

Workshop: Programming With Dyalog APL

ECUST, July 5th 2018

Morten Kromberg

(based on work by Jay Foad, Roger Hui & John Scholes)



Agenda

Morning

- Install APL
- Introduction to APL
- Exercises

Afternoon

- A quick look at tools for
 - Importing Data from Excel
 - Charts
 - Review of MatLab code (SA_Cal) converted to APL
- Gitte + Morten depart for airport at 2PM



History of APL



Kenneth E. Iverson
1920-2004

- Canadian of Norwegian Descent
- Born on a small farm in Alberta (Canada)
- Finished one-room school after 9th grade and worked on the farm
- Army 1942, Flight Engineer in Air Force from 1943
 - Almost finished High School in the service
 - Promised his officers and mates that he would pursue an academic career after the war
- B.A. from Queens University, Kingston Ontario
 - Ken didn't know there was such a thing as University before he joined the army!



History of APL, continued



- Doctoral work at Harvard with Aiken and Leontief
- Taught at Harvard for 6 years,
 - frustrated with inadequacies of mathematical notation
- Developed "Iverson Notation" in response
 - Published "A Programming Language" in 1962

ACM Turing award in 1979:

"For his pioneering effort in programming languages and mathematical notation resulting in what the computing field now knows as APL, for his contributions to the implementation of interactive systems, to educational uses of APL, and to programming language theory and practice."

Syntaxes of Mathematics

 ab $Mat1 \cdot Mat2$

Problems:

- *Wide variety of syntactical forms*
- *Strange and inconsistent precedence rules*
- *Things get worse when you deal with matrices*

See <http://www.jsoftware.com/papers/EvalOrder.htm>

$$\sum_{n=1} 4n$$

$$\prod_{i=1} 4i$$

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$



Iverson Notation: Description of IBM\360

IPL	1 : i_{pl}	0
≠	$t, p_{32,33} \leftarrow \bar{e}(18)$	1
	$t_6 \leftarrow \bar{m}_0 \wedge \forall p_{8,9,10,11}$	2
	$MAC^9((\omega^{24}/p), 2, f, i, I^{p_{32,33}})$	3
	$p_{32,33} \leftarrow (2)\tau 1 + \omega^{p_{32,33}}$	4
	$\omega^{24}/p \leftarrow (24)\tau 2 + \omega^{24}/p$	5
	1 : $\forall t_{5,6}$	6
≤	$\omega^{p_{32,33}} : +/\alpha^2/I^0$	7
	$i \leftarrow O_{\omega^{0,1,2,3}}^{I^0}_{\omega^{4,5,6,7}} I^0$	8
	$n \leftarrow N^i$	9
	$t_{1,2} \leftarrow \bar{n}_0, n_1 \wedge p_{15}$	10
	$\rightarrow (12, 14, 13, 17, 19)n_2$	11
RR	$\alpha_2 \leftarrow \omega^4/I^0$	12
RS	$\alpha_{3,j} \leftarrow (\omega^4/I^0), 0$	13
RX	$j \leftarrow (0 \neq \omega^4/I^0) \times \omega^4/I^0$	14
	$\alpha_2 \leftarrow 2^{24} j + (\omega^{12}/I^1) + (0 \neq \omega^4/I^1) \times \omega^4/I^1$	15
	$\alpha_1 \leftarrow \omega^4/\omega^8/I^0$	16
SS	$l \leftarrow (\omega^8/I^0), (\omega^4/\omega^8/I^0), (\omega^4/I^0)$	17

