C^* (aka Cstar)

Programming Language & Translators - Fall 2021

Joanne Wang

C* Currently

- Thank you for being extremely understanding
- Working in a team is harder than expected…
    - We didn’t have strong plan of action going in on how to implement all of our ideas for Cstar
    - Poor time management, poor communication as well as health issues caused our team to fall apart
- This version is a brand new Cstar compiler created by myself
- There are a lot of functionalities that are not yet implemented

I will be discussing what C* was originally supposed to be as well as the current architecture and the tests suites.

Overview of the Language

Key features of Cstar: (what it was supposed to be)

- General-purpose systems programming language
- Extremely explicit language - an expression oriented language
- Zero overhead costs -> highly optimized
- Postfix syntax
- Algebraic types such as structs and enums
- Pattern matching
- Monadic error handling
- Defer function
- Slices

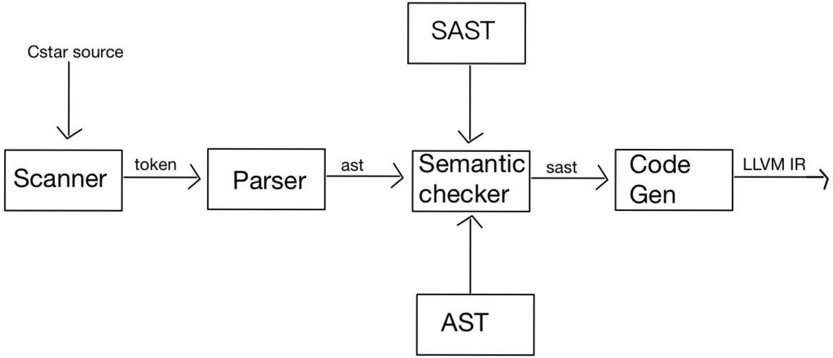
Architectural Design

Scanner
- takes in Cstar source program and generates tokens

Parser
- takes tokens from scanner and creates an AST

Semantic Checker
- traverses the AST and converts it to an SAST

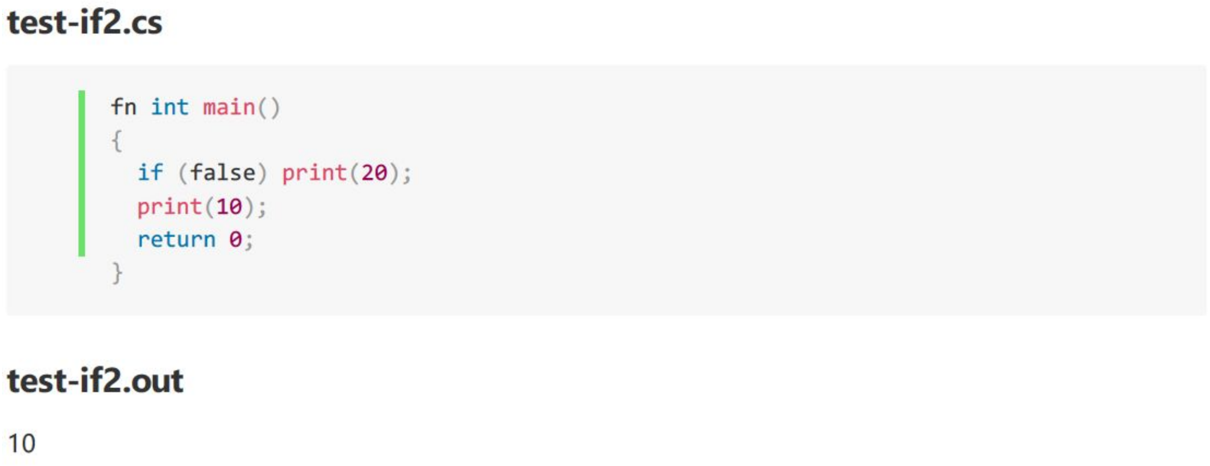
Code Generator
- takes the SAST and generates code for the LLVM IR



Test Plan

- Test Cstar functionalities and compares it to the expected output (success with a .out file or fail with a .err file)
- All tests are automated using a testall shell script

Example test:



test-if2.cs

```
fn int main()
{
  if (false) print(20);
  print(10);
  return 0;
}
```

test-if2.out

```
10
```