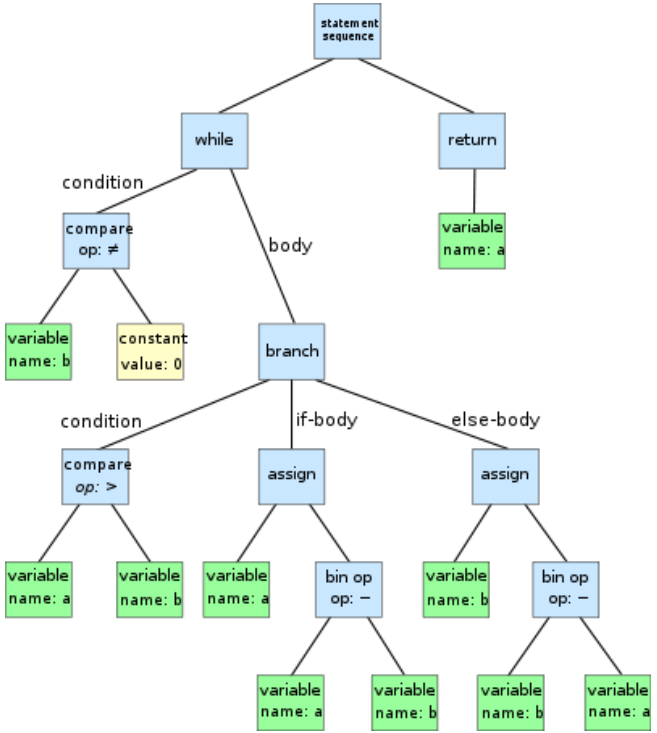
Монадические парсер-комбинаторы на C++

Выполнил: Сергеев Павел
Руководитель: Даниил Березун
Лаборатория языковых инструментов JetBrains

Зачем нужны парсеры?

```
while b ≠ 0
    if a > b
        a := a − b
    else
        b := b − a
return a
```

parser



Подходы к написанию парсеров:

- Описание грамматики и генерация кода (YACC, ANTLR, etc...)
- Рекурсивный спуск
- Парсер-комбинаторы

Монадические парсер-комбинаторы

```
type Parser a = String -> [(a,String)]
```

```
result :: a -> Parser a
result v = \inp -> [(v,inp)]
```

```
bind :: Parser a -> (a -> Parser b) -> Parser b
p ‘bind‘ f = \inp -> concat [f v inp’ | (v,inp’) <- p inp]
```

```
zero :: Parser a
zero = \inp -> []
```

```
item :: Parser Char
item = \inp -> case inp of
                        []     -> []
                        (x:xs) -> [(x,xs)]
```

Арифметические выражения

```
expr = term ‘chainl1‘ addop

term = factor ‘chainr1‘ mulop

factor = nat ++ bracket (char ’(’) expr (char ’)’)

addop = ops [(char ’+’, (+)), (char ’-’, (-))]

mulop = ops [(char ’*’, (*))]
```

В C++11 есть все

- Функции высшего порядка
- Параметрический полиморфизм
- Вывод типов

Существующие решения

- https://github.com/beark/ftl
- https://github.com/keean/Parser-Combinators

Что сделано?

- Реализован прототип C++ библиотеки для работы с парсер-комбинаторами
- Примеры использования
  - Арифметические выражения
  - Парсер json

Арифметические выражения

```
Parser<int> expr_p;


Parser<int> factor()
{
    return integer() || between(symbol('('), expr_p, symbol(')'));
}
Parser<int> term()
{
    return chainl(factor(), mul_op);
}
Parser<int> expr()
{
    return chainl(term(), (plus_op || minus_op));
}
```

Трудности

- Лямбды не имеют общего типа
- Нужно эмулировать ленивость
- Нужно руками замыкать параметры
- Предварительное объявление функций
- Ручное управление памятью

Полученные навыки

- Понимание подхода парсер-комбинаторов
- Навыки использования новых возможностей C++11
- Навыки написания ленивого кода на императивном языке

Репозиторий:

https://github.com/SergeevPavel/cpp_praser_combinators

Использованная литература:

Monadic Parser Combinators
Graham Hutton University of Nottingham
Erik Meijer University of Utrecht

Спасибо за внимание!