



Библиотека обобщенной обработки данных для Ocaml

Мечтаев Сергей Владимирович
Научный руководитель: к.ф.-м.н. Булычев Д.Ю.
Рецензент: Сергей И.Д.

Санкт-Петербургский государственный университет
Математико-Механический факультет
Кафедра системного программирования

27 мая 2011 г.

Введение

Библиотека обобщенного управляемого типом программирования (generic type-driven programming) для языка Objective Caml.

Компилятор языка HaSCoL:

- реализован на языке Ocaml;
- много однотипных алгоритмов обработки данных.

Проблема: большой объем стереотипного кода.

Стереотипный (англ. boilerplate) код — шаблонный, вспомогательный.

Пример

Арифметическое выражение:

```
type expr =  
  Add of expr * expr  
| Sub of expr * expr  
| Neg of expr  
| Var of variable
```

Пример

Идентификация переменных:

```
let rec resolve var = function  
| Add (x, y) → Add (resolve var x, resolve var y)  
| Sub (x, y) → Sub (resolve var x, resolve var y)  
| Neg e      → Neg (resolve var e)  
| Var v      → Var (var v)
```

Сбор переменных:

```
let rec collect = function  
| Add (x, y) → (collect x) @ (collect y)  
| Sub (x, y) → (collect x) @ (collect y)  
| Neg e      → collect e  
| Var v      → [v]
```

Проблема

Стереотипный код:

- уменьшает гибкость;
 - ухудшает выразительность;
 - способствует появлению ошибок;
- ⇒ усложняет реализацию и сопровождение.

Поиск решения

Осaml

Для языка Осaml нет подходящих средств.

Другие языки

Подход “Scrap Your Boilerplate” (SYB) для языка Haskell.
Реализует концепцию generic (type-driven) programming.

- + Решает описанные проблемы.
- Невозможно использовать в Осaml.
- Низкая производительность.

Цель

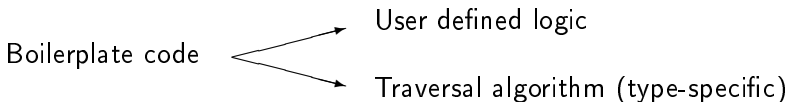
Generic (type-driven) library (адаптация SYB)

Требования:

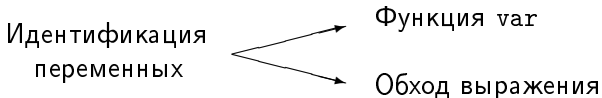
- создание трансформаций и запросов;
- применимость к задачам компилятора;
- типобезопасность;
- поддержка монад;
- потеря производительности менее чем в 2 раза.

Generic Type-Driven Programming (GTDP)

Идея



Пример



Особенности SYB в контексте GTDP

Lifting

Доопределение пользовательской функции с помощью безопасного приведения типов (сравнения типов на равенство во время исполнения).

Пример

Функции `var` доопределяется тождественно

$$(variable \rightarrow variable) \rightsquigarrow \forall \alpha. (\alpha \rightarrow \alpha)$$

с помощью сравнения типов

$$lifted_var\ x = \begin{cases} var\ x & \text{если } x \text{ типа } variable \\ x & \text{если } x \text{ другого типа} \end{cases}$$

Адаптация SYB для Ocaml

Маркеры типов

Маркеры — это пары (*type, identifier*). Равенство маркеров определяется равенством их идентификаторов.

Lifting

Сравнение маркеров вместо сравнения типов.

Специализация и CPS

User
Defined
Logic

Traversal
Algorithm

Specialization
Continuations

Type-
Specific

Syntax Extention

Library

Пример

Идентификация переменных:

```
let resolve var =  
  let module T = Make_transform (Id_monad) in  
  let tr = T.extend (T.return ()) variable_marker var in  
  (tr.specializeT tr expr_data) id
```

Апробация

Разработанное средство будет использоваться при реализации следующей версии компилятора языка NaSCoL.

Проверка на компиляторе простого языка:

- код идентификации переменных был уменьшен в 10 раз;
- код проверки типов был существенно упрощен.

Результаты

- На основе SYB разработан типобезопасный подход к обработке данных.
- Создана поддерживающая его библиотека.
- Показана применимость к задачам компилятора (идентификация переменных и проверка типов).