

Sprint 2



Laboratório de Compiladores - MC911

Grupo

(RA)

Nome

(155253)

Eric Krakauer

(155981)

José Pedro Nascimento

(156331)

Lucas Racoci

(156475)

Luiz Fernando Fonseca

(157055)

Rafael Gois

Análise Sintática - Lexer

- Lexer é capaz de identificar tokens para:
 - Definição de tipos de dados
 - Inteiro
 - Ponto Flutuante
 - String
 - Palavra especial where
 - Tokens para declaração de estrutura: {};
 - Tokens para acesso a estrutura: []

Análise Sintática - Parser

- Implementa AST para:
 - Interpretar entrada (.hs)
 - Gerar uma árvore de execução
- Retorna a árvore de execução para o cliente web
- É capaz de traduzir:
 - Expressões com where
 - Novos tipos: String, Float, Struct

Parser: Compilação vs Interpretação

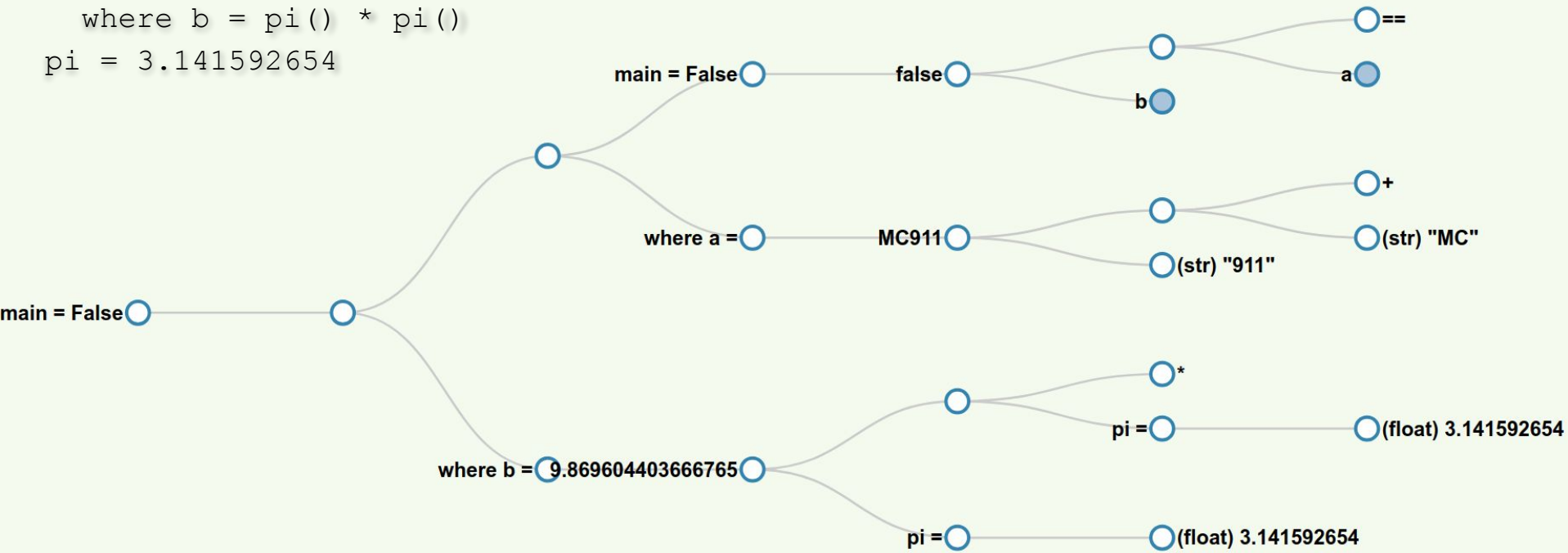
- Antes:
 - Era apenas gerada a árvore de execução do código
 - Mas nada era executado
- Agora,
 - Além de gerar a árvore de execução
 - Esta é também interpretada (executada)

Parser: Interpretação

- Utilização do Design Pattern Visitor
- Implementado em Python usando classes abstratas
- Implementa a “descida” na árvore executando o código
- Auxílio da Symbol Table:
 - Funções globais
 - Argumentos das funções
 - Variáveis do where

