P2. Semantic Analysis

Kelley van Evert
s4046854

Not complete

Far from! D: D:

Splb

- Splb = “Simply C, yeah kind of boring...”
  - i.e.
    - first-order functions at the toplevel
    - lexical scope, by block (function body, if, while, { ... })
    - follow the standard C-like stack discipline
    - Basic/composite types? Don't care yet ;)
  - changed grammar/parser accordingly

- A stepping stone, actually I want to do the language thing
  - write the compiler in the language itself
  - Spla = “Simple yet awesome!”

Type Checking

- Hindley Milner (Algorithm W)
  - Completed for expressions

- Check:
  - Function and variable declarations
    - are first-order OK?
  - Function/operator application
    - happens in expression typing (completed)
  - Variable assignment
  - Return statements

Pragmatics

- Encompassing type checking in state monads
  - for fresh type variables
  - for stepwise gathering unification goals
    - general recursion requires circular references

```
inferExprType :: Env →        Expr → State Int (Type, Subst)
assignOK      :: Env →        Stmt → State Int Subst
returnOK      :: Env → Type → Stmt → State Int Subst
```

(Questions)