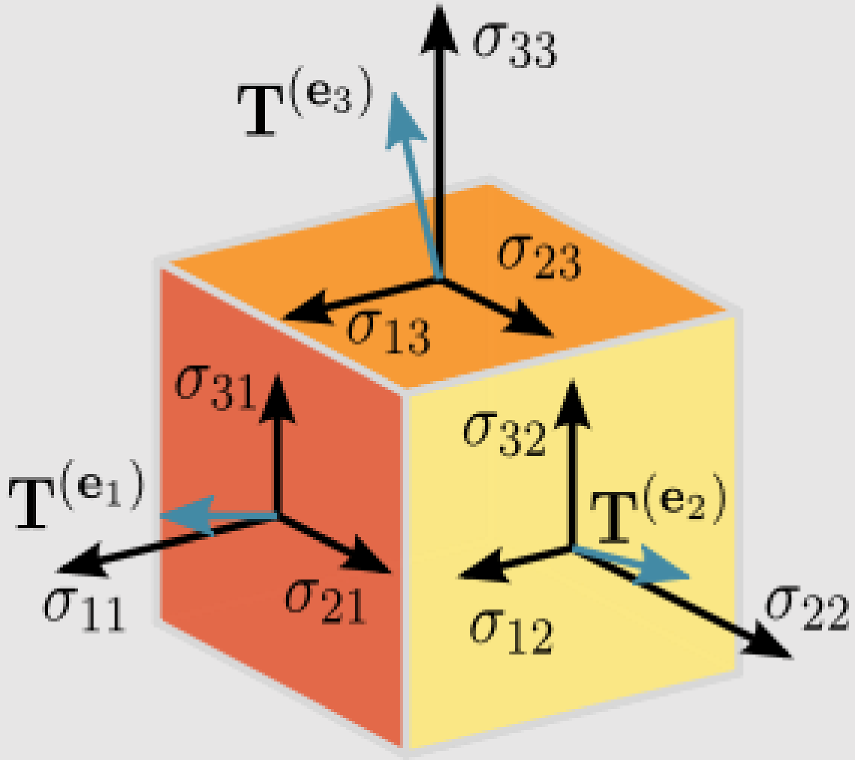
LaTenS

a tensor manipulation language

Mohit Rajpal
Daniel Schwartz
Elsbeth Turcan
Eliana Ward-Lev

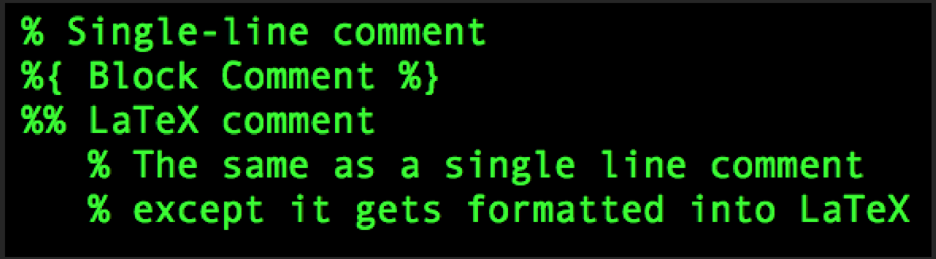
Motivation

- What is a tensor?
- Native n-dimensional tensors
- Intuitive syntax: LaTeX and tensors



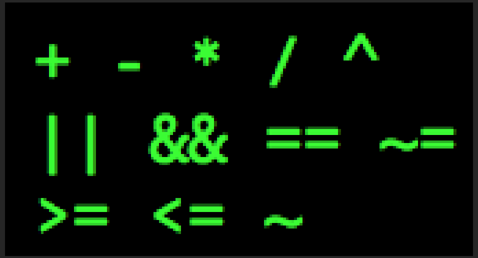
Basic Syntax

Comments



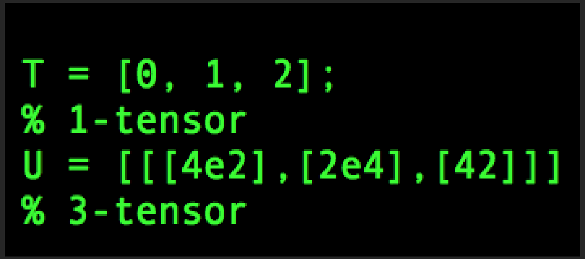
```
% Single-line comment
%{ Block Comment %}
%% LaTeX comment
   % The same as a single line comment
   % except it gets formatted into LaTeX
```

Operators



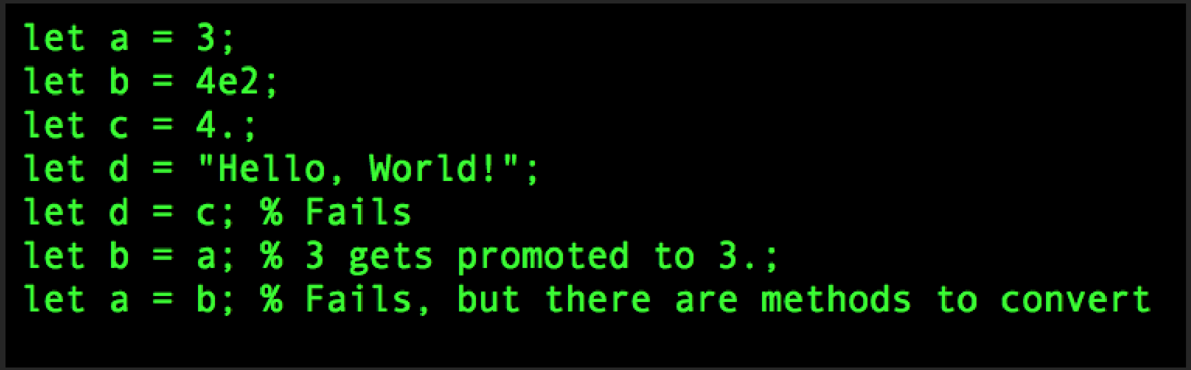
```
+  -  *  /  ^
||  &&  ==  ~=
>=  <=  ~
```