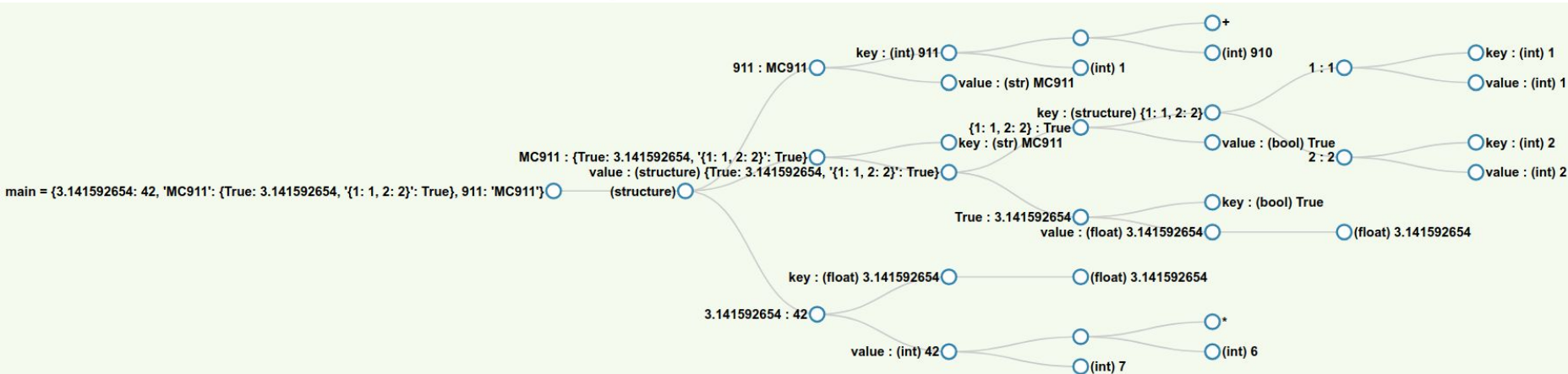
Parser: Novos tipos : String, Float

```
main = a == b
  where a = "MC" + "911"
        where b = pi() * pi()
pi = 3.141592654
```



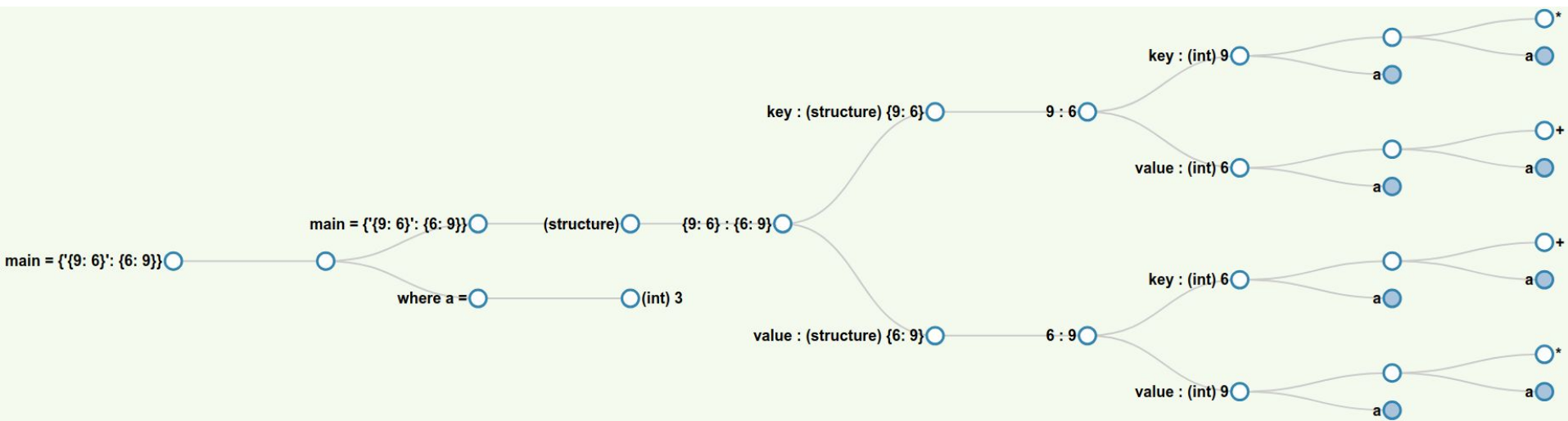
Parser: Novos tipos : Struct

```
pi = 3.141592654
main = {
  910 + 1 : "MC911",
  "MC911": {
    {1:1,2:2} : True,
    True: pi()
  },
  pi(): 6 * 7
}
```



Parser: Novos tipos : Struct

main = {{a*a:a+a}:{a+a:a*a} where a = 3



Análise Semântica

- O Backend agora será capaz de checar o tipo das variáveis e apontar operações inválidas.
 - Identificação do erro e imprimir mensagem no cliente web.

