Universidade do Estado do Amazonas
Escola Superior de Tecnologia
Data: 4 de outubro de 2016
Disciplina: Fundamentos Teóricos da Computação
Professores: Elloá B. Guedes
Aluno:

Projeto Prático 2

1 Apresentação

Neste projeto prático 2 nós vamos tratar da resolução de expressões aritméticas com precedência utilizando o algoritmo *shunting-yard*, proposto por E. Dijkstra em 1961.

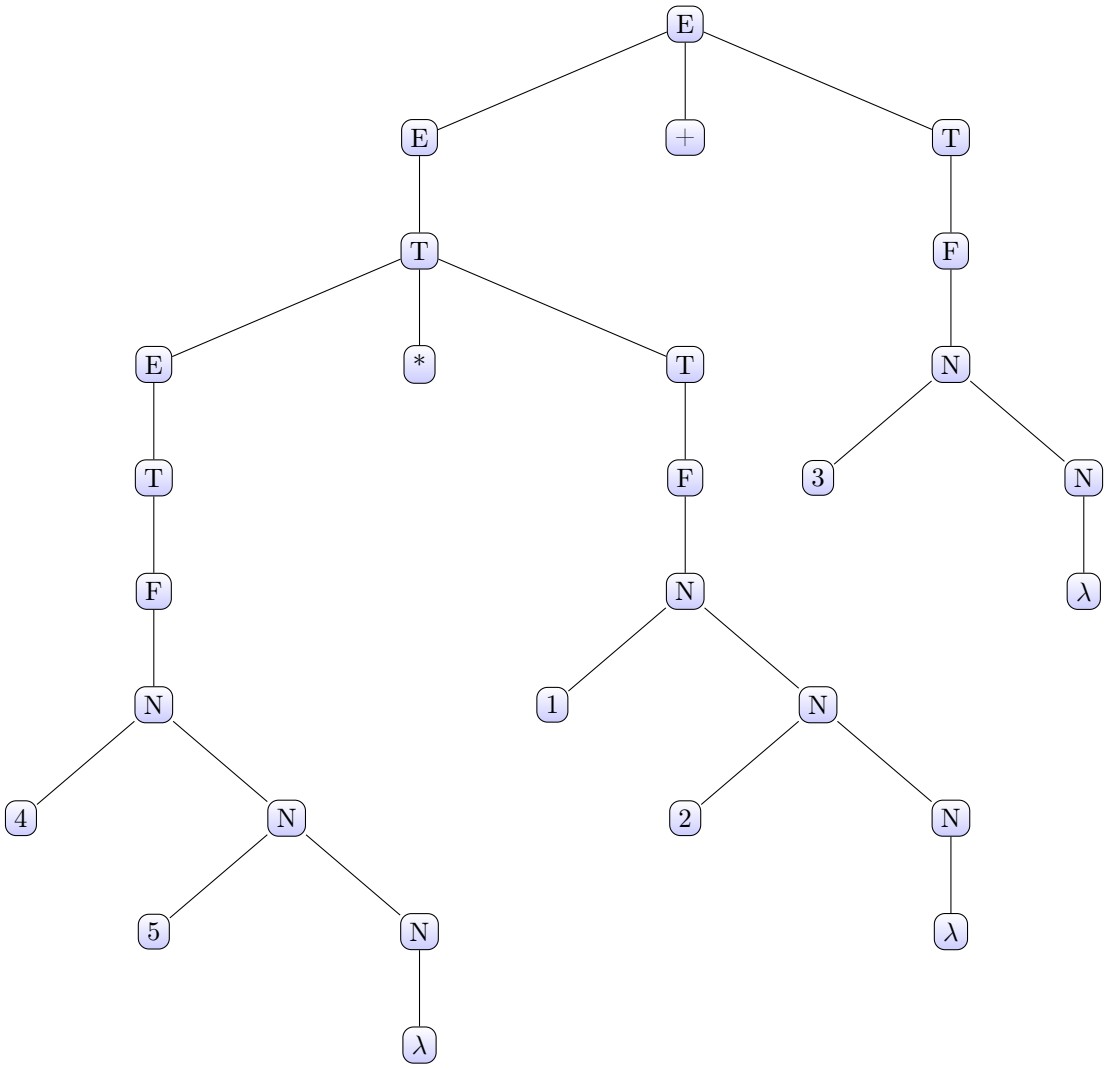
Ao estudar as Linguagens Livres de Contexto e os modelos capazes de reconhecer esta classe de linguagens, as gramáticas livres de contexto e os autômatos finitos não-determinísticos com pilha, vimos que conceitos como recursividade e a estrutura de dados pilha estão intrinsecamente envolvidos. Temos visto ainda que uma das principais aplicações das Linguagens Livres de Contexto se dá na concepção das Linguagens de Programação, o que é visto amplamente na disciplina de Compiladores.

Uma das gramáticas livres de contexto que já vimos em diversos exercícios é a gramática livre de contexto para expressões aritméticas, dada por:

$$\begin{aligned} E &\rightarrow E+T|E-T|T \\ T &\rightarrow E*T|E/T|F \\ F &\rightarrow N|E \\ N &\rightarrow 0N|1N|2N|\ldots|9N|\lambda \end{aligned}$$

em que E é a variável inicial da gramática e denota uma expressão, T denota termo, F denota fator e N denota um número. Observe que, de acordo com esta gramática, podemos ter expressões como $9*6$, $3+155*48$, $965-45/13+98$.