Tensors



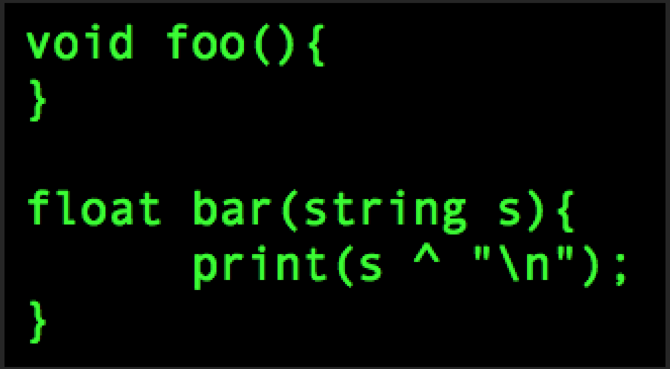
```
T = [0, 1, 2];
% 1-tensor
U = [[[4e2],[2e4],[42]]]
% 3-tensor
```

Inferred, Static Typing



```
let a = 3;
let b = 4e2;
let c = 4.;
let d = "Hello, World!";
let d = c; % Fails
let b = a; % 3 gets promoted to 3.;
let a = b; % Fails, but there are methods to convert
```

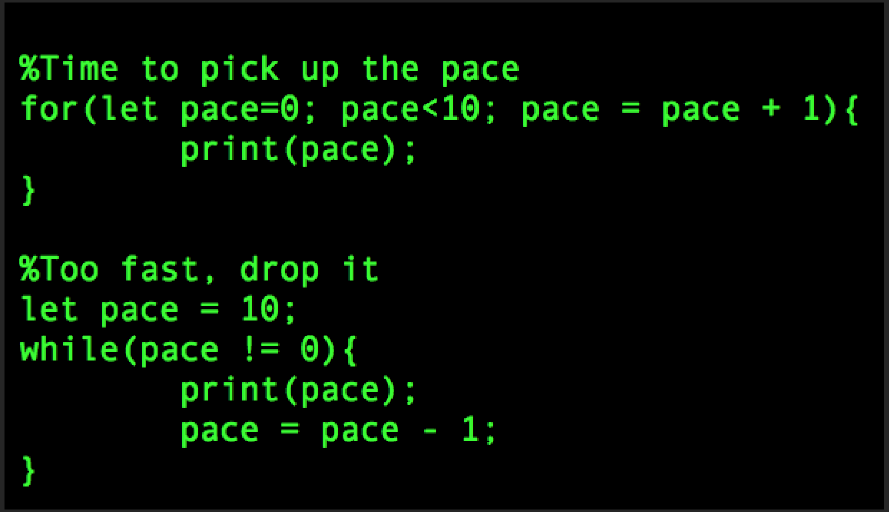
Function Defs



```
void foo(){
}

float bar(string s){
       print(s ^ "\n");
}
```

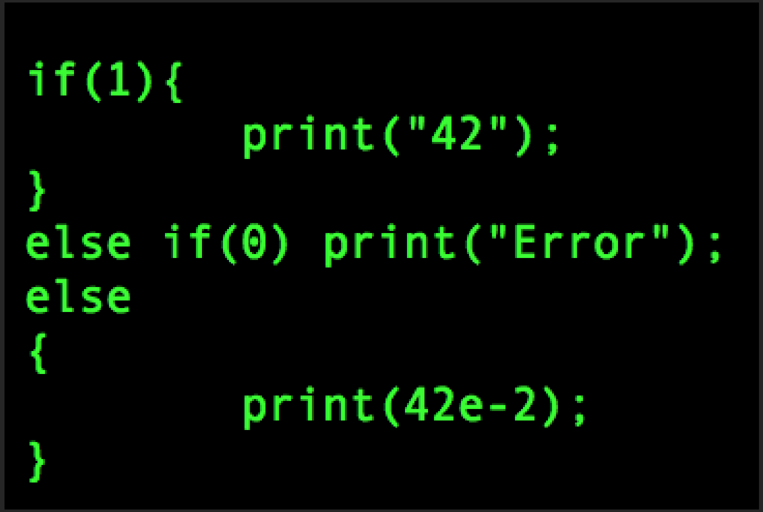
Control Flow

Loops



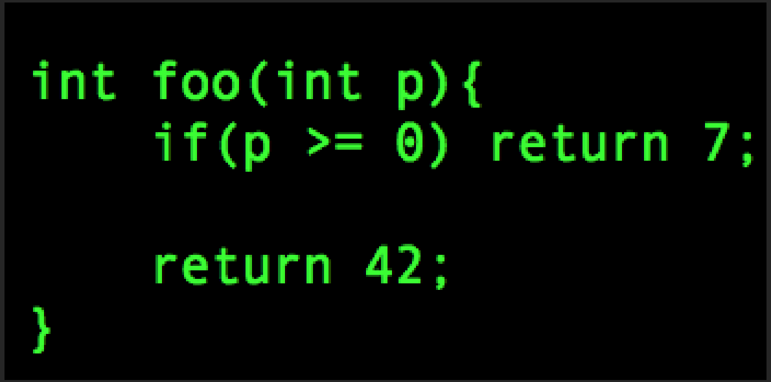
```
%Time to pick up the pace
for(let pace=0; pace<10; pace = pace + 1){
        print(pace);
}

%Too fast, drop it
let pace = 10;
while(pace != 0){
        print(pace);
        pace = pace - 1;
}
```

