

<http://tiny.cc/hsformulaslides>

Ime	Kiflice	Cena (din)
Nemanja	Kiflice sa ukusom pice	640
Igor	Kiflice sa spanaćem	640
Igor	Kiflice sa šunkom i sirom	640
Vesna	Kiflice sa pršutom i sirom	640
Vesna	Kiflice sa eurokremom	640
Nemanja	Kiflice sa masima	640



HSFormula steps



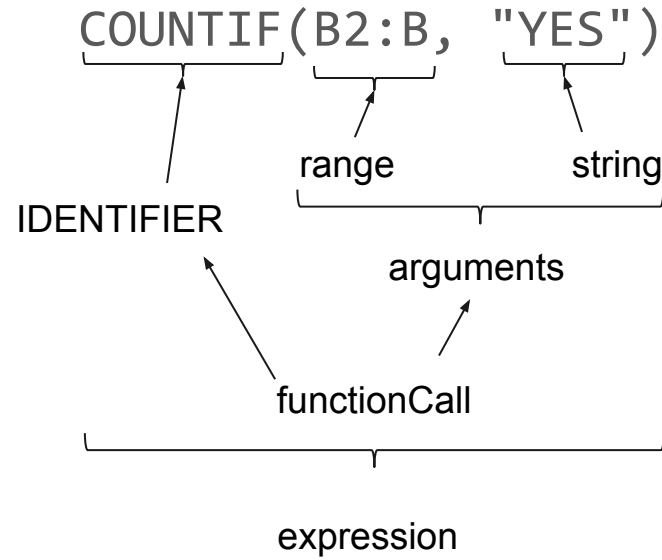


Parsing

COUNTIF(B2:B, "YES")

COUNTIF '(' B2 ':' B ',' "YES" ')'
IDENTIFIER '(' CELL ':' CELL ',' STRING ')'

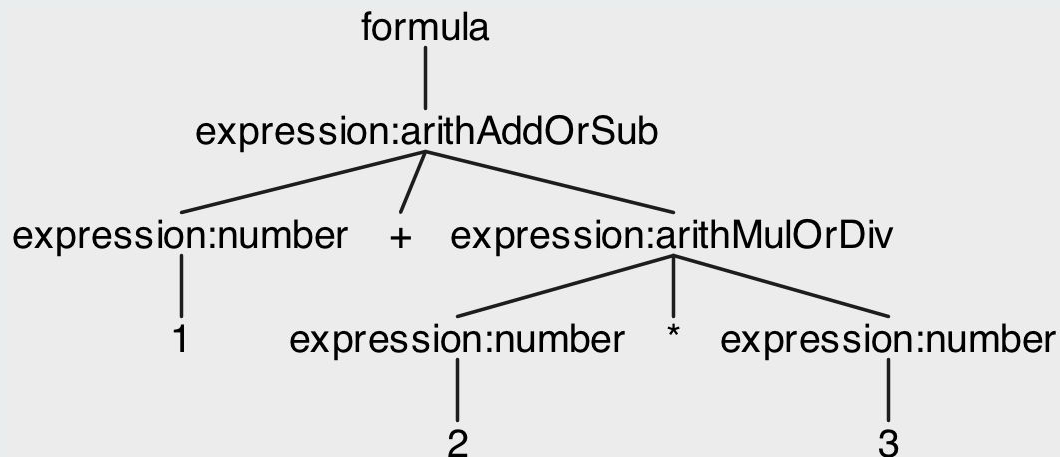
Parsing



Parsing - precedence

expression

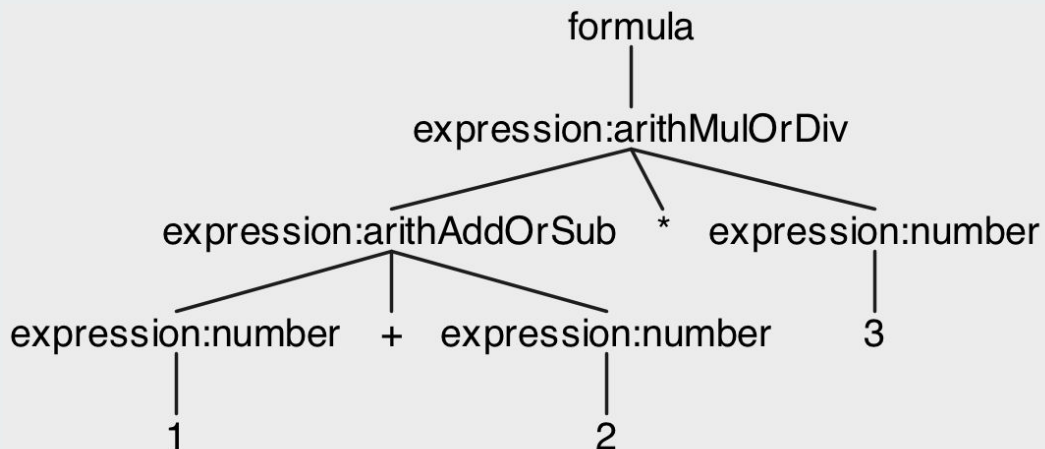
```
: left=expression op=('*' | '/') right=expression # arithMulOrDiv  
| left=expression op=('+' | '-') right=expression # arithAddOrSub
```



