

Postscript

CS4700

Kenneth Sundberg

Postscript

- Adobe product
- First released in 1984
- Combined text and raster graphics in one format
- Its successor PDF has replaced much of its functionality

Early printing

- Early computers were very resource constrained
- Printers had a much higher resolution than displays or memory could handle
- Postscript allowed offloading work to the printer
- When printers were costly the addition of a postscript processor was marginal

Display

- Was tried as a basis for a display engine
- Printing lacks interactivity so extensions were needed
- 'Standardization' of X11 killed project

Language Properties

- Turing complete
- Concatenative language
- Enforces a 'points-free' style
- Stack based with postfix notation

Stack

- Central to Postscript is the stack
- There are three stacks
 - Value
 - Dictionary
 - Function (implicit)

Literals

- Literals have the effect of placing their bound values on the stack
- The units on integers are points
- There are 72 points in an inch

Operators

- Operators (functions) take some number of values from the stack (possibly zero) and put their results onto the stack

Manipulation

- clear
- dup
- exch
 - Very useful when dealing with function parameters
 - `a b exch -> b a`
- pop
- roll
 - `a b c 3 1 roll -> c a b`

Operators

- add
- sub
- div
- idiv - integer division
- mod
- mul
- neg
- sin - angle in degrees
- cos - angle in degrees

Length example

- Write post script for distance between points
- The coordinates are (x_1, y_1) and (x_2, y_2)

Drawing

- Drawing happens in two steps
 - Create a path
 - Stroke or fill that path
- All colors are completely opaque.
- showpage

Path

- newpath
- moveto
- lineto
- rmoveto
- rlineto
- There are also a number of curve operations

Text

- findfont
- scalefont
- setfont
- show
 - draws a string of text
 - strings are denoted with ()

Coordinates

- translate
- rotate
- scale
- gsave
- grestore

Stroke

- setlinewidth
- stroke
- fill
- setrgbcolor
- setgray

Dictionaries

- Postscript provides a dictionary (map) type
- There is a stack of dictionaries
 - The bottom is a system dictionary
 - The next is the user dictionary
- Variables are looked up in this stack

Dynamic Scoping

- The visibility of a definition depends on the order of execution

Adding variables

- def

Executable Arrays

- A list of literals in braces will not be executed but rather compiled into an executable array
- These are used like function objects for control structures

Functions

- If you define a name for an executable array (using `def`) then you can invoke that array via the name

Relation

- eq
- ne
- gt
- ge
- lt
- le

Conditions

- if
- ifelse

Loops

- for
 - $j\ k\ l\ \text{proc} \rightarrow \text{look } i=j \text{ to } k \text{ step } l.$
 - i is on top of stack
- loop
- exit

Textbook sections covered:

- Section 05-05 (frame 18)