Апробация библиотеки Generic-transformers на примере компилятора языка Oberon0

автор: студент Озерцов Александр, гр 344

Научный руководитель: доцент кафедры системного программирования
кандидат физико-математических наук Булычев Дмитрий Юрьевич

Санкт-Петербург
2016

Семейство языков Oberon-0

- Oberon-0 императивный язык программирования, разработанный Н. Виртом;
- Член семейства языков, часть которых его подъязыки, а часть — надъязыки;
- Один язык семейства есть расширение другого.

Уровни языка

L0 Константы, переменные, атомарные типы, простые выражения и присваивания
L1 Конструкции IF и WHILE
L2 Конструкции FOR и CASE
L3 Процедуры
L4 Массивы и структуры
L5 Вложенные процедуры

Задачи компиляции

- Форматирование кода;
- проверка типов;
- разрешение имен;
- вычисление выражений;
- трансляция кода в язык C.

Expression problem

- Расширение функциональности требует создания новой функции;
- частичное изменение функции требует создания новой.

Существующие решения

- Полиморфные варианты;
- обобщенные функции;
- наследование классов и переопределение методов.

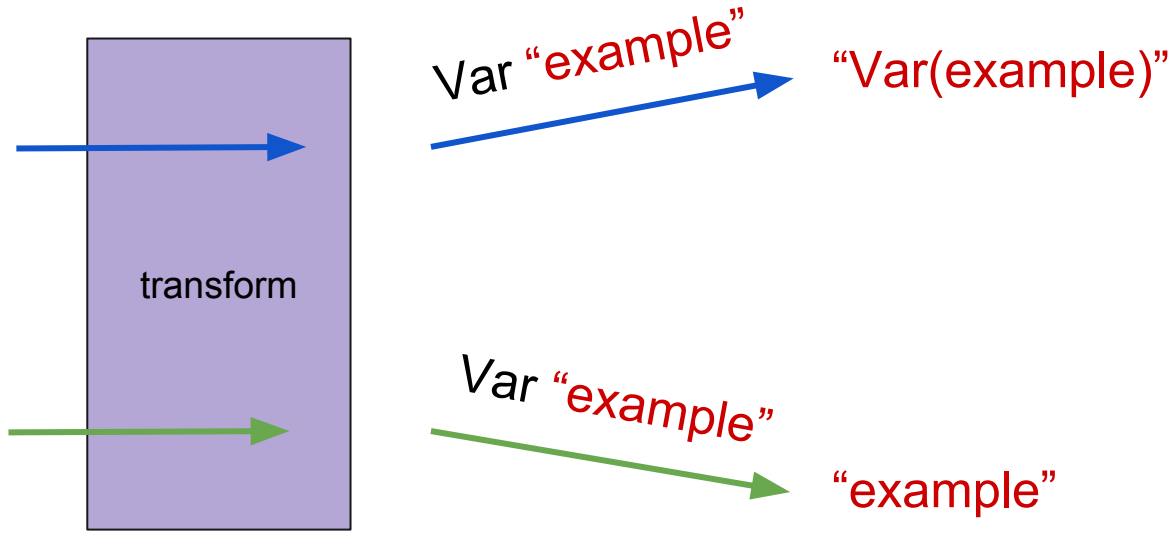
Объекты трансформаторы

- Объекты трансформаторы - классы, методы которых преобразуют определенный конструктор типа.

```
type t = Var of string

class show
  method Var s = ”Var (” s ”)”

class my_show : show
  method Var s = s
```



Generic-transformers

- Generic-transformers - библиотека, реализующая структуру для обобщенного программирования на языке OCaml;
- совмещает все известные способы переиспользования кода;
- генерирует функцию обхода на основе атрибутных грамматик.

