Stack Applications

Lecture 25
Section 7.4 - 7.5

Robb T. Koether

Hampden-Sydney College

Mon, Mar 23, 2009
Outline

1. Postfix Expressions
2. Function Calls
3. The Triangle Puzzle
4. Assignment
An infix expression is an arithmetic expression in which the binary operators are written in between the operands.

For example, to add 3 and 4, we write

\[ 3 + 4. \]
A postfix expression with one (binary) operator is written in the order: left-operand, right-operand, operator.

For example, to add 3 and 4, we write

\[ 3 4 + . \]
A prefix expression with one binary operator is written in the order:

operator, left-operand, right-operand.

For example, to add 3 and 4, we write $+ 3 4$. 
Example (Postfix Expressions)

- Expression: 3 4 + 5 6 + *.
- Left operand of * is 3 4 +.
- Right operand of * is 5 6 +.

In postfix expressions, parentheses are never needed!
Postfix Expression Evaluation

Example (Postfix Expressions)

- Rewrite the infix expression

\[ (((1 + 2) \times 3 - 4) / (5 + 6 - 7 \times 8)) \times 9 \]

as a postfix expression.
Example (Postfix Expressions)

- Rewrite the infix expression

\[((1 + 2) \times 3 - 4/(5 + 6 - 7 \times 8)) \times 9\]

as a postfix expression.

1 2 +
Example (Postfix Expressions)

- Rewrite the infix expression

\[((1 \, + \, 2) \, \times \, 3 \, - \, 4/((5 \, + \, 6 \, - \, 7 \, \times \, 8)) \, \times \, 9\]

as a postfix expression.

\[1 \, 2 \, + \, 3 \, *\]
Example (Postfix Expressions)

Rewrite the infix expression

\[((1 + 2) \ast 3 - 4)/(5 + 6 - 7 \ast 8))\ast 9\]

as a postfix expression.

1 2 + 3 \ast 4
Example (Postfix Expressions)

- Rewrite the infix expression

\[((1 + 2) \times 3 - 4/(5 + 6 - 7 \times 8)) \times 9\]

as a postfix expression.

1 2 + 3 * 4 5 6 +
Example (Postfix Expressions)

- Rewrite the infix expression

\[((1 + 2) \times 3 - 4/(5 + 6 - 7 \times 8)) \times 9\]

as a postfix expression.

1 2 + 3 * 4 5 6 + 7 8 *
Example (Postfix Expressions)

- Rewrite the infix expression

\[((1 + 2) \times 3 - 4/(5 + 6 - 7 \times 8)) \times 9\]

as a postfix expression.

1 2 + 3 * 4 5 6 + 7 8 * -
Example (Postfix Expressions)

- Rewrite the infix expression

\[ ((1 + 2) \times 3 - \frac{4}{(5 + 6 - 7 \times 8)}) \times 9 \]

as a postfix expression.

\[ 1\ 2\ +\ 3\ \times\ 4\ 5\ 6\ +\ 7\ 8\ \times\ -\ / \]
Example (Postfix Expressions)

- Rewrite the infix expression

\[
((1 + 2) \times 3 - 4/(5 + 6 - 7 \times 8)) \times 9
\]

as a postfix expression.

\[
1 2 + 3 \times 4 5 6 + 7 8 \times - / -
\]
Example (Postfix Expressions)

- Rewrite the infix expression

$(((1 + 2) \times 3 - 4/(5 + 6 - 7 \times 8)) \times 9)$

as a postfix expression.

$1\ 2\ +\ 3\ \times\ 4\ 5\ 6\ +\ 7\ 8\ \times\ -\ /\ -\ 9\ \times$
To evaluate a postfix expression:

- Begin with an empty stack.
- Each number or operator is a token.
- Process the tokens in the postfix expression from left to right.
Postfix Expression Evaluation

- For each token,
  - If the token is a number,
    - Push it onto the stack.
  - If the token is a binary operator,
    - Pop two numbers off the stack.
    - Combine them under the operator.
    - Push the result onto the stack.
- The single remaining value on the stack is the value of the expression.
Example (Postfix Evaluation)

Evaluate $1 \ 2 \ + \ 3 \ * \ 4 \ 5 \ 6 \ + \ 7 \ 8 \ * \ - \ / \ - \ 9 \ *$.

