

ANTLR

Another **T**ool for **L**anguage **R**ecognition

Overview

- ANTLR is a parser generator (also called a *compiler compiler*)
- First developed in 1989, and still actively developed. Maintained by Prof. Terence Parr of the University of San Francisco
- Used to build programming languages, domain specific languages, translators, compilers
- This tutorial refers to ANTLR v3

Features

- Language description in a grammar file (Grammar.g) using EBNF → using a formal language to describe a formal language
- Grammar is compiled into Lexer and Parser
- Code generation available for various programming languages: Java, Python, C++, C#

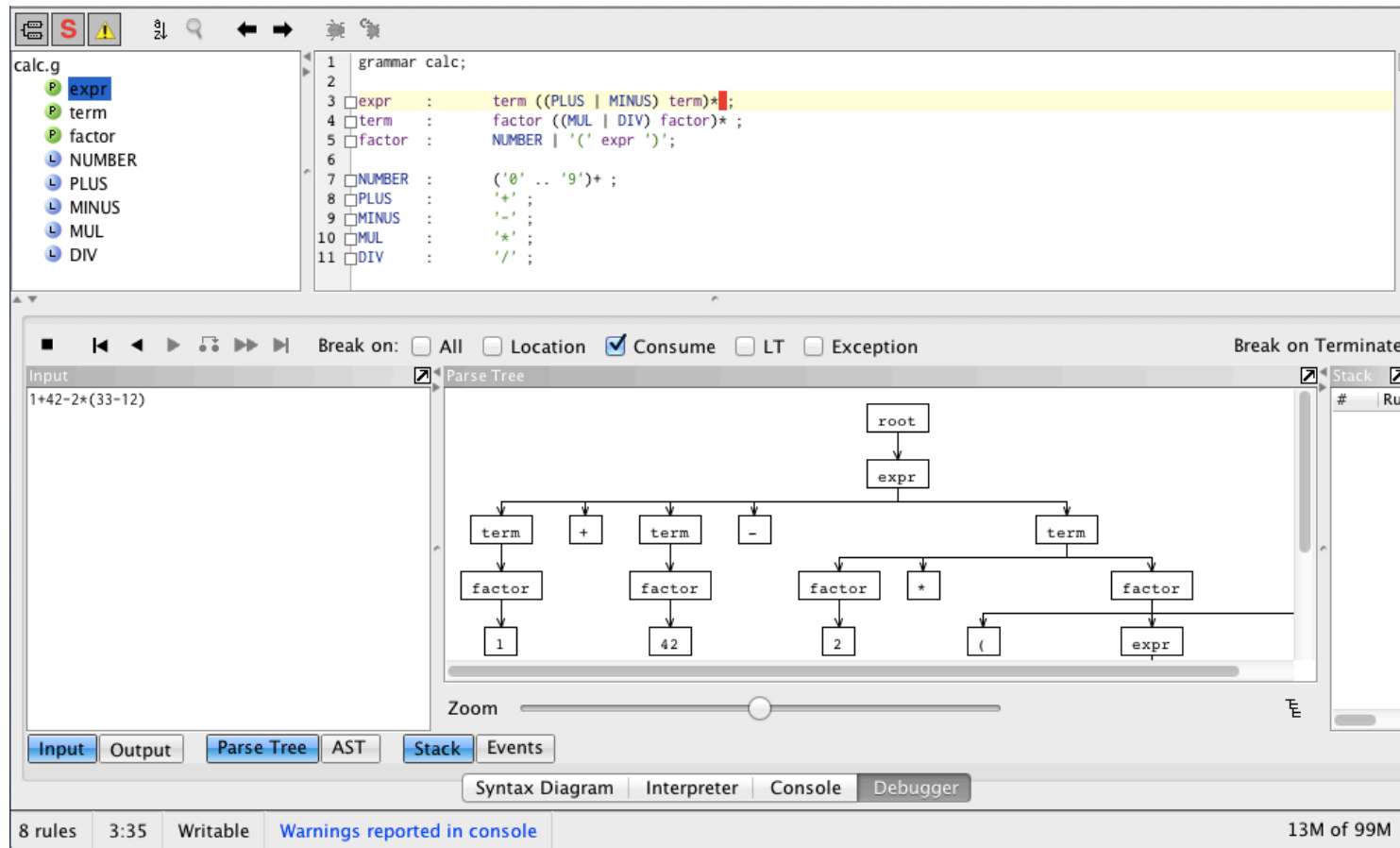
General Workflow

- Describe grammar
- Compile grammar into...
 - **Lexer**: takes a stream of characters and divides the stream into tokens according to pre-set rules
 - **Parser**: reads tokens and interprets them according to its rules
- Use Lexer and Parser as part of own program

Extended Backus-Naur Form (EBNF)

- ? preceding token may occur 0 or 1 times
- * preceding token may occur 0 or more times
- + preceding token may occur 1 or more times
- . any character or token
- ~ following character/token may not occur at this position
- .. spans a range
- () grouping

ANTLRWorks



The screenshot displays the ANTLRWorks IDE interface. At the top, a file named 'calc.g' is open, showing a grammar definition with 11 rules. The current rule being viewed is 'expr', which is highlighted in yellow. Below the grammar editor, the 'Input' field contains the expression '1+42-2*(33-12)'. The 'Parse Tree' view shows a hierarchical tree structure representing the parsed expression. The root node is 'root', which points to 'expr'. 'expr' branches into 'term', '+', 'term', and '-'. The first 'term' branches into 'factor', which points to the terminal '1'. The second 'term' branches into 'factor', which points to the terminal '42'. The third 'term' branches into 'factor' and '*'. The 'factor' node further branches into '(', 'expr', and ')'. The 'expr' node under this factor branches into 'term', '+', and 'term'. The first 'term' branches into 'factor', which points to the terminal '2'. The second 'term' branches into 'factor', which points to the terminal '33'. The final 'term' branches into 'factor', which points to the terminal '12'. The 'Stack' view on the right is empty. At the bottom, the status bar shows '8 rules', '3:35', 'Writable', 'Warnings reported in console', and '13M of 99M'.

- Allows to develop, analyze and debug grammars

Resources

- ANTLR v3: <http://antlr3.org>
- ANTLRWorks v3: <http://antlr3.org/works/>
- ANTLR v3 documentation:
<http://www.antlr.org/wiki/display/ANTLR3>
- ANTLR v3 reference:
<http://pragprog.com/book/tpantlr/the-definitive-antlr-reference>
- Example grammars:
<http://antlr3.org/grammar/list.html>