Языки программирования. Семантика и система типов

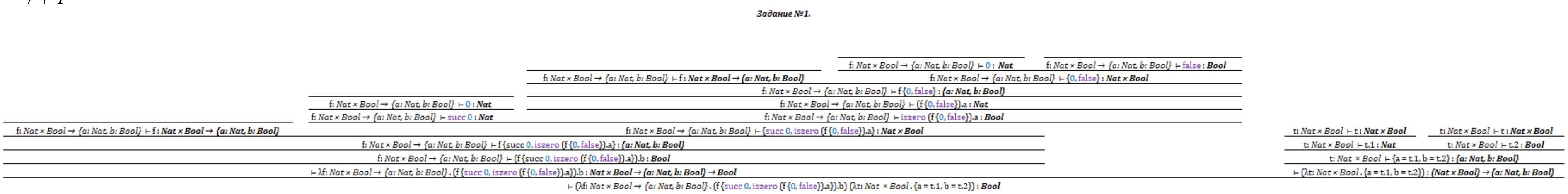
Теоретическое задание. Тема 3

Бронников Егор

Задание 1. $(\lambda f.\,(f\,\{\mathrm{succ}\,0,\ \mathrm{iszero}\,(f\,\{0,\ \mathrm{false}\}).a\}).b)\,(\lambda t.\,\{a = t.1,\ b = t.2\})$

1. Дополнение пропущенных аннотаций.
$(\lambda f : \boldsymbol{Nat} \times \boldsymbol{Bool} \to \boldsymbol{\{a : Nat,\ b : Bool\}}.\,(f\,\{\mathrm{succ}\,0,\ \mathrm{iszero}\,(f\,\{0,\ \mathrm{false}\}).a\}).b)\,(\lambda t : \boldsymbol{Nat} \times \boldsymbol{Bool}.\,\{a = t.1,\ b = t.2\}) : \boldsymbol{Bool}$

2. Дерево вывода типа.



Примечание. Рекомендуется изучить исходный Excel-документ `source.xlsx`.

Задание 2.

Условие. В нетипизированном λ-исчисления, пары термов могут быть представлены при помощи кодировки Чёрча. Можно ли использовать это представление, чтобы представить пары как производную форму поверх простого типизированного λ-исчисления с логическими и арифметическими выражениями?

(a) Выпишите функцию раскрытия сокращений, соответствующих такому определению пар. Должны быть явно представлены раскрытия паро ($\{t_1,\ t_2\}$), проекции ($t.1,\ t.2$), и типа-произведения ($T_1 \times T_2$).

(b) Покажите, что функция раскрытия сокращений сохраняет вычисление и типизацию, если возможно. Иначе – продемонстрируйте на контрпримере, почему сохранение вычисления или типизации невозможно.

Смотреть продолжение на следующей странице.

Решение.

a. Функция раскрытия сокращения.

Пусть пара $\{t_1,\, t_2\}$ кодируется как $\lambda x\, y.\, x\, t_1\, t_2$, тогда функция раскрытия сокращений будет иметь следующий вид:

- Пара

  $pair\, t_1\, t_2 = \lambda x\, y.\, x\, t_1\, t_2$

- Первая проекция

  $first\, p = p\, (\lambda x.\, \lambda y.\, x)$

- Вторая проекция

  $second\, p = p\, (\lambda x.\, \lambda y.\, x)$

Исходя из этого, раскрытие можно представить следующим образом:

$\{t_1,\, t_2\} := pair\, t_1\, t_2$

$t.1 := first\, t$

$t.2 := second\, t$

b. Сохранение вычисления и типизации

Сохранение вычисления. Функция раскрытия сокращения должна преобразовывать термы таким образом, что результат вычисления преобразованного терма будет эквивалентен результату исходного вычисления. Так как кодировка Чёрча предоставляет точный механизм для представления пар и операций над ними, то вычисление будут сохраняться.

Сохранение типизации. При раскрытии сокращений типы термов должен быть сопоставимы с типами в исходном выражении. В типизированном λ-исчислении типы пар должны соответствовать ожидаемому обобщённому типу произведения. Если $t : T$ в исходной системе, то $t' : T'$ в преобразованной системе, таким образом, что T соответствует T'. Однако в случае противоречия, когда типы не совпадают или не могут быть выведены, типизация термов не будет сохраняться.

Рассмотрим пары в типизированном λ-исчислении:

$$pair\, t_1\, t_2 : T_1 \to T_2 \to (T_1 \to T_2 \to R) \to R$$

В процессе раскрытия сокращений $pair\, t_1\, t_2$ и применении функции $first$ или $second$, результатом будет t_1 или t_2. Это демонстрирует, что преобразованный терм сохраняет вычисление исходного терма.

Проблема сохранения типизации может возникать, потому что типизированное λ-исчисление накладывает ограничение на формы типов, которые может использовать в термах. Для пар $\{t_1,\, t_2\}$ тип каждого компонента t_1 и t_2 должен быть известен и типизирован отдельно, тогда как в нетипизированном λ-исчислении не существует данного ограничения.

Контрпример.
Пусть мы знаем, что $t_1 : T_1$ и $t_2 : T_2$. Используем $pair\, t_1\, t_2$, тогда будет создан терм типа $\lambda x\, y.\, x\, t_1\, t_2$, где x и y являются абстракциями переменных с типами T_1 и T_2, а T_1 и T_2 являются типами t_1 и t_2. Тогда тип данного терма будет неоднозначным, T_1 или T_2. Следовательно, это является нарушением правил типизации в контексте типизированного λ-исчисления.

Ответ. Нельзя использовать кодировку Чёрча.
