Context-free Parsing

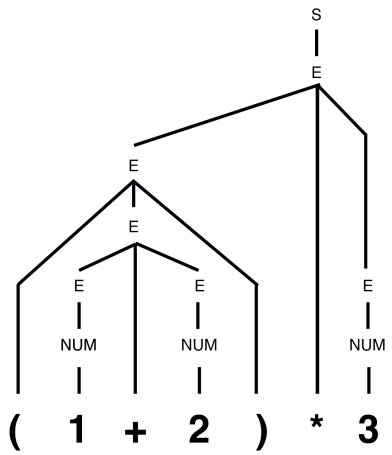
Irene Font Peradejordi

The following context-free grammar G describes the structure of simple mathematical expressions.

G = (V, Σ, R, S):

V={ E, NUM }
Σ = { (,), +, -, *, /, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 }
R = {
E → (E),
E → E + E,
E → E - E,
E → E * E,
E → E / E,
E → NUM,
NUM → 0,
NUM → 1,
NUM → 2,
NUM → 3,
NUM → 4,
NUM → 5,
NUM → 6,
NUM → 7,
NUM → 8,
NUM → 9
}
S = E

1. **Draw a parse tree of the sequence "(1 + 2) * 3" according to the grammar.**



2. **Rewrite the grammar rules such that all rules conform to Chomsky Normal Form.**

NUM → 0
NUM → 1
NUM → 2
NUM → 3
NUM → 4
NUM → 5
NUM → 6
NUM → 7
NUM → 8
NUM → 9

P1 → (

P2 →)

SUM → +

MUL → *

NEG → -

DIV → /

X1 → P1 NUM

NUM → X1 P2

X2 → NUM SUM

NUM → X2 NUM

X3 → NUM NEG

NUM → X3 NUM

X4 → NUM MUL

NUM → X4 NUM

X5 → NUM DIV

NUM → X5 NUM

3. **Apply the Earley algorithm using the original grammar to the sequence "0 + 1 - 2" and write down the resulting chart as your answer.**

$_0$ **0** $_1$ **+** $_2$ **1** $_3$ **–** $_4$ **2** $_5$

Entry 0

∂	→	· S	[0 – 0]
S	→	· E	[0 – 0]

E	→	· (E)	[0 – 0]
E	→	· E + E	[0 – 0]
E	→	· E - E	[0 – 0]
E	→	· E * E	[0 – 0]
E	→	· E / E	[0 – 0]
E	→	· NUM	[0 – 0]
NUM	→	· 0	[0 – 0]
NUM	→	· 1	[0 – 0]
NUM	→	· 2	[0 – 0]
NUM	→	· 3	[0 – 0]
NUM	→	· 4	[0 – 0]
NUM	→	· 5	[0 – 0]
NUM	→	· 6	[0 – 0]
NUM	→	· 7	[0 – 0]
NUM	→	· 8	[0 – 0]
NUM	→	· 9	[0 – 0]

Entry 1

NUM	→	0 ·	[0 – 1]
E	→	NUM ·	[0 – 1]
E	→	E · + E	[0 – 1]
E	→	E · - E	[0 – 1]
E	→	E · * E	[0 – 1]
E	→	E · / E	[0 – 1]
S	→	E ·	[0 – 1]
∂	→	S ·	[0 – 1]

Entry 2

E	→	E + · E	[0 – 2]

Entry 3

NUM	→	1 ·	[0 – 3]
E	→	NUM ·	[0 – 3]
E	→	E + E ·	[0 – 3]
E	→	E · - E	[0 – 3]
S	→	E ·	[0 – 3]
∂	→	S ·	[0 – 3]

Entry 4

E	→	E - · E	[0 – 4]

Entry 5

NUM	→	2 ·	[0 – 5]
E	→	NUM ·	[0 – 5]
E	→	E – E ·	[0 – 5]
S	→	E ·	[0 – 5]
∂	→	S ·	[0 – 5]

All complete states

NUM	→	0 ·	[0 – 1]
NUM	→	1 ·	[0 – 3]
E	→	E + E ·	[0 – 3]
NUM	→	2 ·	[0 – 5]
E	→	NUM ·	[0 – 5]
E	→	E – E ·	[0 – 5]
S	→	E ·	[0 – 5]
∂	→	S ·	[0 – 5]

4. **Does the sequence you have parsed in question 3 have an ambiguous syntactic structure? How can you tell? Relate your answer to the information the chart provides you.**

   Yes, it is ambiguous because we can either solve the + first (0 + 1) with the rule E → E + E and then do the subtraction (1 – 2) with the rule E → E – E, or do the subtraction first and then the addition.

We can see that it is ambiguous because we solve the rule S → E in three different stages. A non-ambiguous sentence would solve it maximum twice as we cannot have a tree with more than 2 branches.

[I am not completely sure about the "2 branches" reasoning. I know it is ambiguous – because you can create two different trees- but I don't know how to identify it with the Earley parsing results. So, after the assignment is graded, I will send you an email asking is this make sense]