SCT222-0127/2023 Franas Baraka Jumaa

Data structures and Algorithms.

~~Convert arithmetic to postfix notation.~~

Convert into an expression tree (a+b/c*d-e).

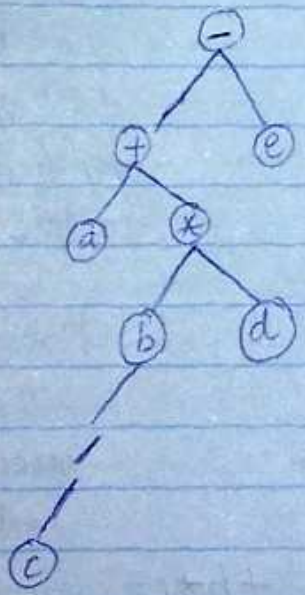
~~Symbol~~ - first convert to postfix.

Incoming token	Stack	postfix.
a	-	a
+	+	a
b	+	ab
/	+/	ab
c	+/ pop	abc
*	+*	abc/
d	+* pop	abc/d
-	-	abc/d*+
e	-	abc/d*+e
-		abc/d*+e-

Building the expression tree.

((a+((b/c)*d))-e) - The

- The root of the tree is the - node.



2. Convert the same into a postfix a+b/c*d-e. (show the process.)

Incoming token	stack	output
a	-	a
+	+	a
b	+	ab
/	+/	ab
c	+/ - pop and print	abc
*	+*	abc/
d	+* ~~pop and print~~	abc/d*+
-	-	~~abc/d*+~~ abc/d*+e
e	-	abc/d*+e
-		abc/d*+e-

abc/d*+e- postfix.

3. Convert +ax bc into an infix.

Incoming Token	stack	Operation.
c	c	push operand c
b	cb	push operand b.
*	b*c	pop b and c, combine to b*c and push.
a	b*c a	push operand a
+	a+b*c	pop a and b*c, combine to a+b*c and push.

The resultant Infinix expression is a+b*c