- Read 1: Push 1.
- Read 2: Push 2.
- Read +: Pop 2; Pop 1; Compute $1 + 2 = 3$; Push 3.
- Read 3: Push 3.
- Read *: Pop 3; Pop 3; Compute $3 \ * \ 3 = 9$; Push 9.
- Read 4: Push 4.
- Read 5: Push 5.
- Read +: Pop 6; Pop 5; Compute $5 + 6 = 11$; Push 11.
- Read 7: Push 7.
- Read 8: Push 8.
- Read *: Pop 8; Pop 7; Compute $7 \ * \ 8 = 56$; Push 56.
Example (Postfix Evaluation)

- **Evaluate** $1 \ 2 \ + \ 3 \ * \ 4 \ 5 \ 6 \ + \ 7 \ 8 \ * \ - \ / \ - \ 9 \ *$.
- **Read 1**: Push 1.
  - **Read 2**: Push 2.
  - **Read +**: Pop 2; Pop 1; Compute $1 + 2 = 3$; Push 3.
  - **Read 3**: Push 3.
  - **Read ***: Pop 3; Pop 3; Compute $3 \times 3 = 9$; Push 9.
  - **Read 4**: Push 4.
  - **Read 5**: Push 5.
  - **Read 6**: Push 6.
  - **Read +**: Pop 6; Pop 5; Compute $5 + 6 = 11$; Push 11.
  - **Read 7**: Push 7.
  - **Read 8**: Push 8.
  - **Read ***: Pop 8; Pop 7; Compute $7 \times 8 = 56$; Push 56.
Postfix Expression Evaluation

**Example (Postfix Evaluation)**

Evaluate \(1 2 + 3 * 4 5 6 + 7 8 * - / - 9 *\).

- **Read 1**: Push 1.
- **Read 2**: Push 2.
  - **Read +**: Pop 2; Pop 1; Compute \(1 + 2 = 3\); Push 3.
  - **Read 3**: Push 3.
  - **Read ***: Pop 3; Pop 3; Compute \(3 * 3 = 9\); Push 9.
- **Read 4**: Push 4.
- **Read 5**: Push 5.
- **Read 6**: Push 6.
- **Read +**: Pop 6; Pop 5; Compute \(5 + 6 = 11\); Push 11.
- **Read 7**: Push 7.
- **Read 8**: Push 8.
- **Read ***: Pop 8; Pop 7; Compute \(7 * 8 = 56\); Push 56.
Example (Postfix Evaluation)

Evaluate $1 \ 2 \ + \ 3 \ * \ 4 \ 5 \ 6 \ + \ 7 \ 8 \ * \ \div \ - \ - \ 9 \ *$.

- Read 1: Push 1.
- Read 2: Push 2.
- Read $+$: Pop 2; Pop 1; Compute $1 + 2 = 3$; Push 3.
- Read 3: Push 3.
- Read $*$: Pop 3; Pop 3; Compute $3 \times 3 = 9$; Push 9.
- Read 4: Push 4.
- Read 5: Push 5.
- Read $+$: Pop 6; Pop 5; Compute $5 + 6 = 11$; Push 11.
- Read 7: Push 7.
- Read 8: Push 8.
- Read $*$: Pop 8; Pop 7; Compute $7 \times 8 = 56$; Push 56.
Postfix Expression Evaluation

Example (Postfix Evaluation)

Evaluate $1\ 2\ +\ 3\ \times\ 4\ 5\ 6\ +\ 7\ 8\ \times\ -\ /\ -\ 9\ \times$.

- Read 1: Push 1.
- Read 2: Push 2.
- Read +: Pop 2; Pop 1; Compute $1 + 2 = 3$; Push 3.
- Read 3: Push 3.
- Read \times: Pop 3; Pop 3; Compute $3 \times 3 = 9$; Push 9.
- Read 4: Push 4.
- Read 5: Push 5.
- Read +: Pop 6; Pop 5; Compute $5 + 6 = 11$; Push 11.
- Read 7: Push 7.
- Read 8: Push 8.
- Read \times: Pop 8; Pop 7; Compute $7 \times 8 = 56$; Push 56.
Example (Postfix Evaluation)

- **Evaluate** $1 \ 2 \ + \ 3 \ * \ 4 \ 5 \ 6 \ + \ 7 \ 8 \ * \ - \ / \ - \ 9 \ *$.
  - Read 1: Push 1.
  - Read 2: Push 2.
  - Read +: Pop 2; Pop 1; Compute $1 + 2 = 3$; Push 3.
  - Read 3: Push 3.
  - Read *: Pop 3; Pop 3; Compute $3 \times 3 = 9$; Push 9.
  - Read 4: Push 4.
  - Read 5: Push 5.
  - Read +: Pop 6; Pop 5; Compute $5 + 6 = 11$; Push 11.
  - Read 7: Push 7.
  - Read 8: Push 8.
  - Read *: Pop 8; Pop 7; Compute $7 \times 8 = 56$; Push 56.
Example (Postfix Evaluation)

Evaluate $1 \ 2 \ + \ 3 \ * \ 4 \ 5 \ 6 \ + \ 7 \ 8 \ * \ - \ / \ - \ 9 \ *$.

