



Языки программирования. Семантика и система типов  
Теоретическое задание. Тема 4

Бронников Егор

---

**Задание 1.** Выпишите функцию раскрытия сокращений, соответствующую такому варианту цикла **for**, для простого типизированного  $\lambda$ -исчисления с логическими выражениями, списками и комбинатором неподвижной точки (исключая **letrec**!).

*Решение.*

*Функция раскрытия сокращений.*

```
for ::=
  fix (
     $\lambda for : T \rightarrow (T \rightarrow Bool) \rightarrow (T \rightarrow T) \rightarrow (T \rightarrow Unit) \rightarrow List[T].$ 
     $\lambda t_1 : T. \lambda t_2 : T \rightarrow Bool. \lambda t_3 : T \rightarrow T. \lambda t_4 : T \rightarrow Unit.$ 
      if  $t_2 t_1$  then
         $cons[T] t_1 ((\lambda x : Unit. t_4 t_1) (for (t_3 t_1) t_2 t_3 t_4))$ 
      else
         $nil[T]$ 
    )
```

---

**Задание 2.** Покажите, что функция раскрытия сокращений сохраняет типизацию, если возможно. Иначе – продемонстрируйте на контрпримере, почему сохранение типизации невозможно.

*Решение.*

Для демонстрации сохранения типизации достаточно показать:

- (a) Сохранение типизации для аргументов  $t_1, t_2, t_3, t_4$ ;
- (b) Сохранение типизации для  $for$ .

(a) Сохранение типизации для аргументов  $t_1, t_2, t_3, t_4$ .

Терм  $t_1$  сохраняет типизацию по правилу типизации  $T\text{-}Var$ .

Терм  $t_2$  сохраняет типизацию по правилу типизации  $T\text{-}Abs$ , поскольку возвращает значение, тип которого указан после стрелки и соответствует типу  $Bool$ .

Терм  $t_3$  сохраняет типизацию по правилу  $T\text{-}Abs$ , поскольку возвращает значение того же типа, который был дан в качестве аргумента.

Терм  $t_4$  сохраняет типизацию по правилу  $T\text{-}Abs$ , поскольку выражение будет типизировано как `Unit`.

(b) *Сохранение типизации для `for`.*

Терм `for` сохраняет типизацию за счёт правил типизации комбинатора неподвижной точки  $T\text{-}Fix$ , абстракции  $T\text{-}Abs$ , применения  $T\text{-}App$ , логических выражений  $T\text{-}If$ , списков  $T\text{-}Nil$  и  $T\text{-}Sec$ . В конечном итоге дерево на верхнем уровне будет сохранять типизацию за счёт правила типизации  $T\text{-}Var$ .

*Ответ.* Функция раскрытия сокращений для цикла `for` будет сохранять типизацию.

---

**Задание 3.** Можно ли убрать явную аннотацию типа `for[T]`? В любых выражениях или только в некоторых? Аргументируйте свой ответ.

*Ответ.* Явную аннотацию типа `for[T]` можно убрать только тогда, когда известен тип первого аргумента  $t_1$ . В выражениях, где первый аргумент  $t_1$  является хорошо типизированным. Явную аннотацию можно убрать, так как можем вывести тип `for` через тип первого аргумента.

---