Stacks and Queues

Hsuan-Tien Lin

Dept. of CSIE, NTU

March 21–22, 2011

System Stack

- recall: function call ⇔ 拿新的草稿紙來算
- old (original) scrap paper: temporarily not used, 可以壓在下面

System Stack: 一疊草稿紙, 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?

Stacks with Fixed C Array (Part of Sec. 3.1)

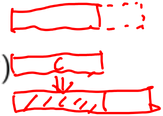
Reading Assignment

be sure to go ask the TAs or me if you are still confused

Stacks with Dynamically Growing Array (Sec. 3.2)

when stack full, grow array by size M

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



- $M = 1$ or any constant: very conservative
  —worst case, $O(n^2)$ for n pushes (why?)
- $M = capacity$:
  —growth when exceeding 1, 2, 4, 8, 16, ...
  —each growth takes time around 1, 2, 4, 8, 16, ...
  —when n pushes with $n = 13$?

$1 + 2 + 4 + 8 = 15$

$= O(2n - 1) = O(n)$

—$2^k < n \leq 2^{k+1}$, time $2^{k+1} - 1$ on growth and n on pushes
—$O(n)$ for n pushes

Stack for Expression Evaluation (Sec. 3.6)

$$a/b - c + d * e - a * c$$

- precedence: $\{*, /\}$ first; $\{+, -\}$ later
- steps
  - $f = a/b$ ab/
  - $g = f - c$ fc− ⇒ ab/c−
  - $h = d * e$ de*
  - $i = g + h$ gh+ ⇒ ab/c−de*+
  - $j = a * c$ ac*
  - $\ell = i - j$ ij− ⇒ 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