IF-THEN-ELSE



```
if(1){
        print("42");
}
else if(0) print("Error");
else
{
        print(42e-2);
}
```

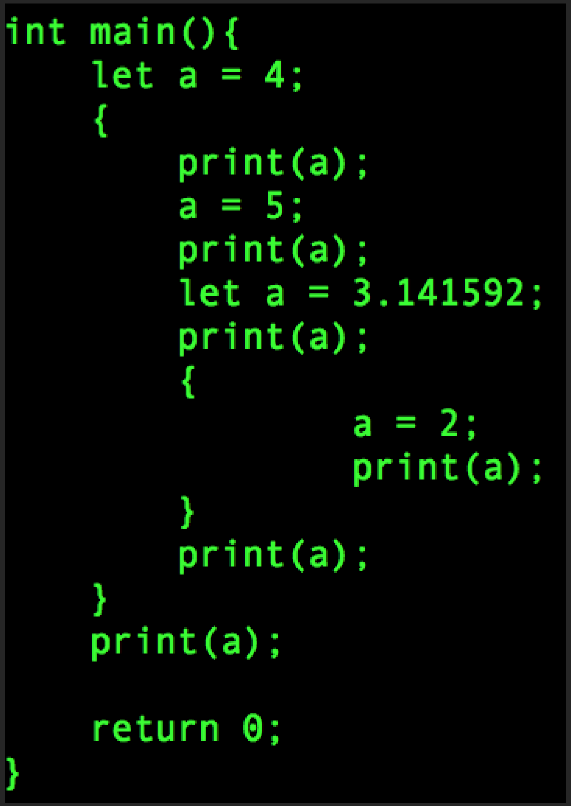
Return



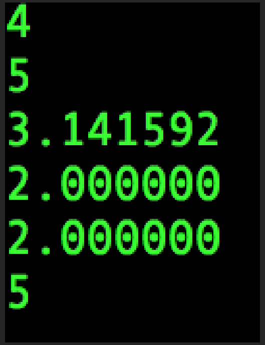
```
int foo(int p){
    if(p >= 0) return 7;

    return 42;
}
```

More Features

Scoping: Static, C-Style

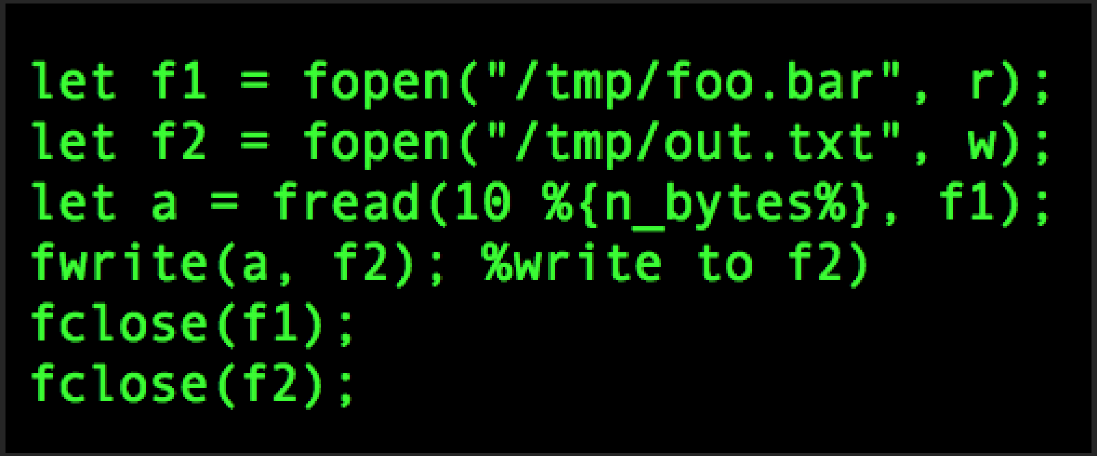


```
int main(){
    let a = 4;
    {
        print(a);
        a = 5;
        print(a);
        let a = 3.141592;
        print(a);
        {
                a = 2;
                print(a);
        }
        print(a);
    }
    print(a);

    return 0;
}
```



```
4
5
3.141592
2.000000
2.000000
5
```

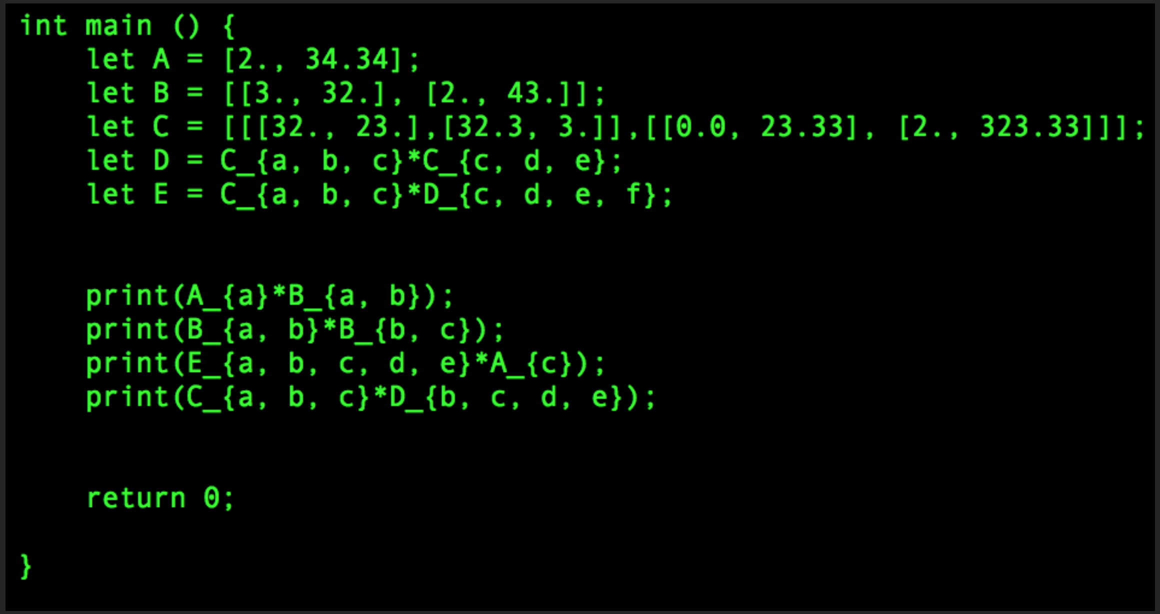
File I/O, Formatted Strings



```
let f1 = fopen("/tmp/foo.bar", r);
let f2 = fopen("/tmp/out.txt", w);
let a = fread(10 %{n_bytes%}, f1);
fwrite(a, f2); %write to f2)
fclose(f1);
fclose(f2);
```

