

Conversion of Infix to Postfix Using a Stack

We will only consider the operators for addition, subtraction, multiplication, exponentiation, and negation. To avoid the ambiguity I discussed in class, we use Python (not FORTRAN) operator precedence and associativity.

The ambiguity arose with an expression like $a^b - c$. It could mean a^{b-c} or $a^b - c$. By adopting Python precedence, it means $a^b - c$. The other expression is written in L^AT_EX as $a^{\{b-c\}}$, and we write $a^{(b-c)}$ for this problem.

Operator Precedence

Here are the precedences of the operators.

	operator	symbol	associativity
1.	negation	\sim	
2.	exponentiation	\wedge	right-to-left
3.	multiplication	$*$	left-to-right
4.	addition, subtraction	$+, -$	left-to-right

Operator precedence is not relevant in prefix and postfix expressions.

The Stack Algorithm

We assume there is a stack, and the bottom symbol is $\$$. We also assume the input ends with an end of file symbol, which we also write as $\$$. For convenience, we replace each “ $-$ ” in the input which denotes negation by \sim .¹ Here are the rules of the algorithm which changes infix to postfix.

1. If the current input symbol is a variable, read and write it.
2. If the current input symbol is a left parenthesis, read and push it.
3. If the current input symbol is a right parenthesis and the top symbol of the stack is an operator, pop the stack and write it.
4. If the current input symbol is a right parenthesis and the top symbol of the stack is a left parenthesis, read and pop and discard both symbols.
5. If the current input symbol and the top symbol of the stack are both operators, read and push if the input symbol has higher precedence, otherwise pop and write.

¹How can you tell? If the $-$ is preceded by an operator or a left parenthesis, or is the first symbol in the expression, it denotes negation; otherwise it denotes subtraction.

The following table summarizes the actions. The column header indicates the next symbol of the input file, while the row header indicates the top symbol of the stack.

	Variable	+	-	*	^	~	()	\$
\$	read write	read push	read push	read push	read push	read push	read push		HALT
+	read write	pop write	pop write	read push	read push	read push	read push	pop write	pop write
-	read write	pop write	pop write	read push	read push	read push	read push	pop write	pop write
*	read write	pop write	pop write	pop write	read push	read push	read push	pop write	pop write
^	read write	pop write	pop write	pop write	read push	read push	read push	pop write	pop write
~	read write	pop write	pop write	pop write	pop write	pop write	read push	pop write	pop write
(read write	read push	read push	read push	read push	read push	read push	read pop	