	$\alpha_2 \leftarrow 2^{24} (\omega^{12}/I^2) + (0 \neq \omega^4/I^2) \times \omega^4/I^2$	18
SI	$\alpha_1 \leftarrow 2^{24} (\omega^{12}/I^1) + (0 \neq \omega^4/I^1) \times \omega^4/I^1$	19
	EXC	20
=	1 : $(n_3 = b_0) \wedge \sim \forall t_{3,5,6}$	21
=	0 : $\forall t$	22
	$p_{32,33} \leftarrow ((t_4 \wedge t_5) \vee ?) \wedge p_{32,33}$	23
	$h_1, p_{\omega^{16}(16)} \leftarrow 1, (16)\tau^1/t/\omega^0$	24
	0 : $\forall h \wedge (1, 1, 1, p_7, (\alpha^7/p), \% B_8)$	25
MCIE	$h \leftarrow ((h \wedge (1, 1, 1, p_7, (\alpha^7/p), \% B_8))/\omega^0)_0$	26
	$g_0 \leftarrow (h \cong (0, 3, 4))$	27
=	1 : g_0	28
	$h_h \leftarrow 0$	29
=	1 : $g_1 \wedge (h = 4)$	30
	$MAC^9((48, 40, 32, 24, 56)_h, 8, s, g, p)$	31
	$MAC^9((112, 104, 96, 88, 120)_h, 8, f, g, p)$	32
	$p_{\omega^{16}(18)} \leftarrow ?(18)$	33
	operating state \leftarrow /stop; rate sw = process; operating state/	34
RESET	on: manual light \leftarrow /off; operating state = stop; on/	35
	on: wait light \leftarrow /off; p_{14} ; on/	36

A Programming Language

- The book, 1962

Quotient	$z \leftarrow x \div y$	z is the quotient of x and y
Absolute value	$z \leftarrow x $	$z = x \times [(x > 0) - (x < 0)]$
Floor	$k \leftarrow \lfloor x \rfloor$	$k \leq x < k + 1$
Ceiling	$k \leftarrow \lceil x \rceil$	$k \geq x > k - 1$
j -Residue mod h	$k \leftarrow h _j i$	$i = hq + k; j \leq k < j + h; \text{ and } q \text{ is integral.}$



Linearization => APL\360

- The 5: Ken Iverson, Adin Falkoff, Larry Breed, Dick Lathwell, Roger Moore.
Operated by “Quaker Consensus”.

Quotient	$z \leftarrow x \div y$	z is the quotient of x and y
Absolute value	$z \leftarrow x $	$z = x \times [(x > 0) - (x < 0)]$
Floor	$k \leftarrow \lfloor x \rfloor$	$k \leq x < k + 1$
Ceiling	$k \leftarrow \lceil x \rceil$	$k \geq x > k - 1$
j -Residue mod h	$k \leftarrow h j i$	$i = hq + k; j \leq k < j + h$; and q is integral.



$a \times b$ $a b$ $Mat1 \cdot Mat2$ $Mat1 + \cdot \times Mat2$ $* x$ $f g x$ e^x $x \div y$ $\frac{x}{y}$ $(f + g) x$ $f g x^{f+g} x$ $(3 \circ x) * 2$ $\log_b a$ $\sqrt[n]{a}$ $\tan^{2/4 \times 16} x$ $\times / 4 \times 16$ $a * \div n$ $\sum_{n=1}^6 4n$ $\prod_{i=1}^6 4i$ $\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$ $(2 \times a) \div \sim (-b) (+, -) 0.5 * \sim (b * 2) - 4 \times a \times c$ 

APL Was Born...

- The first running system started running in November of 1966.
- Let's do some exercises.



Exercises

- The folder "Exercises" contains numbered files, starting with "01-syntax.txt"
- Enter each expression (line)



Saving Your Work

- Historically, APL users have saved workspaces containing code and data as a single file
 - Similar to an Excel Spreadsheet
 - Takes a "snapshot of the VM"
 - Beware: also saves the execution stack if there is one
- Save your work using

```
)save /path/mywsname[.dws]
```
- Load it again with

```
)load /path/mywsname
```
- You can extract named objects from a workspace:

```
)copy /path/mywsname foo goo x y
```
- Saved workspaces can have a "latent expression" $\square L X$, which is executed when the workspace is loaded, unless you

```
)xload /path/mywsname
```



Saving Your Work, continued

- In the last few years, it has become more popular to use Unicode text files (and SVN/GIT), especially for source code
- You can save a fn/var, namespace or class using `]save` (`]save` is a "user command" - written in APL):
 `]save name /path/name[.dyalog]`
(it is customary to use the same name for the file)
- You can bring it back into the active workspace using
 `]load /path/name`
 `]load /path/*`
- If you edit objects that were `]loaded`, the system will offer to update the file each time you make a change
- From version 15.0, the interpreter (editor) knows how to open and view source files without user commands.



