

FERIT

Basics of Compiler Theory

Author: Teo Samaržija

My experience with making compilers...


AEC-to-WebAssembly.com x +

← → ↻ flatassembler.github.io/analogClock.html ☆ | 👤 ⋮

Analog Clock

This is my attempt to port the analogClock.aec from [Linux and FreeDOS](#) to WebAssembly, using my new [compiler](#). **This only works in very modern browsers.** I'm not too interested in targeting archaic browsers here. This only works in Firefox 62 and newer or Chrome 69 and newer. After a few years (if not months), all JavaScript environments will become as capable as they are, rather than like Internet Explorer.

AEC program output:



```
*****
*** 12 ***
***11 1***
**      **
*        *
**10      2**
*        *
**      sss **
*      ss  *
*      sss  *
*9 hmmmmm 3*
* hh      *
* h      *
** h      **
*8 h      4*
**      **
*        *
xxx    ** 7    5 **    xxx
xxxxxxx ***    *** xxxxxxxx
Analog Clock for WebAssembly xxxxx*** 6 ***xxxxxx
Made in AEC by    xxxxxxxxxx
Teo Samarzija    xxxxx xxxxx    17:16:10
```

You can see the source code of the AEC program [here](#).

I have made two compilers for my programming language. One is targeting x86 and the other one targeting WebAssembly.

Tokenizer

- A part (usually the first one) of the compiler that tells other parts of the compiler where one word in a programming language ends and where another begins.

`sin(3.14*x+arccos(y)...`



漢字（かんじ）は、中国古代の黄河文明で発祥した表語文字。

Non-tokenized code looks to a computer like Japanese writing looks to us.

Lexical Analyzer

- What type of the word is each word.
- Necessary in programming languages such as C (`typedef` – usually considered a bug in the C programming language).

Time flies like an arrow.

Is *time* a noun or an adjective? Is *flies* a verb or a noun? Is *like* a verb or a conjunction?

Parser

- Tells other parts of the compiler which word is connected to each word grammatically.
- Many frameworks that are supposed to make writing parsers easier: YACC, BISON...

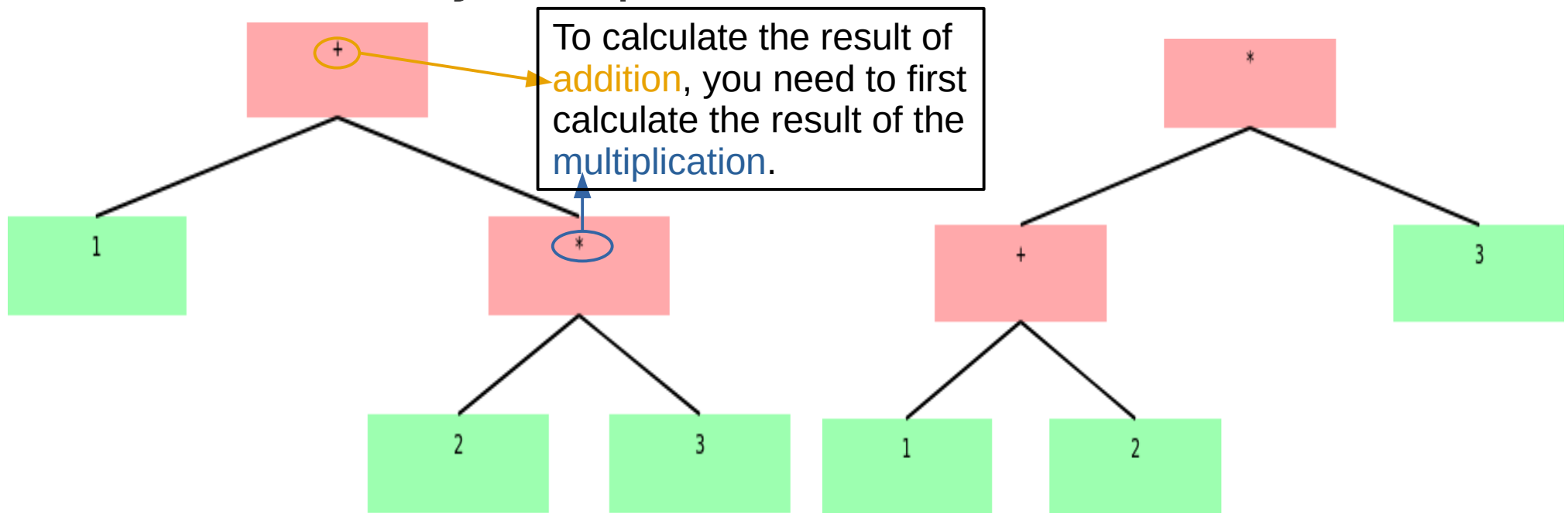
Ea, quae fertilissima totius Germaniae sunt, loca
Graecis aliquibus nota fama esse loquuntur.

“known rumor” - wrong

“known places” - correct

Abstract Syntax Tree

- A structure in the memory of the computer created by the parser.



Abstract syntax tree of $1+2*3$

Abstract syntax tree of $(1+2)*3$



Semantic Analyzer

- Not all sentences that the parser accepts are syntactically correct. The same seems to be true for the parser of natural language in the human brain: *More people have been to Russia than I have.*
- Parser does not denote parts of the sentence (*types* in programming languages), that is the job of semantic analyzer.



Compiler

- The word has two meanings:
 - (broad sense) A program that converts code written in one programming language to another programming language (usually machine code): GCC, CLANG, Visual Studio C++...
 - (strict sense) A subprogram that converts an abstract syntax tree into assembly code. Also called *code generator*. LLVM is a framework supposed to help with that.

Assembly Code

- The machine code (ones and zeros that the computer understands) full of shortened forms (to be easier to read and write).
- Assembler – the program that replaces those contracted forms with ones and zeros.

BTW, IMHO, U R taking it 2 serious.

Assembly language reads to a computer similar to what this sentence reads to an English speaker unfamiliar with the Internet slang.



Linker

- A file with raw machine code is not very useful to a computer.
 - How can the computer even know it is machine code, rather than image or a sound (also represented by ones and zeros)?
- Two types of linkers:
 - Dynamic linker – executable file only contains a part of the program, it calls external libraries assumed to be present in the operating system.
 - Static linker – executable file contains everything necessary for the program to run on some operating system.



Thank you for watching!