- Read 1: Push 1.
- Read 2: Push 2.
- Read +: Pop 2; Pop 1; Compute $1 + 2 = 3$; Push 3.
- Read 3: Push 3.
- Read *: Pop 3; Pop 3; Compute $3 * 3 = 9$; Push 9.
- Read 4: Push 4.
- Read 5: Push 5.
- Read +: Pop 6; Pop 5; Compute $5 + 6 = 11$; Push 11.
- Read 7: Push 7.
- Read 8: Push 8.
- Read *: Pop 8; Pop 7; Compute $7 * 8 = 56$; Push 56.
## Postfix Expression Evaluation

### Example (Postfix Evaluation)

Evaluate $1 \ 2 \ + \ 3 \ * \ 4 \ 5 \ 6 \ + \ 7 \ 8 \ * \ - \ / \ - \ 9 \ *$.

1. **Read 1**: Push 1.
2. **Read 2**: Push 2.
3. **Read +**: Pop 2; Pop 1; Compute $1 + 2 = 3$; Push 3.
4. **Read 3**: Push 3.
5. **Read ***: Pop 3; Pop 3; Compute $3 * 3 = 9$; Push 9.
7. **Read 5**: Push 5.
9. **Read +**: Pop 6; Pop 5; Compute $5 + 6 = 11$; Push 11.
11. **Read 8**: Push 8.
12. **Read ***: Pop 8; Pop 7; Compute $7 * 8 = 56$; Push 56.
Postfix Expression Evaluation

Example (Postfix Evaluation)

Evaluate $1 \ 2 \ + \ 3 \ * \ 4 \ 5 \ 6 \ + \ 7 \ 8 \ * \ - \ / \ - \ 9 \ *$.

- Read 1: Push 1.
- Read 2: Push 2.
- Read +: Pop 2; Pop 1; Compute $1 + 2 = 3$; Push 3.
- Read 3: Push 3.
- Read *: Pop 3; Pop 3; Compute $3 \ * \ 3 = 9$; Push 9.
- Read 4: Push 4.
- Read 5: Push 5.
- Read +: Pop 6; Pop 5; Compute $5 + 6 = 11$; Push 11.
- Read 7: Push 7.
- Read 8: Push 8.
- Read *: Pop 8; Pop 7; Compute $7 \ * \ 8 = 56$; Push 56.
Example (Postfix Evaluation)

Evaluate $1 2 + 3 * 4 5 6 + 7 8 * - / - 9 *$.

- Read 1: Push 1.
- Read 2: Push 2.
- Read +: Pop 2; Pop 1; Compute $1 + 2 = 3$; Push 3.
- Read 3: Push 3.
- Read *: Pop 3; Pop 3; Compute $3 * 3 = 9$; Push 9.
- Read 4: Push 4.
- Read 5: Push 5.
- Read +: Pop 6; Pop 5; Compute $5 + 6 = 11$; Push 11.
- Read 7: Push 7.
- Read 8: Push 8.
- Read *: Pop 8; Pop 7; Compute $7 * 8 = 56$; Push 56.
Postfix Expression Evaluation

Example (Postfix Evaluation)

- **Evaluate** \(1 \ 2 \ + \ 3 \ * \ 4 \ 5 \ 6 \ + \ 7 \ 8 \ * \ − \ / \ − \ 9 \ *\).
  - Read 1: Push 1.
  - Read 2: Push 2.
  - Read +: Pop 2; Pop 1; Compute \(1 + 2 = 3\); Push 3.
  - Read 3: Push 3.
  - Read *: Pop 3; Pop 3; Compute \(3 \times 3 = 9\); Push 9.
  - Read 4: Push 4.
  - Read 5: Push 5.
  - Read +: Pop 6; Pop 5; Compute \(5 + 6 = 11\); Push 11.
  - Read 7: Push 7.
  - Read 8: Push 8.
  - Read *: Pop 8; Pop 7; Compute \(7 \times 8 = 56\); Push 56.
Postfix Expression Evaluation

Example (Postfix Evaluation)

