DeepLang
Parser结构设计

马昊天

2020-12-20

目 录
Contents

01 Deeplang前端

02 Parser设计

03 后续思考

1.Deeplang 前端

首先确定Deeplang语法，然后手动写递归下降分析器

1.Deeplang 前端

除了完成语法树的建立.....

- 错误诊断和处理
- 代码静态分析
- IR级优化
- 源码级工具

```
$ gcc-4.9 -fsyntax-only t.c
t.c: In function 'int f(int, int)':
t.c:7:39: error: invalid operands to binary + (have 'int' and 'struct A')
   return y + func(y ? ((SomeA.X + 40) + SomeA) / 42 + SomeA.X : SomeA.X);
                                       ^
$ clang -fsyntax-only t.c
t.c:7:39: error: invalid operands to binary expression ('int' and 'struct A')
  return y + func(y ? ((SomeA.X + 40) + SomeA) / 42 + SomeA.X : SomeA.X);
                       ~~~~~~~~~~~~~~ ^ ~~~~~
```

2.Parser设计之

Clang

主要介绍Clang前端的建树，错误处理

主要介绍Clang前端的建树，错误处理

主要介绍Clang前端的建树，错误处理

2.Parser设计之

Rust

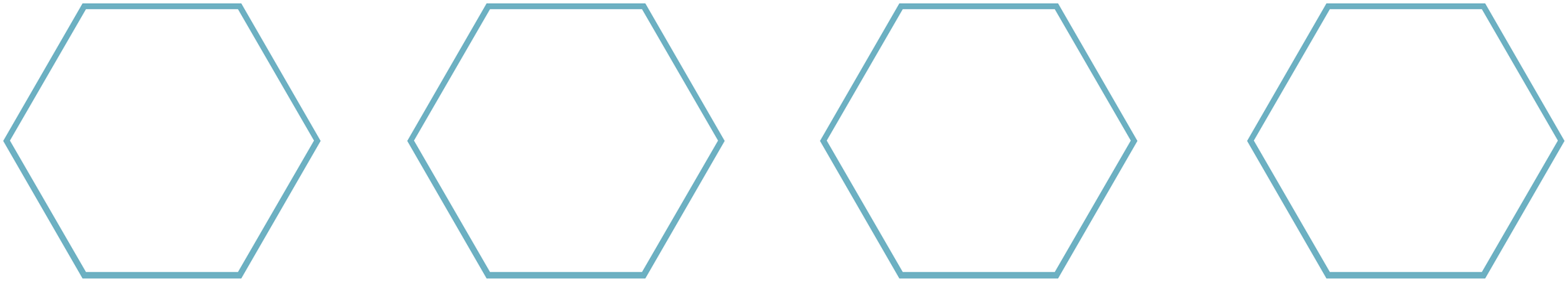
主要介绍Rust前端的建树，错误处理

主要介绍Rust前端的建树，错误处理

主要介绍Rust前端的建树，错误处理

3.后续思考

Clang is an "LLVM native" C/C++/Objective-C compiler, which aims to deliver amazingly fast compiles, extremely useful error and warning messages and to provide a platform for building great source level tools. The Clang Static Analyzer and clang-tidy are tools that automatically find bugs in your code, and are great examples of the sort of tools that can be built using the Clang frontend as a library to parse C/C++ code.



Thanks

谢谢观看

2020-12-20