Cliente Web

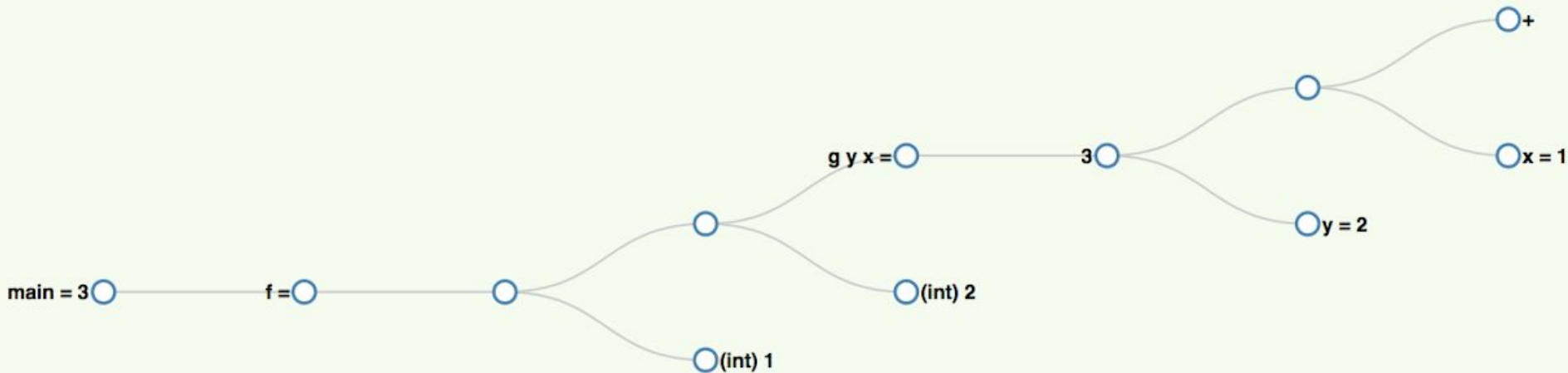
- O cliente web mostra a redução eta.
- Modificações do CSS, HTML e JavaScript usando as funções de D3.
 - Criação de botão para aplicar ou não a otimização visualmente (decisão do cliente).

Otimização

- Foi feita Redução Eta como otimização de código
- Redução Eta é quando há uma função que apenas chama outra, então nesses casos uma chamada pode ser economizada
- Aqui foi também utilizado o Design Pattern Visitor para descer em trechos de árvore para renomear variáveis

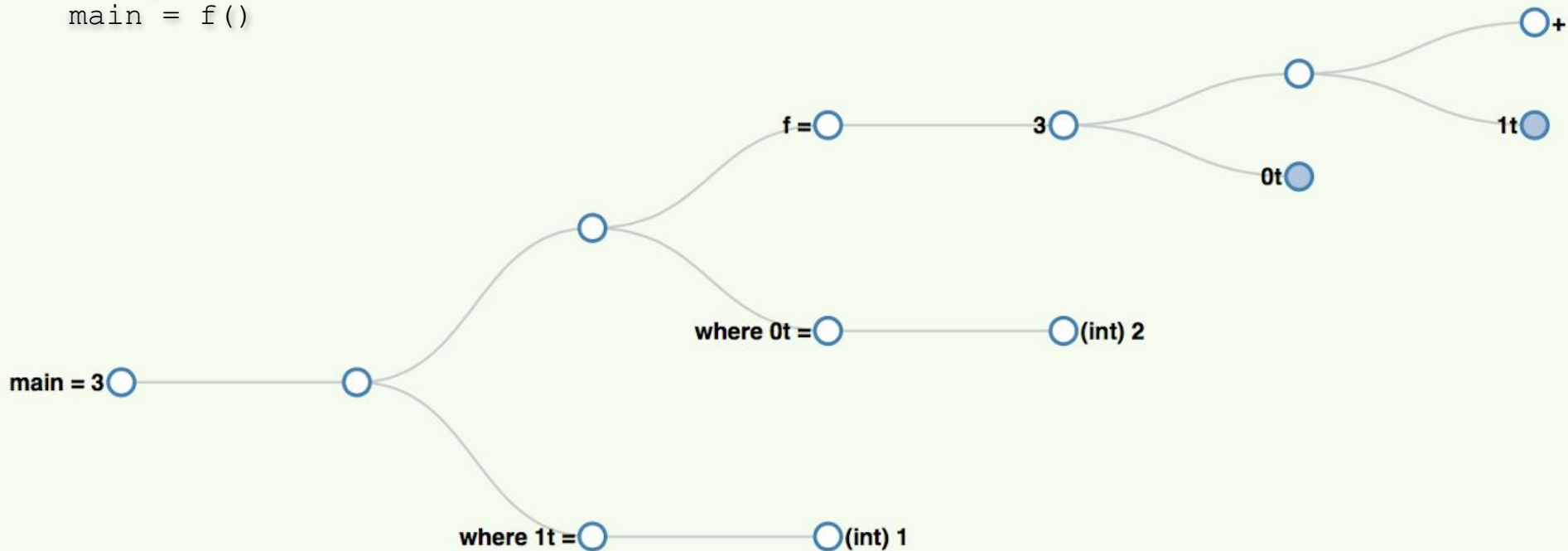
Otimização Eta (Desligada)

```
g y x = x + y  
f = g(2)(1)  
main = f()
```



Otimização Eta (Ligada)

```
g y x = x + y  
f = g(2)(1)  
main = f()
```



Docker

- Configurar mais testes para o docker.
 - Testar funcionalidades implementadas no Sprint 2:
 - Checagem de erros
 - Interpretação de novos tipos de dados
 - Where
 - Struct
 - Otimização

Próxima Sprint

- Declaração de listas e duplas
- Implementação de Pattern Matching
- Otimizações Constant Folding e Constant Propagation
- Adaptar cliente web para sintaxe de listas
- Configurar mais testes para o docker