Reading APL

$$CB \leftarrow \{ \omega [1 + (\rho \omega) \mid X \circ . + X \leftarrow (\iota \alpha) - 1] \}$$

Imagine arguments:

8 CB ' '



Reading APL

What does this function do?

$$\{(\sim R \in R \circ . \times R) / R \leftarrow 1 \downarrow \iota \omega\}$$

Or:

$$\{\{(\sim \omega \in \omega \circ . \times \omega) / \omega\} 1 \downarrow \iota \omega\}$$



Procedures / Tradfns

Monadic:

```
    ▽ R←Sum X  
[1]    R←+/X  
    ▽
```

Dyadic:

```
    ▽ R←A MatMult B  
[1]    R←A+.×B  
    ▽
```

Niladic:

```
    ▽ Run  
[1]    □←'Boo Hiss!'  
    ▽
```



Procedures / Tradfns

"Ambi-valent" (+ use a control structure)

```
    ▽ R←{Window} Sum X  
[1]    :If 0=□NC 'Window' ♦ R←+/X  
[2]    :Else ♦ R←Window +/ X  
[3]    :EndIf  
    ▽
```



Procedures / Tradfns

Name Elements of Right Argument

+ Local Variable

+ Class / DotNet declarations

▽ r←Round(n decimals);base

[1] :Access Public Shared

[2] :Signature Double←Round Double N, Int32 Decimals

[3] base←10×decimals

[4] r←(⌊0.5+n×base)÷base

▽



Errors

```
1 2 3÷4 5
LENGTH ERROR
```

```
1 2 3÷4 5
      ^
```

```
□EN
```

```
5
```

```
1÷0
```

```
DOMAIN ERROR: Divide by zero
```

```
1÷0
```

```
      ^
```

```
□EN (□EM 11)
```

```
11 DOMAIN ERROR
```

```
□DMX.Message
```

```
Divide by zero
```



Errors

1	WS FULL	18	FILE TIE ERROR
2	SYNTAX ERROR	19	FILE ACCESS ERROR
3	INDEX ERROR	20	FILE INDEX ERROR
4	RANK ERROR	21	FILE FULL
5	LENGTH ERROR	22	FILE NAME ERROR
6	VALUE ERROR	23	FILE DAMAGED
7	FORMAT ERROR	24	FILE TIED
10	LIMIT ERROR	25	FILE TIED REMOTELY
11	DOMAIN ERROR	26	FILE SYSTEM ERROR
12	HOLD ERROR	28	FILE SYSTEM NOT AVAILABLE
13	OPTION ERROR	30	FILE SYSTEM TIES USED UP
15	LST FULL	31	FILE TIE QUOTA USED UP
16	NONCE ERROR	32	FILE NAME QUOTA USED UP
17	ACCESS ERROR	34	FILE SYSTEM NO SPACE
		35	FILE ACCESS ERROR - CONVERTING FILE
		36	INCOMPATIBLE ARRAY
		38	FILE COMPONENT DAMAGED



Error Trapping: Dfns

```
div←{0::'Something Else is Wrong'  
    11::0 A DOMAIN error: return 0  
    α÷ω}  
    3 div 0  
0  
    1 2 3 div 4 5  
Something Else is Wrong
```



Error Trapping: Tradfns

Using :Trap

```
▽ R←A DIV B
[1]   :Trap 0
[2]   R←A÷B
[3]   :Case 11 ♦ R←0 A DOMAIN error
[4]   :Else ♦ R←'Something Else is Wrong'
[5]   :EndTrap
▽
```



Error Trapping: Tradfns

Using □TRAP

```
▽ R←A DIVQ B;□TRAP
[1]
[2]   □TRAP←(11 'E' '→DOMERR')(0 'C' '→CATCHALL')
[3]   R←A÷B ◇ →0
[4]
[5]   DOMERR:→R←0
[6]   CATCHALL:R←'Something Else is Wrong'
▽
```