Ao analisarmos uma expressão matemática produzida por esta gramática, podemos inicialmente ter dúvidas quanto ao valor da expressão resultante. Por exemplo, para a expressão $45*12+3$, pode-se pensar nos resultados 171 ou 675. Mas qual deles é o correto? A resposta virá da árvore de derivação para esta expressão mostrada na figura a seguir.



Para que a expressão E que está na raiz da árvore seja resolvida, é necessário resolver suas subárvores direita e esquerda, e assim recursivamente. Deste modo, a operação de multiplicação será realizada antes da operação de soma, indicando que a multiplicação possui maior precedência do que a soma. No nosso caso, as operações de multiplicação e divisão possuem maior precedência que as operações de soma e subtração.

Quando as expressões são apenas compostas por números positivos e não são parentizadas, o algoritmo de *shunting-yard* pode nos ajudar a resolvê-las. De acordo com este algoritmo, para resolvermos uma expressão aritmética precisamos de duas pilhas: uma pilha que irá receber apenas números e uma pilha que receberá apenas operadores ($+$,$-$,$*$,$/$). A entrada é escaneada da esquerda para a direita. Sempre que encontrar um número na entrada, empilhe-o na pilha de números. A cada operador encontrando, compare-o com o topo da pilha. Se o que está no topo da pilha possui maior precedência, desempilhe-o, desempilhe dois números da pilha de números, aplique-lhes o operador e empilhe o resultado de volta na pilha de números. Repita a comparação comparando o

operador "pendente" com o operador que está no topo da pilha.

Para exemplificar o algoritmo, considere que você deseja resolver a expressão $1 + 2 * 3 + 4 * 5$. Empilhe o 1, com o + note que a pilha está vazia, empilhe-o na pilha de operadores, empilhe o 2, com o $*$, note que o + que está no topo da pilha não possui maior precedência, então empilhe o $*$, empilhe o 3 na pilha de números. Apenas para facilitar o entendimento, visualize a situação das pilhas e da entrada a seguir:

- Entrada: $1 + 2 * \underline{3} + 4 * 5$
- Pilha de números: (topo) [3, 2, 1]
- Pilha de operadores: (topo)[*,+]

Procedendo com a entrada, note que o operador + será lido. Comparando-o com o topo da pilha, este possui maior precedência, então desempilha-se o $*$, desempilham-se dois números da pilha de números, realiza-se a operação com eles e empilha-se o resultado na pilha de números. Compara o + com o topo da pilha, que também é +, então empilha-se o +. Observe a situação das pilhas e da entrada a seguir:

- Entrada: $1 + 2 * 3\underline{+}4 * 5$
- Pilha de números: (topo) [6, 1]
- Pilha de operadores: (topo)[+,+]

Dando prosseguimento, empilha-se o 4 na pilha de números. Ao ler o $*$, percebe-se que o topo da pilha não possui maior precedência, então empilha-o. Empilha o 5 na pilha de números. Como a entrada acabou, para cada operador na pilha de operadores, desempilha-se dois números da pilha de números e empilha-se o resultado. O valor resultante na pilha de números é o resultado correto da expressão aritmética. Neste caso, temos:

- Entrada: $1 + 2 * 3\underline{+}4 * 5$
- Pilha de números: (topo) [5,4, 6, 1]
- Pilha de operadores: (topo)[*,+,+]

Resolvendo, tem-se $[20, 6, 1]$ e $[+, +]$, depois $[26, 1]$ e $[+]$ e, por fim, $[27]$ e $[]$. Conclui-se então que o resultado da expressão é 27.

Neste projeto, você receberá uma string como entrada contendo uma expressão aritmética. A saída do seu programa deve ser um número correspondendo ao resultado da expressão aritmética correspondente, seguindo a precedência dos operadores conforme especificado no enunciado. O programa que resolve o problema proposto deve implementar o algoritmo de shunting-yard. Soluções que fizerem uso de bibliotecas de terceiros para implementar este algoritmo ou outro equivalente serão desconsideradas. A linguagem de programação a ser considerada será o Python 3.

O resultado a ser considerado das expressões aritméticas será um número do tipo float impresso com duas casas decimais. Caso a expressão resulte em erros ou tenha sido mal construída, apresente a mensagem "Error". Em nenhuma das entrada será considerada a ocorrência de parênteses ou outros caracteres que não os especificados no enunciado.

2 Links Úteis

- Artigo original do autor do algoritmo
- Explicação do algoritmo Shunting-yard
- Algoritmo Shunting-yard
- Filas e Pilhas em Python.

3 Exemplos de Entrada e Saída

Entrada	**Saída**
1 + 2 + 3+ 4 + 5	15.00
1 * 2 * 3 * 4 * 5	120.00
1 / 0	Error
1 + * 24	Error

4 Prazos Importantes

- Apresentação da atividade: 04/10
- Cadastro da atividade no Run.Codes: 04/10
- Período de submissões: 04/10 até 13/10
- Data limite de entrega: 13/10, 13h.

Mãos à obra!
"Se você não trabalhar pelos seus próprios sonhos, acabará contratado para trabalhar pelos sonhos de alguém." (Steve Jobs)