Stacks and Queues

Hsuan-Tien Lin

Dept. of CSIE, NTU

March 21–22, 2011
System Stack

- recall: function call \(\Leftrightarrow\) 拿新的草稿纸來算
- old (original) scrap paper: temporarily not used, 可以壓在下面

System Stack: 一叠草稿纸，each paper (stack frame) contains

- return address: where to return to the previous scrap paper
- local variables (including parameters): to be used for calculating within this function
- previous frame pointer: to be used when escaping from this function

some related issues: stack overflow? security attack?
Reading Assignment

be sure to go ask the TAs or me if you are still confused
when stack full, grow array by size $M$

- successful (direct) growth: constant time
- if unlucky, growth by copying: $O(\text{capacity})$

$M = 1$ or any constant: very conservative
  — worst case, $O(n^2)$ for $n$ pushes (why?)

$M = \text{capacity}$:
  — growth when exceeding 1, 2, 4, 8, 16, ...
  — each growth takes time around 1, 2, 4, 8, 16, ...
  — when $n$ pushes with $n = 13$?

\[
1 + 2 + 4 + 8 + \cdots + 2^{13-1} = \mathcal{O}(2^n - 1) = \mathcal{O}(n)
\]

— $2^k < n \leq 2^{k+1}$, time $2^{k+1} - 1$ on growth and $n$ on pushes
— $\mathcal{O}(n)$ for $n$ pushes
Stack for Expression Evaluation (Sec. 3.6)

\[ \frac{a}{b} - c + d \cdot e - a \cdot c \]

- precedence: \{\ast, \div\} first; \{+, -, \} later
- steps
  - \( f = \frac{a}{b} \)
  - \( g = f - c \)
  - \( h = d \cdot e \)
  - \( i = g + h \)
  - \( j = a \cdot c \)
  - \( \ell = i - j \)

\[ \frac{ab}{fc} - \Rightarrow \frac{ab}{c} - \]
\[ d \cdot e * \]
\[ gh + \Rightarrow ab/c - de* + \]
\[ ac* \]
\[ ij - \Rightarrow ab/c - de* + ac* - \]

Postfix Notation

same operand order, but put “operator” after needed operands
—can “operate” immediately when seeing operator
—no need to look beyond for precedence