Infix to Postfix

- Note that the postfix form an expression does not require parenthesis.
- Consider ‘4+3*5’ and ‘(4+3)*5’. The parenthesis are not needed in the first but they are necessary in the second.
- The postfix forms are:

4+3*5	435*+
(4+3)*5	43+5*

Converting Infix to Postfix

- Consider the infix expressions ‘A+B*C’ and ‘ (A+B)*C’.
- The postfix versions are ‘ABC*+’ and ‘AB+C*’.
- The order of operands in postfix is the same as the infix.
- In scanning from left to right, the operand ‘A’ can be inserted into postfix expression.

Converting Infix to Postfix

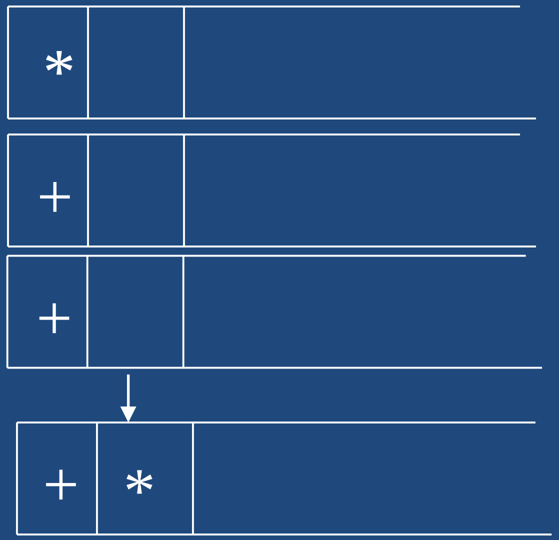
- The ‘+’ cannot be inserted until its second operand has been scanned and inserted.
- The ‘+’ has to be stored away until its proper position is found.
- When ‘B’ is seen, it is immediately inserted into the postfix expression.
- Can the ‘+’ be inserted now? In the case of ‘A+B*C’ cannot because * has precedence.

Converting Infix to Postfix

- In case of ‘(A+B)*C’, the closing parenthesis indicates that ‘+’ must be performed first.
- Assume the existence of a function ‘prcd(op1,op2)’ where op1 and op2 are two operators.
- Prcd(op1,op2) returns TRUE if op1 has precedence over op2, FASLE otherwise.

Converting Infix to Postfix

- prcd(‘*’,’+’) is TRUE
- prcd(‘+’,’+’) is TRUE
- prcd(‘+’,’*’) is FALSE



Converting Infix to Postfix

- Example: A + B * C

symb	postfix	stack
A	A	

Converting Infix to Postfix

- Example: A + B * C

symb	postfix	stack
A	A	
+	A	+

Converting Infix to Postfix

- Example: A + B * C

symb	postfix	stack
A	A	
+	A	+
B	AB	+

Converting Infix to Postfix

- Example: A + B * C

symb	postfix	stack
A	A	
+	A	+
B	AB	+
*	AB	+ *

Converting Infix to Postfix

- Example: A + B * C

symb	postfix	stack
A	A	
+	A	+
B	AB	+
*	AB	+ *
C	ABC	+ *

Converting Infix to Postfix

- Example: A + B * C

symb	postfix	stack
A	A	
+	A	+
B	AB	+
*	AB	+ *
C	ABC	+ *
	ABC *	+

Converting Infix to Postfix

- Example: A + B * C

symb	postfix	stack
A	A	
+	A	+
B	AB	+
*	AB	+ *
C	ABC	+ *
	ABC *	+
	ABC * +	

Converting Infix to Postfix

- Handling parenthesis
- When an open parenthesis ‘(‘ is read, it must be pushed on the stack.
- This can be done by setting prcd(op,‘(‘) to be FALSE.
- Also, prcd(‘(‘,op) == FALSE which ensures that an operator after ‘(‘ is pushed on the stack.

Converting Infix to Postfix

- When a ‘)’ is read, all operators up to the first ‘(‘ must be popped and placed in the postfix string.
- To do this, prcd(op,’)’) == TRUE.
- Both the ‘(‘ and the ‘)’ must be discarded: prcd(‘(‘,’)’) == FALSE.

```
if( s.empty()  ||  symb != ‘)’ )
        s.push( c );
else
        s.pop(); // discard the ‘(‘
```

Converting Infix to Postfix

prcd(‘(‘, op) = FALSE	for any operator
prcd(op, ‘(’) = FALSE	for any operator other than ‘)’
prcd(op, ‘)’) = TRUE	for any operator other than ‘(‘
prcd(‘(‘,’)’) == FALSE.	
prcd(‘)’, op) = error	for any operator.

- Here is the algorithm that converts infix expression to its postfix form.

Converting Infix to Postfix

```
Stack s;
While( not end of input ) {
    c = next input character;
    if( c is an operand )
        add c to postfix string;
    else {
        while( !s.empty() && prcd(s.top(),c) ){
            op = s.pop();
            add op to the postfix string;
        }
        if( s.empty()  ||  c != ‘)’ )
            s.push( c );
        else
            s.pop();                    // discard the ‘(‘
}
    while( !s.empty() ) {
        op = s.pop();
        add op to postfix string;
    }
```

Converting Infix to Postfix

- Example: (A + B) * C

symb	postfix	stack
(		(
A	A	(
+	A	(+
B	AB	(+
)	AB +	
*	AB +	*
C	AB + C	*
	AB + C *	

Task

Infix	Postfix
A + B	A B +
12 + 60 – 23	12 60 + 23 –
(A + B)*(C – D)	A B + C D – *
A ↑ B * C – D + E/F	A B ↑ C*D – E F/+