- **Evaluate** $1 \ 2 \ + \ 3 \ * \ 4 \ 5 \ 6 \ + \ 7 \ 8 \ * \ - \ / \ - \ 9 \ *$.
  - Read 1: Push 1.
  - Read 2: Push 2.
  - Read +: Pop 2; Pop 1; Compute $1 + 2 = 3$; Push 3.
  - Read 3: Push 3.
  - Read *: Pop 3; Pop 3; Compute $3 * 3 = 9$; Push 9.
  - Read 4: Push 4.
  - Read 5: Push 5.
  - Read +: Pop 6; Pop 5; Compute $5 + 6 = 11$; Push 11.
  - Read 7: Push 7.
  - Read 8: Push 8.
  - Read *: Pop 8; Pop 7; Compute $7 * 8 = 56$; Push 56.
Example (Postfix Evaluation)

Evaluate $1\ 2\ +\ 3\ \ast\ 4\ 5\ 6\ +\ 7\ 8\ \ast\ -\ /\ -\ 9\ \ast$.

- Read 1: Push 1.
- Read 2: Push 2.
- Read $+$: Pop 2; Pop 1; Compute $1 + 2 = 3$; Push 3.
- Read 3: Push 3.
- Read $\ast$: Pop 3; Pop 3; Compute $3 \ast 3 = 9$; Push 9.
- Read 4: Push 4.
- Read 5: Push 5.
- Read $+$: Pop 6; Pop 5; Compute $5 + 6 = 11$; Push 11.
- Read 7: Push 7.
- Read 8: Push 8.
- Read $\ast$: Pop 8; Pop 7; Compute $7 \ast 8 = 56$; Push 56.
Example (Postfix Evaluation)

Evaluate $1 \ 2 \ + \ 3 \ \ast \ 4 \ 5 \ 6 \ + \ 7 \ 8 \ \ast \ - \ / \ - \ 9 \ \ast$.

- Read $-$: Pop 56; Pop 11; Compute $11 - 56 = -45$; Push $-45$.
- Read $/$: Pop $-45$; Pop 4; Compute $4 / (-45) = -0.0888 \ldots$; Push $-0.0888 \ldots$.
- Read $-$: Pop $-0.0888 \ldots$; Pop 9; Compute $9 - (-0.0888 \ldots) = 9.0888 \ldots$; Push 9.0888 \ldots$.
- Read $\ast$: Pop 9; Pop 9.0888 \ldots; Compute $(9.0888 \ldots) \ast 9 = 81.8$; Push 81.8.
Example (Postfix Evaluation)

- **Evaluate** $1 \ 2 \ \ + \ 3 \ \ * \ 4 \ 5 \ 6 \ \ + \ 7 \ 8 \ \ * \ - \ / \ - \ 9 \ *$.
- **Read** $-$: Pop 56; Pop 11; Compute $11 - 56 = -45$; Push $-45$.
- **Read** /: Pop $-45$; Pop 4; Compute $4 / (-45) = -0.0888\ldots$; Push $-0.0888\ldots$.
- **Read** $-$: Pop $-0.0888\ldots$; Pop 9; Compute $9 - (-0.0888\ldots) = 9.0888\ldots$; Push 9.0888\ldots.
- **Read** *: Pop 9; Pop 9.0888\ldots; Compute $(9.0888\ldots) \* 9 = 81.8$; Push 81.8.
Example (Postfix Evaluation)

- **Evaluate** $1 \ 2 \ + \ 3 \ \ast \ 4 \ 5 \ 6 \ + \ 7 \ 8 \ \ast \ - \ / \ - \ 9 \ \ast$.
  - Read $-$: Pop 56; Pop 11; Compute $11 - 56 = -45$; Push $-45$.
  - Read $/$: Pop $-45$; Pop 4; Compute $4 / (-45) = -0.0888\ldots$; Push $-0.0888\ldots$.
  - Read $-$: Pop $-0.0888\ldots$; Pop 9; Compute $9 - (-0.0888\ldots) = 9.0888\ldots$; Push 9.0888\ldots.
  - Read $\ast$: Pop 9; Pop 9.0888\ldots; Compute $(9.0888\ldots) \ast 9 = 81.8$; Push 81.8.
Example (Postfix Evaluation)

- **Evaluate** $1 2 + 3 * 4 5 6 + 7 8 * - / - 9 *$.

