

CS 421 Lecture 15

- ▶ Today's class: APL
 - ▶ Functional programming – “no side effects”

Functional Programming

$$if (f() + 1 > f()) \dots$$
$$e + e \neq e$$

- ▶ “The assignment statement splits programming into two worlds. The first world comprises the right sides of assignment statements. This is an orderly world of expressions, a world that has useful algebraic properties.... It is the world in which most useful computation takes place.
- ▶ “The second world... is the world of statements. ... This world of statements is a disorderly one, with few useful mathematical properties.”

John Backus (creator of Fortran), “Can Programming be liberated from the von Neumann Style? A Functional Style and its Algebra of Programs.” Turing Award lecture, 1977.

$$e_1 + e_2 = e_2 + e_1$$

APL

- ▶ Computations on matrices using operators that have matrix arguments.
- ▶ Ken Iverson – “A Programming Language” – 1960
- ▶ Defined a set of operators on matrices, plus a typeface for those operators, and built terminals

APL operations

- ▶ Binary operations on numbers extended naturally to matrices
 - Comparison and boolean ops treated as arithmetic
- ▶ Reduction operations: $+ /$, $\times /$, $\wedge /$, ...
 - For vectors, put operator between every element
 - For matrices, reduce each row
- ▶ Compression $B \ / \ V$
 - selects elements (or rows) of V where $B = 1$

boolean (0's & 1's)

