n \rangle = x_n$
and for $n=1$ $\text{last}:\langle x_n \rangle = x_n$ and otherwise $\text{last}:x = \perp$

→ ...

Class Notes, I cont'd

Feb 19, 1975
John Backus

## Examples, cont'd

or iota = 2·(power [decrease·1, jn])·[id, [id, (constant ∅)]]

(2) Definition: iota = (power f)∘[id, (constant ∅)]
where
f = jn∘[(constant 1), α increase]

descriptions

$[id, (\text{constant } \emptyset)] : x \equiv \langle x, \emptyset \rangle$

$f : x \equiv x = \emptyset \rightarrow \langle 1 \rangle;\ x = \langle n_1, \ldots, n_m \rangle \ \&\ n_i \in \text{integer} \rightarrow \langle 1, n_1+1, \ldots, n_m+1 \rangle;\ \perp$

$(\text{power } f) : \langle x, \emptyset \rangle \equiv x = 0 \rightarrow \emptyset;\ x \in \text{integer} \rightarrow \langle 1, \ldots, x \rangle;\ \perp$

$\text{iota} : x \equiv x = 0 \rightarrow \emptyset;\ x = n \rightarrow \langle 1, \ldots, n \rangle;\ \perp$

(3) Definition: mm = ααIP ∘ αdistl ∘ distr ∘ (apr trans)
matrix multiply
where
IP = (insert +)∘(ee ×)

descriptions (partial)

inner product: $IP : \langle\langle x_1, \ldots, x_n \rangle, \langle y_1, \ldots, y_n \rangle\rangle = \sum_{i=1}^{n} x_i \times y_i$

$(\text{apr trans}) : \langle m, n \rangle = \langle m, n' \rangle$
where $n' = \text{tranpose} : n$

$\text{distr} : \langle m, n' \rangle = \langle\langle m_1, n' \rangle, \ldots, \langle m_r, n' \rangle\rangle$
where $m = \langle m_1, \ldots, m_r \rangle$

$\alpha\text{distl} : \text{distr} : \langle m, n' \rangle \equiv \langle \text{distl} : \langle m_1, n' \rangle, \ldots, \text{distl} : \langle m_r, n' \rangle \rangle$

for each $\langle m_i, n' \rangle$

$\text{distl} : \langle m_i, n' \rangle = \text{row}_i = \langle\langle m_i, n_1 \rangle, \ldots \langle m_i, n_s \rangle\rangle$
where $n' = \langle n_1, \ldots, n_s \rangle$

Thus $\alpha\text{distl} : \text{distr} : \langle m, n' \rangle = \langle \text{row}_1, \ldots, \text{row}_r \rangle$

→ ...

Feb 19, 1975
John Backus

## Examples, cont'd

now $\alpha\alpha IP : \langle row_1, \ldots, row_n \rangle = \langle \alpha IP{:}row_1, \ldots, \alpha IP{:}row_n \rangle$
and for each $row_i$

$$\alpha IP : row_i = \alpha IP : \langle\langle m_i\, n_1\rangle, \ldots, \langle m_i\, n_s\rangle\rangle$$
$$= \langle IP{:}\langle m_i\, n_1\rangle, \ldots, IP{:}\langle m_i, n_s\rangle\rangle$$

since $n_j$ is the $j^{th}$ column of $n$
and $m_i$ is the $i^{th}$ row of $m$
$\alpha IP : row_i$ is the $i$th row of the product of $m$ and $n$.

MM will multiply any pair of conformable matrices each given as the sequence of its rows.