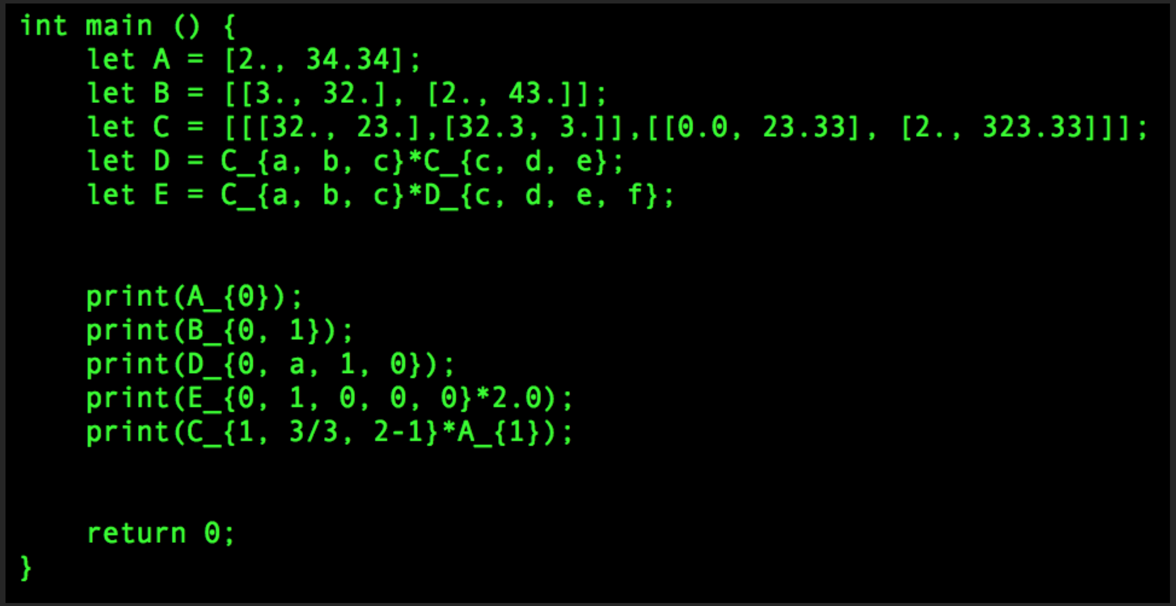
Still More Features

Tensor Multiplication



```
int main () {
    let A = [2., 34.34];
    let B = [[3., 32.], [2., 43.]];
    let C = [[[32., 23.],[32.3, 3.]],[[0.0, 23.33], [2., 323.33]]];
    let D = C_{a, b, c}*C_{c, d, e};
    let E = C_{a, b, c}*D_{c, d, e, f};


    print(A_{a}*B_{a, b});
    print(B_{a, b}*B_{b, c});
    print(E_{a, b, c, d, e}*A_{c});
    print(C_{a, b, c}*D_{b, c, d, e});


    return 0;

}
```

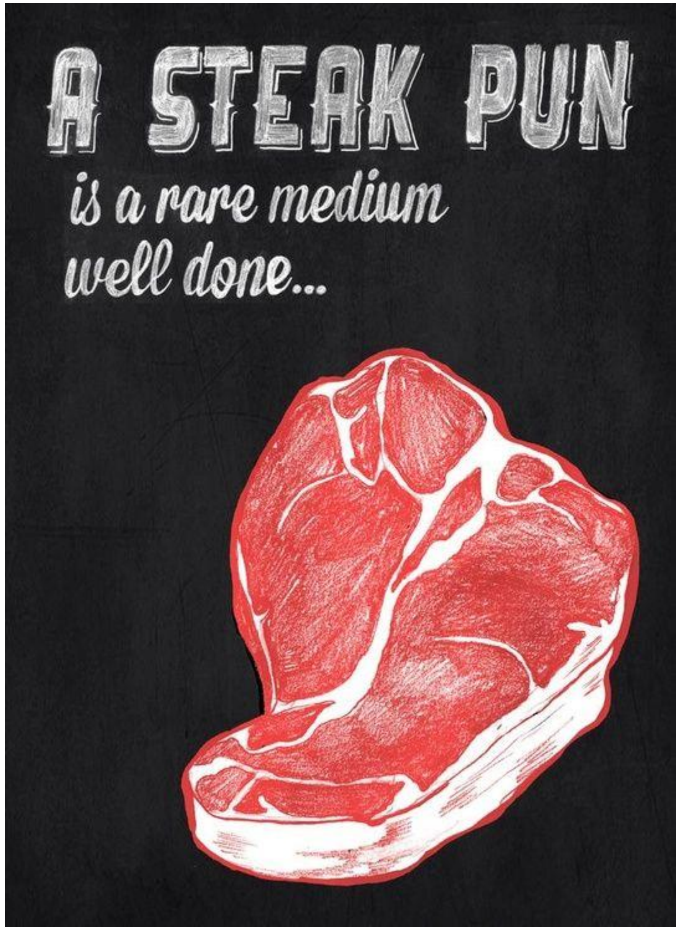
Tensor Slicing



```
int main () {
    let A = [2., 34.34];
    let B = [[3., 32.], [2., 43.]];
    let C = [[[32., 23.],[32.3, 3.]],[[0.0, 23.33], [2., 323.33]]];
    let D = C_{a, b, c}*C_{c, d, e};
    let E = C_{a, b, c}*D_{c, d, e, f};


    print(A_{0});
    print(B_{0, 1});
    print(D_{0, a, 1, 0});
    print(E_{0, 1, 0, 0, 0}*2.0);
    print(C_{1, 3/3, 2-1}*A_{1});


    return 0;
}
```