No precedence rules, evaluate right-to-left

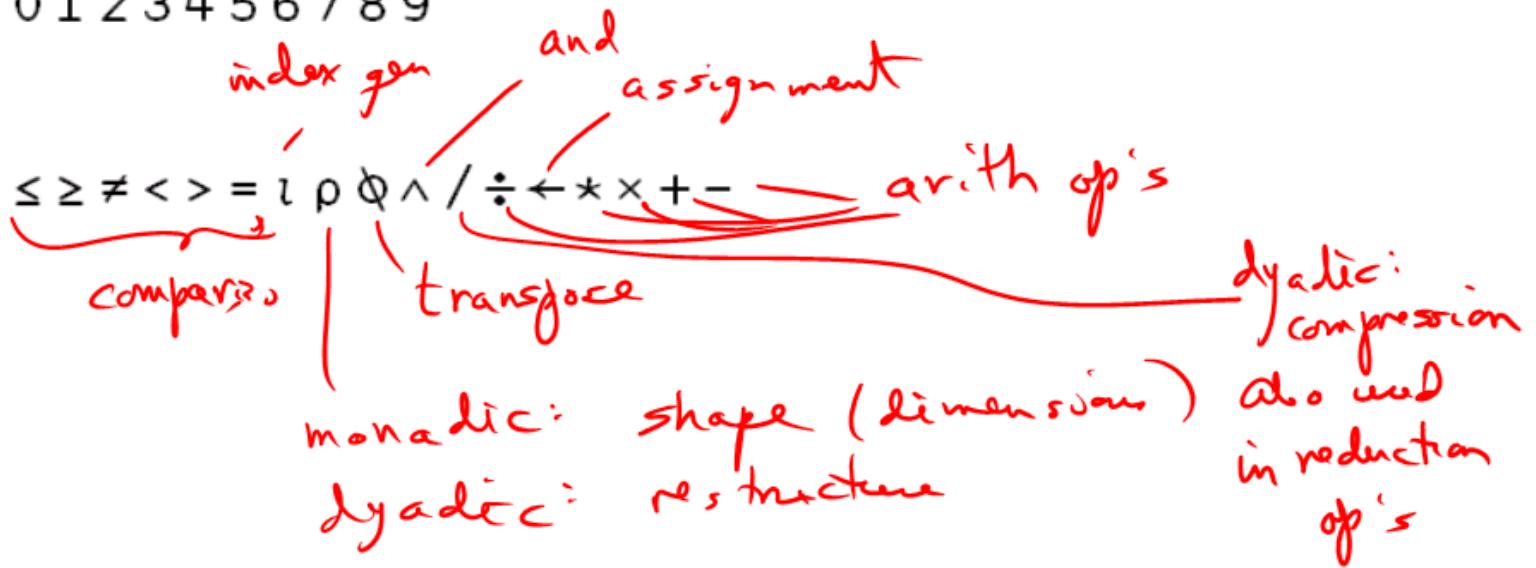
APL font

Monadic - unary

Dyadic - binary

ABCDEFGHIJKLMNOPQRSTUVWXYZ

0123456789



APL examples

▶ $1+M$ — $\begin{pmatrix} 2 & 3 & 4 \\ 5 & 6 & 7 \end{pmatrix}$

▶ $(+/V) \div n$ If $n=4$, gives average — 5

▶ $(+/V) \div pV$ — Average of V

$(+/M) \div pM$

▶ $((V \div 2) \times 2) = V$ / V

$$\begin{array}{r} 6 \ 15 \div 2 \ 3 \\ = 3 \ 5 \end{array}$$

would not work if M had > 2 rows

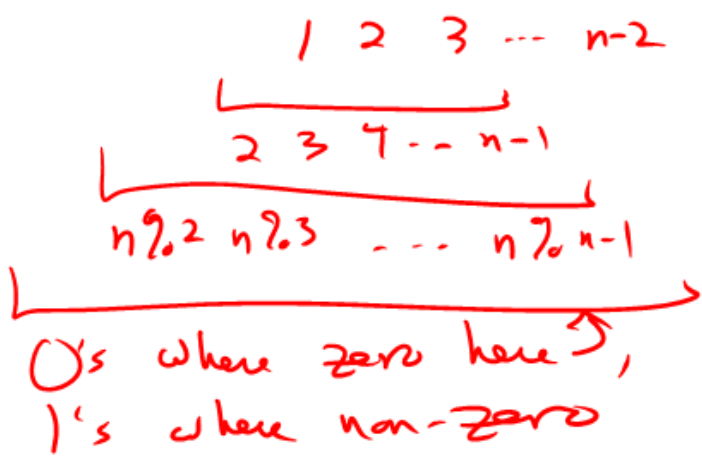
$V \leftarrow 2 \ 4 \ 6 \ 8$

$M \leftarrow 2 \ 3 \ 4 \ 5 \ 6 = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$

APL examples

$n * .5$
x

▶ `prime n = ^/(0≠n%(1+i(n-2)))`

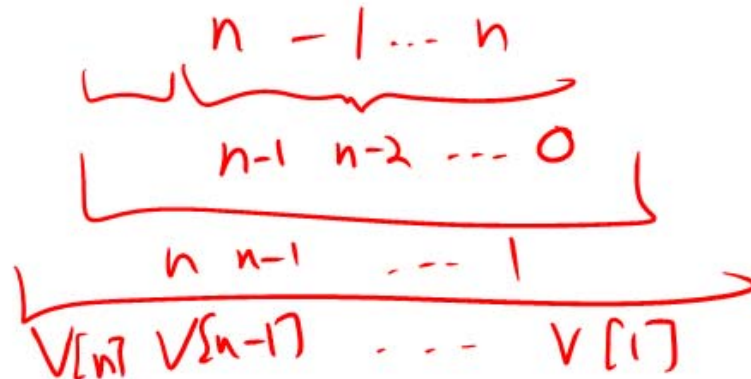


APL examples

- ▶ Subscripting: $V[V^T]$ – elements of V in positions given by V^T .

reverse $V = V[1+(pV)-:pV]$

Suppose $pV = n$



$V = 2 \ 4 \ 6 \ 8$

$V[1 \ 2 \ 1 \ 2] = 2 \ 4 \ 2 \ 4$

APL examples

- ▶ Dyadic ρ – “restructure”
- ▶ $V \rho A$ returns a value with shape V , values drawn from A

$$2 \ 3 \rho \ 2 \ 6 = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$$

$$2 \ 3 \rho \ 2 \ 5 = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 1 \end{pmatrix}$$