- **Read** $-$: Pop 56; Pop 11; Compute $11 - 56 = -45$; Push $-45$.
- **Read** $/$: Pop $-45$; Pop 4; Compute $4 / (-45) = -0.0888\ldots$; Push $-0.0888\ldots$.
- **Read** $-$: Pop $-0.0888\ldots$; Pop 9; Compute $9 - (-0.0888\ldots) = 9.0888\ldots$; Push $9.0888\ldots$.
- **Read** $*$: Pop 9; Pop 9.0888\ldots; Compute $(9.0888\ldots) \times 9 = 81.8$; Push $81.8$. 
Example (Postfix Evaluation)

- Evaluate $1\ 2\ +\ 3\ *\ 4\ 5\ 6\ +\ 7\ 8\ *\ -\ /\ -\ 9\ *$.

- Read $-$: Pop 56; Pop 11; Compute $11 - 56 = -45$; Push $-45$.
- Read $/$: Pop $-45$; Pop 4; Compute $4 / (-45) = -0.0888\ldots$; Push $-0.0888\ldots$.
- Read $-$: Pop $-0.0888\ldots$; Pop 9; Compute $9 - (-0.0888\ldots) = 9.0888\ldots$; Push $9.0888\ldots$.
- Read $*$: Pop 9; Pop $9.0888\ldots$; Compute $(9.0888\ldots) * 9 = 81.8$; Push $81.8$. 
Postfix Expression Evaluation

Example (Postfix Evaluation)

Evaluate $1 \ 2 \ + \ 3 \ * \ 4 \ 5 \ 6 \ + \ 7 \ 8 \ * \ \ - \ / \ \ - \ 9 \ *$.

Read $-$: Pop 56; Pop 11; Compute $11 - 56 = -45$; Push $-45$.

Read $/$: Pop $-45$; Pop 4; Compute $4/(-45) = -0.0888\ldots$; Push $-0.0888\ldots$.

Read $-$: Pop $-0.0888\ldots$; Pop 9; Compute $9 - (-0.0888\ldots) = 9.0888\ldots$; Push 9.0888\ldots.


Read $*$: Pop 9; Pop 9.0888\ldots; Compute $(9.0888\ldots) \times 9 = 81.8$; Push 81.8.
PostfixEvaluator.cpp

Download PostfixEvaluator.cpp
When a function is called, the program

- Pushes the values of the parameters.
- Pushes the address of the next instruction (to which the function should return later).
- Allocates space on the stack for the local variables.
- Branches to the first line in the function.
Handling Function Calls

The Stack

Begin with the current stack
Handling Function Calls

The Stack

Other Stuff

Function Parameters

Push the function parameters
Handling Function Calls

The Stack

Other Stuff

Function Parameters

Return Address

Push the return address
Handling Function Calls

The Stack

- Other Stuff
- Function Parameters
- Return Address
- Local Variables

Push the local variables
When a function returns, the program
- Pops (and discards) the values of the local variables.
- Pops the return address (and stores it in the IP register).
- Branches to the return address.
- Pops the parameters.

The stack has now been returned to its previous state.
Handling Function Calls

The Stack

Other Stuff

Function Parameters

Return Address

Pop the local variables
Handling Function Calls

- The Stack
- Other Stuff
- Function Parameters
- Pop the return address
Handling Function Calls

The Stack

Other Stuff

Pop the function parameters
While waiting for your order in a restaurant, you pick up and play the Triangle Puzzle.
The Restaurant Triangle Puzzle

- There are fifteen holes.
- One hole is empty.
- The other 14 holes have pegs in them.
- You make moves by jumping, as in checkers.
- One peg jumps an adjacent peg, landing in an empty hole.
- Remove the jumped peg.
The Restaurant Triangle Puzzle

- The goal is to remove all but one peg.
The Restaurant Triangle Puzzle

There are 36 possible moves (6 in each of 6 directions).
Six moves up this diagonal.
The Restaurant Triangle Puzzle

- Six moves down this diagonal.
Six moves up this diagonal.
The Restaurant Triangle Puzzle

- Six moves down this diagonal.
Six moves right horizontally.
The Restaurant Triangle Puzzle

Six moves left horizontally.
List the 36 possible moves in a particular order.
The program will systematically consider the moves until it finds a valid move.
Then it will
  - Make that move.
  - Push that move onto the stack.
The Restaurant Triangle Puzzle

- If no move if possible, then it will
  - Pop the last move off the stack.
  - Continue through the list of possible moves until it finds a legal move.
  - Make that move and continue as before.
- When a single peg remains, the contents of the stack is a winning sequence of moves.
Assignment

Homework

- Read Section 7.4, pages 366 - 370.
- Read Section 7.5, pages 371 - 383.