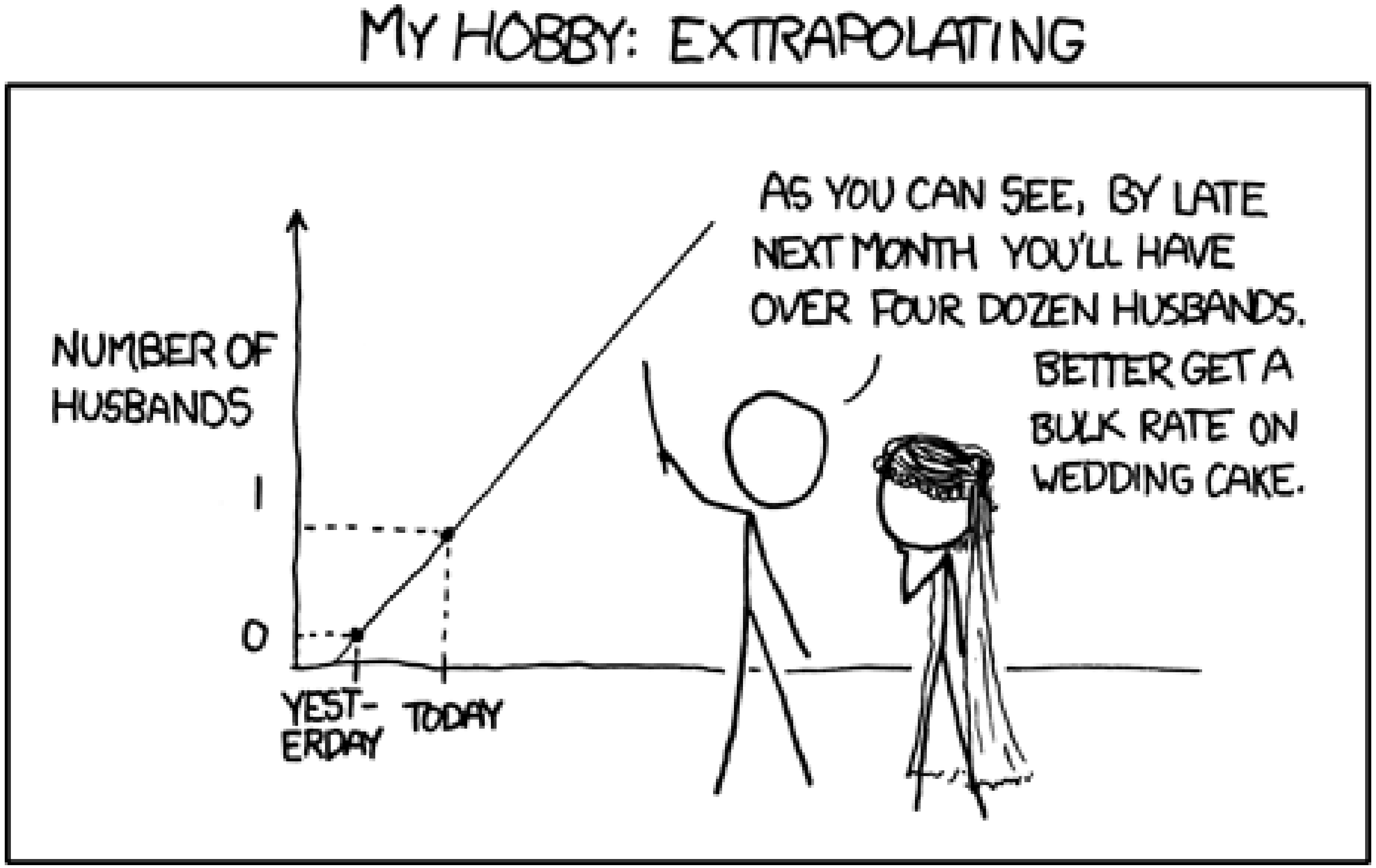
Little Language Demo

How to Run a Ranch, East Coast Style

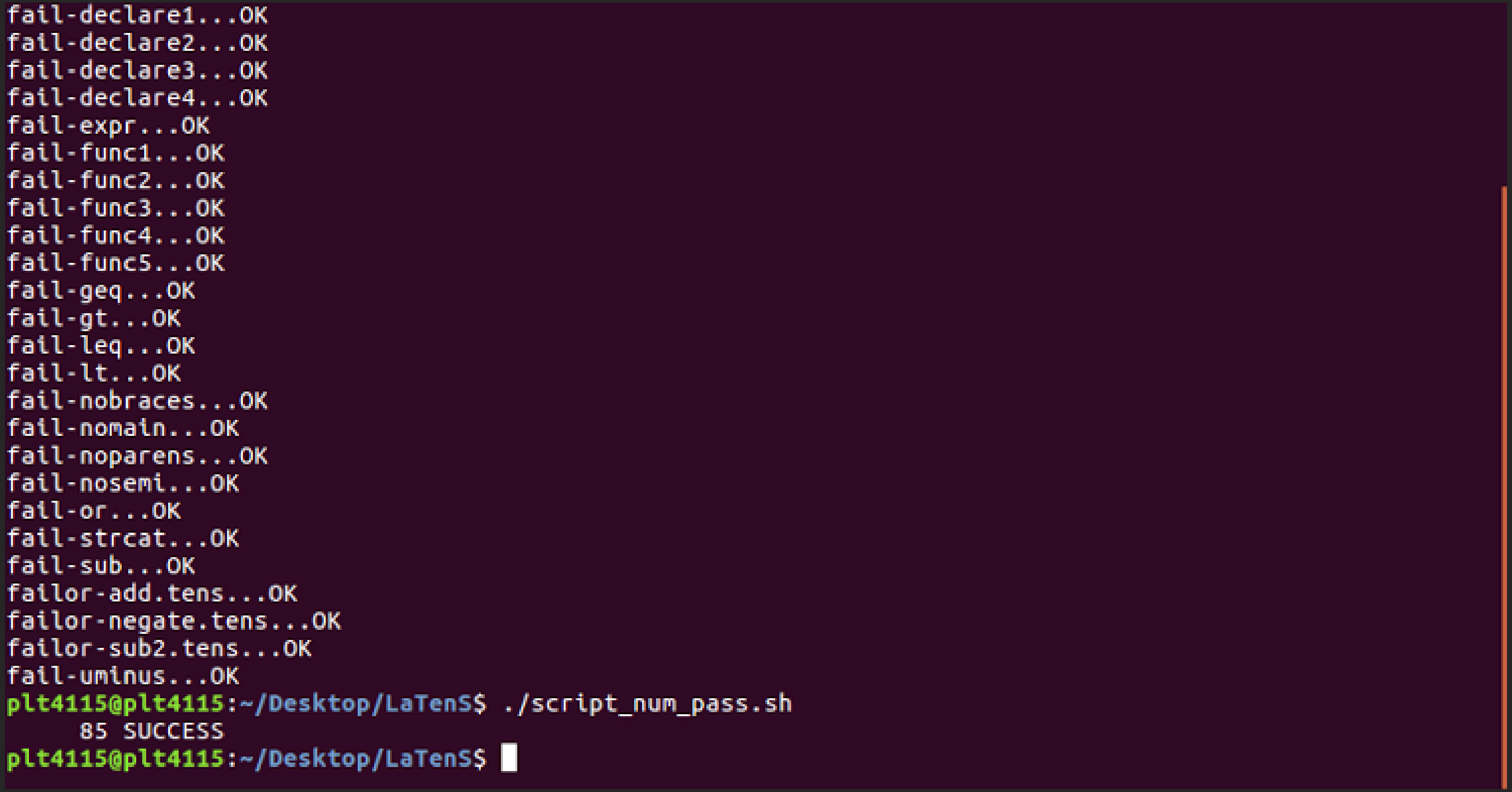


Time Dilation of GPS Satellites

Perceptron