Parsing - precedence

expression

```
: left=expression op=('+' | '-') right=expression # arithAddOrSub  
| left=expression op=('*' | '/') right=expression # arithMulOrDiv
```



Lexing - precedence

CELL

: [a-zA-Z][0-9]*

;

B1 : B

CELL ‘.’ CELL

IDENTIFIER

: [a-zA-Z]+

;

Lexing - precedence

IDENTIFIER

: [a-zA-Z]+

;

CELL

: [a-zA-Z][0-9]*

;

B1 : B

CELL ‘.’ IDENTIFIER

ANTLR supports context
sensitive lexing [\[link\]](#)



Listener and Visitor

```
public interface HSFormulaListener extends ParseTreeListener {  
    void enterFormula(HSFormulaParser.FormulaContext ctx);  
    void exitFormula(HSFormulaParser.FormulaContext ctx);  
    void enterNumber(HSFormulaParser.NumberContext ctx);  
    void exitNumber(HSFormulaParser.NumberContext ctx);  
    void enterString(HSFormulaParser.StringContext ctx);  
    void exitString(HSFormulaParser.StringContext ctx);  
    // ...  
}
```



Listener and Visitor

```
public interface HSFormulaVisitor<T> extends ParseTreeVisitor<T> {  
    T visitFormula(HSFormulaParser.FormulaContext ctx);  
    T visitNumber(HSFormulaParser.NumberContext ctx);  
    T visitString(HSFormulaParser.StringContext ctx);  
    T visitFunctionCall(HSFormulaParser.FunctionCallContext ctx);  
    // ...  
}
```



Analysis

- Function calls must be validated
 - Check if function exists
 - Check argument count
 - Check argument type
 - Propagate function return value
- Arithmetic operators must be validated
- We don't validate ranges, let it blow up at runtime instead



Evaluation

HSFormula bails on out of range cells or cells that don't have properly formatted values.

Pretty straightforward - visit and evaluate!



ANTLR resources

grammars-v4 is a repository of ANTLR4 grammars that you can use (check the license first) or let them serve as inspiration.

<https://github.com/antlr/grammars-v4>

The Definitive ANTLR 4 Reference

<https://pragprog.com/book/tpantlr2/the-definitive-antlr-4-reference>

picoComputer Web IDE

The screenshot displays the picoComputer Web IDE interface. At the top, there are control buttons: a play button, a square button, and a download button, followed by a 'Format' button. The main area is divided into three sections:

- Assembly Code:** A list of instructions with line numbers 8 through 21. Line 18, 'sub n, n, 1', is highlighted in blue. The code includes:

```
8  
9   org 8  
10  
11   in n  
12   mov s1, 0  
13   mov s2, 0  
14 petlja:  
15   add s1, s1, n  
16   mul k, n, n  
17   add s2, s2, k  
18   sub n, n, 1  
19   bgt n, 0, petlja  
20   stop s1, s2  
21
```
- Symbol Table:** A table with two columns: 'Symbol' and 'Value'.

Symbol	Value
n	3
s1	3
s2	9
k	9

Register	Value
PC	16
SP	65535
- Console:** A section at the bottom with a red circle containing '0' next to the 'Errors' label. The 'Console' tab is active, showing the text 'Input at address 1: 3'.

Demo

<https://nemanjamiljkovic.me/pico/>

Parsing, Analysis, Bytecode generation, VM

<https://github.com/nmiljkovic/pico-asm>

Vue Frontend, CodeMirror editor

<https://github.com/nmiljkovic/pico-sim>

Q&A

<http://tiny.cc/hsformula>

<http://tiny.cc/hsformulaslides>

@namiljkovic

<https://nemanjamiljkovic.me/>