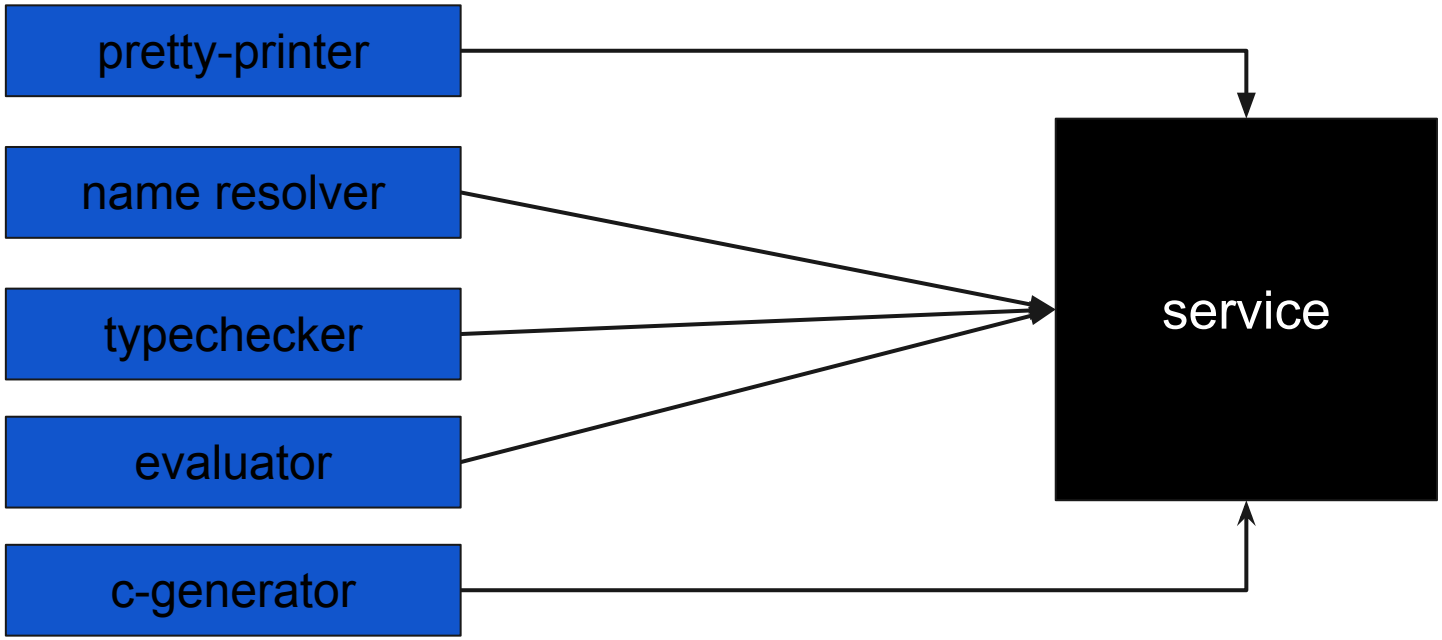
Постановка задачи

1. Изучить основные подходы обобщенного программирования. Изучить библиотеку Generic-transformers;
2. реализовать компилятор Oberon-0 с использованием библиотеки Generic-transformers;
3. сравнить полученный результат с предыдущей реализацией.

Ход работы

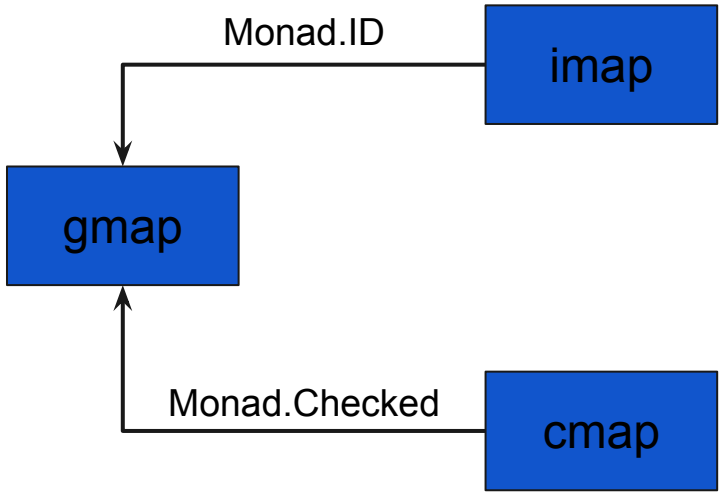
Определение типов:

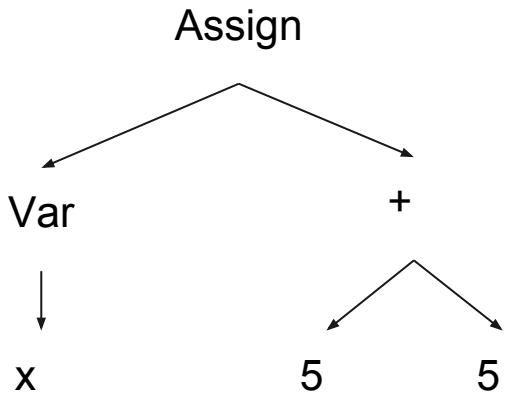
- Разбор работы парсера;
- связь с предыдущей реализацией;
- использование неполиморфных вариантов;



Порядок обхода деревьев

- Избавление от промежуточных слоев;
- Определение атрибутов.





Наличие лишь одной функции обхода вынуждает создавать лишние обертки над типами данных.

Сначала сложить, а потом искать переменную или наоборот?

Проблемы

- Функция gmap требует GT-обертки над типами;
- Расширение типа требует неявного переопределения всех методов объектов-трансформеров.

```
type ‘a expr = Const of int | Unop of [Neg | Not] * ‘a

type ‘b var =  Var of ‘b

type (‘a, ‘b) expr = [‘a expr | ‘b var]
```

```
class show
  method Const self s = ”Const (” f self ”)”
  method Unop a b = match a with
    | Neg -> “-” b
    | Not -> “!” b

class show_expr : show
  method Var self s = “Var” f self
```

Если s имеет тип ‘b var, а f ожидает (‘a, ‘b) expr - ошибка
=> нужно неявно переопределять метод

Результаты

1. Изучены основные подходы обобщенного программирования. Изучена библиотека Generic-transformers;
2. реализован L1 уровень компилятора Oberon-0 с использованием библиотеки Generic-transformers;
3. найдены ошибки в некоторых функциях библиотеки Generic-transformers.