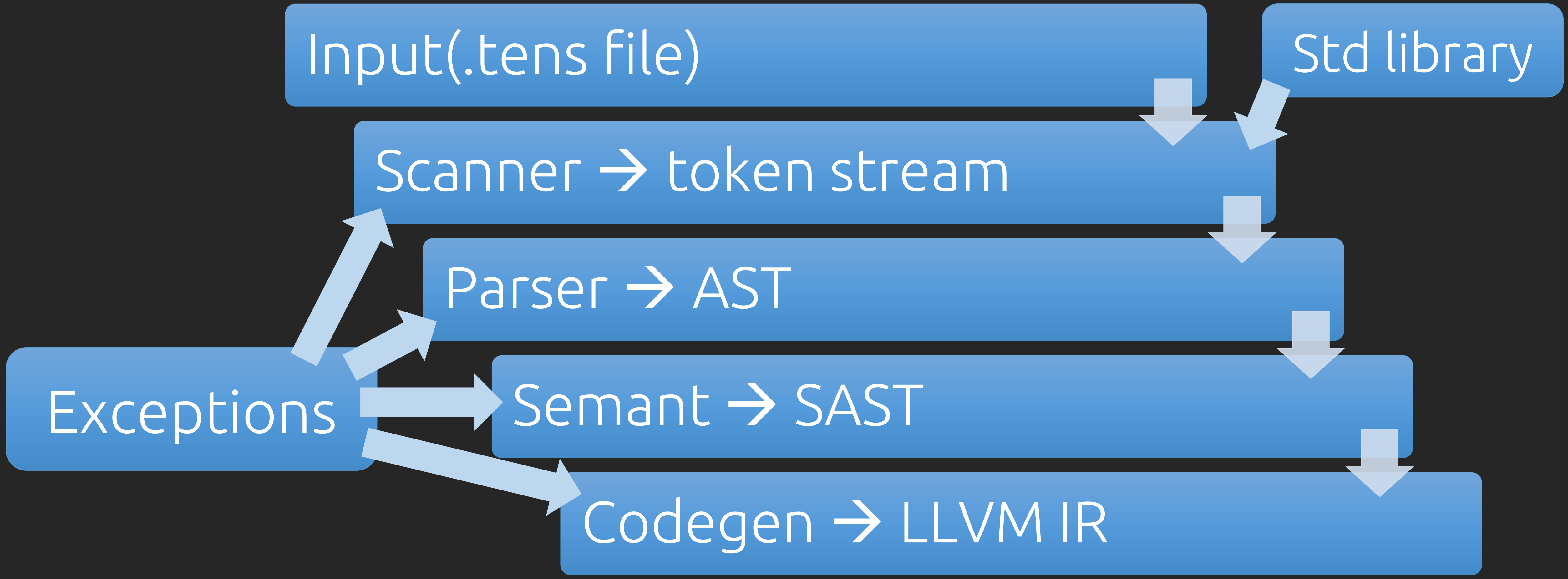

MY HOBBY: EXTRAPOLATING
NUMBER OF HUSBANDS
1
0
YEST-ERDAY
TODAY
AS YOU CAN SEE, BY LATE NEXT MONTH YOU'LL HAVE OVER FOUR DOZEN HUSBANDS.
BETTER GET A BULK RATE ON WEDDING CAKE.