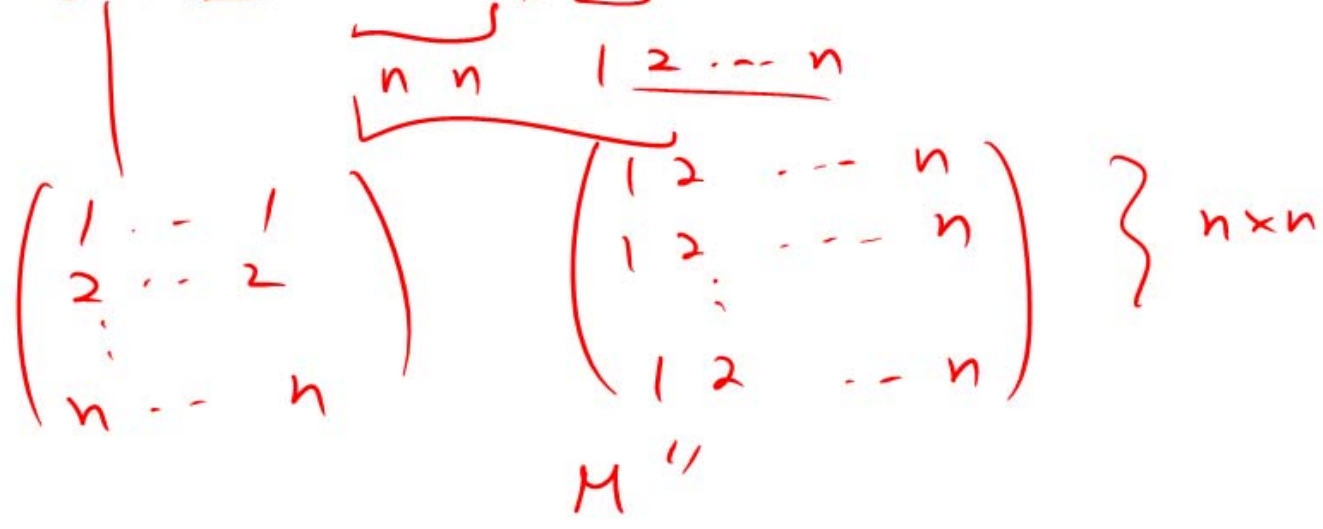
$$\underbrace{(2 \rho n)}_{n \ n} \underbrace{\rho}_{\text{vector of } n \ 0's} 1, \underbrace{n \rho 0}_{\text{vector of } n \ 0's} = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ & & & & 1 \end{pmatrix}$$

APL examples

- ▶ ← assignment
- ▶ ⍉ transpose

quote quad
⍉ ←

$$I \leftarrow (\ominus M) \boxed{=} M \leftarrow (2 \rho n) \boxed{\rho} \underbrace{1 \ 2 \ \dots \ n}$$



APL examples

let zero = newint 0;;

let four = newint 4;;

let a = rho(newveci [2;3]) (indx (newint 6));;

2 3 4 5 6

let v = newveci [2;4;6];;

let c = newveci [1;0];;

let d = newveci [1;0;1];;

a *@ a

$a \times a$

v -@ one

$v - 1$

a >@ four

$a > 4$

} arith. op's

!+v

+ - reduction

APL examples

maxR a

d % v

c % a

shape a

ravel a

rho (shape a) v

rho (shape v) c

a ^ @ c

trans

@

shar

, minR - max reduction
min reduction
} compression

- turns matrix or scalar into vector

- concatenate

- transpose
subscript

APL examples

indx (newint 5)

$\uparrow 5$

trans a

ϕa

v @@ (indx two)

$v [2 2]$

a @@ one

$a [1]$

(trans a) @@ (indx two)

$(\phi a) [2 2]$

APL examples

let incr a = a +@ (newint 1);;

let fac n = !* (indx n);;

let avg v = (!+v) /@ (shape v);;

let reverse v =

let sz = (shape v) @@ one

in v @@ (incr (sz -@ (indx sz)));;

let prime n = !& (zero <@ (n %@@ (incr

(indx (n -@ two)))));;

- Same as prime in APL

APL

a + 1

! * n

(! + v) / p v

let sz = (p v)[1]
in v [1 + sz - ?sz]