Testing Suite



```
fail-declare1...OK
fail-declare2...OK
fail-declare3...OK
fail-declare4...OK
fail-expr...OK
fail-func1...OK
fail-func2...OK
fail-func3...OK
fail-func4...OK
fail-func5...OK
fail-geq...OK
fail-gt...OK
fail-leq...OK
fail-lt...OK
fail-nobraces...OK
fail-nomain...OK
fail-noparens...OK
fail-nosemi...OK
fail-or...OK
fail-strcat...OK
fail-sub...OK
failor-add.tens...OK
failor-negate.tens...OK
failor-sub2.tens...OK
fail-uminus...OK
plt4115@plt4115:~/Desktop/LaTenS$ ./script_num_pass.sh
     85 SUCCESS
plt4115@plt4115:~/Desktop/LaTenS$ 
```

Implementation: Overview



Input(.tens file)

Std library

Scanner → token stream

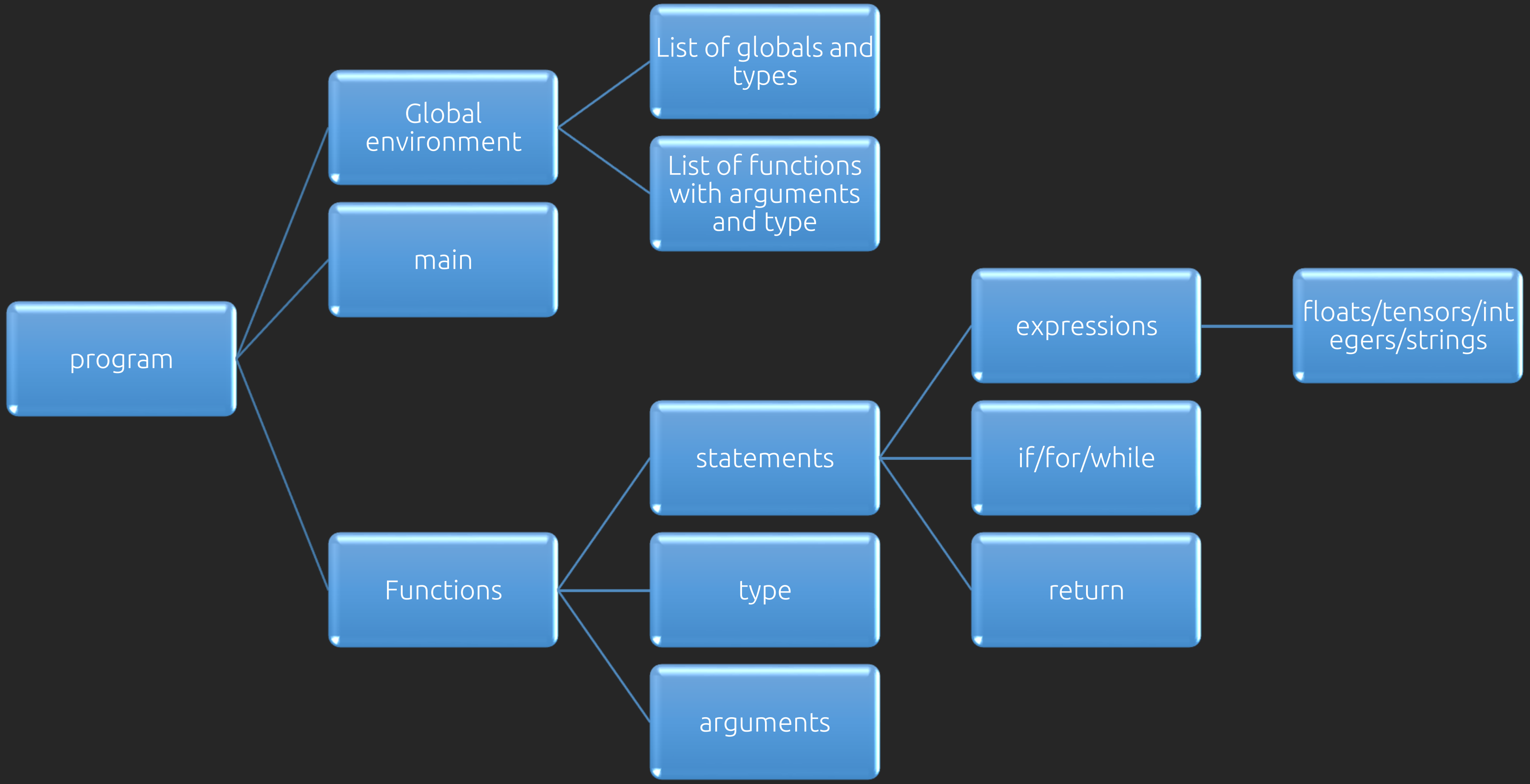
Parser → AST

Semant → SAST

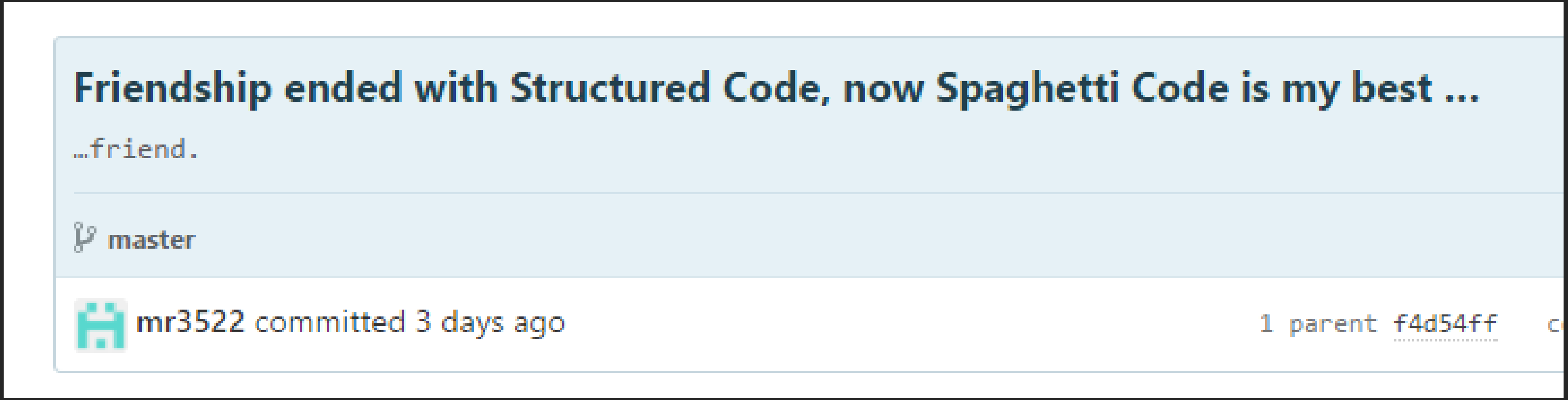
Codegen → LLVM IR

Exceptions

Implementation: SAST


program
Global environment
List of globals and types
List of functions with arguments and type
main
Functions
statements
expressions
floats/tensors/integers/strings
if/for/while
return
type
arguments

Codegen



Friendship ended with Structured Code, now Spaghetti Code is my best …

```
…friend.
```

master

mr3522 committed 3 days ago

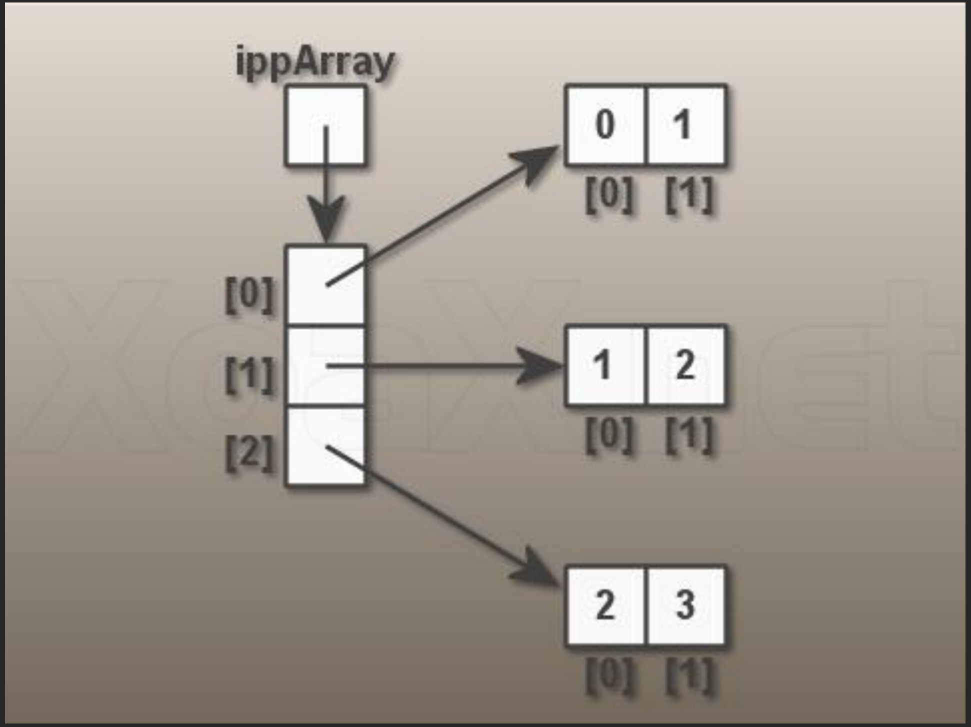
1 parent f4d54ff c

Challenges

- Expressions which are implicity a AST subtree
- Most of our functionality is code expansion
- Non-trivial as code expands to constructs which do not exist in our language
- Callback statement (Lllvm.llbuilder -> Llvm.llbuilder)
- Continuation Passing Style (yay!)
- Stack depth of 32! on a tensor multiplication

Tensor memory layout

- Nested arrays
- Why? Recursive heavy languages prefer recursive data structures
- Last level is a pointer to a float. Why? Array slicing made easy.



Lessons Learned

- .gitignore can be extremely valuable
- Start early, work in a group as much as possible, and never commit broken code
- Mohit is 26. Whaaaaaat?!?
- Different people code in different ways. Someone (the manager) should be figuring out people’s strengths and playing too them.
    - This gets especially interesting when half the group prefers to work hard but infrequently and the other half prefers frequently for short periods of time

Moving Forward

Some of the features we would like to implement, given the chance:

- Modify stdlib.tens to be an importable library
    - Modify our scanner to take any number of imports and include them as necessary
- Print directly into LaTeX
    - Have our LaTeX pretty-printer methods be callable within .tens files
- Variables, Functions, and Derivatives
    - Adding a data type and stdlib functions to play with it