

Домашние задания по курсу «Теория типов»

1. На вход вашей программе дается файл `task1.in`, содержащий лямбда-выражение в следующей грамматике:

$$\begin{aligned}\langle \text{Абстракция} \rangle &::= \backslash ' \langle \text{Переменная} \rangle ' . ' \langle \text{Абстракция} \rangle \mid \langle \text{Применение} \rangle \\ \langle \text{Применение} \rangle &::= \langle \text{Применение} \rangle \langle \text{Терм} \rangle \mid \langle \text{Терм} \rangle \\ \langle \text{Терм} \rangle &::= ' (' \langle \text{Абстракция} \rangle ') ' \mid \langle \text{Переменная} \rangle \\ \langle \text{Переменная} \rangle &::= (' a ' \dots ' z ') \{ ' 0 ' \dots ' 9 ' \}^* \{ ' _ ' \}^*\end{aligned}$$

Аргументы в применении должны разделяться пробелом, если оба аргумента - переменные. Любые пробелы между нетерминальными символами (кроме пробела, разделяющего аргументы в применении), — а также начальные и конечные пробелы в строке — должны игнорироваться. Символы табуляции, возврата каретки и перевода строки должны трактоваться как пробелы.

Требуется расставить все недостающие скобки вокруг всех абстракций и применений, и напечатать получившийся результат в файле `task1.out`.

Например:

```
'\a.\b.a b c (\d.e \f.g) h'
должно превратиться в
'\a.(\b.(((a b) c) (\d.(e (\f.g)))) h))'
```

2. В файле `task2.in` задано некоторое лямбда-выражение, требуется найти список свободных переменных в нем и напечатать в алфавитном порядке, по идентификатору на строке.
3. В файле `task3.in` задана подстановка в некоторое лямбда-выражение в следующем синтаксисе (расширение определения из первой задачи):

$$\langle \text{Условие} \rangle ::= \langle \text{Абстракция} \rangle [' \langle \text{Переменная} \rangle ' := ' \langle \text{Абстракция} \rangle$$

Требуется в выходном файле `task3.out` привести результат подстановки.

4. В файле `task4.in` дано лямбда-выражение, требуется преобразовать его в нормальную форму и результат записать в файл `task4.out`. Если выражение не имеет нормальной формы, результат работы программы может быть произвольным.
5. Преобразовать лямбда-выражение в комбинаторное выражение в базисе *SKI*.

На вход в файле `task5.in` задано лямбда-выражение, выведите в файл `task5.out` бета-эквивалентное ему выражение в базисе *SKI*. Результирующее выражение должно соответствовать грамматике для лямбда-выражений, но не должно содержать лямбда-абстракций. В качестве значений можно использовать только большие буквы 'S', 'K' и 'I' и свободные переменные из исходного выражения.

6. Унификация термов. На вход в файле `task6.in` задан список уравнений в алгебраических термах, по уравнению на строке. Каждое уравнение соответствует следующей грамматике:

$$\begin{aligned}\langle \text{Уравнение} \rangle &::= \langle \text{Терм} \rangle = ' \langle \text{Терм} \rangle \\ \langle \text{Терм} \rangle &::= \langle \text{Функция} \rangle ' (' \langle \text{Терм} \rangle (' , ' \text{Терм})^* ') ' \mid \langle \text{Переменная} \rangle \\ \langle \text{Функция} \rangle &::= (' a ' \dots ' h ') \{ ' 0 ' \dots ' 9 ' \}^* \{ ' _ ' \}^* \\ \langle \text{Переменная} \rangle &::= (' i ' \dots ' z ') rep ' 0 ' \dots ' 9 ' * \{ ' _ ' \}^*\end{aligned}$$

Решите эту систему уравнений и выведите в файл task6.out наиболее общую подстановку, по строке на каждую переменную, используя следующую грамматику:

$$\langle \text{Строка подстановки} \rangle ::= \langle \text{Переменная} \rangle '=' \langle \text{Терм} \rangle$$

7. Просто типизированное лямбда-исчисление. На вход в файле task7.in задано лямбда-выражение. Выведите в файл task7.out какой-нибудь наиболее общий тип для этого выражения в просто типизированном лямбда-исчислении (если этот тип существует), или укажите, что выражение типа не имеет.

Результат должен соответствовать следующей грамматике:

$$\begin{aligned} \langle \text{Ответ} \rangle &::= \langle \text{Тип} \rangle \mid \text{'Лямбда-выражение не имеет типа.'} \\ \langle \text{Тип} \rangle &::= \langle \text{Перем} \rangle \mid \langle \text{Перем} \rangle '->' \langle \text{Тип} \rangle \mid '(' \langle \text{Тип} \rangle ')' \\ \langle \text{Перем} \rangle &::= ('a' \dots 'z') \{ '0' \dots '9' \}^* \{ '_ ' \}^* \end{aligned}$$

8. Расширенное лямбда-исчисление. На вход программы в файле task12.in задается выражение в лямбда-исчислении с конструкцией let. Выражение соответствует следующей грамматике:

$$\begin{aligned} \langle \text{Выражение} \rangle &::= \text{'let'} \langle \text{Переменная} \rangle '=' \langle \text{Выражение} \rangle \text{'in'} \langle \text{Выражение} \rangle \\ \langle \text{Абстракция} \rangle &::= \text{'.'} \langle \text{Переменная} \rangle \text{'.'} \langle \text{Абстракция} \rangle \mid \langle \text{Применение} \rangle \\ \langle \text{Применение} \rangle &::= \langle \text{Применение} \rangle \langle \text{Терм} \rangle \mid \langle \text{Терм} \rangle \\ \langle \text{Терм} \rangle &::= '(' \langle \text{Абстракция} \rangle ')' \mid \langle \text{Значение} \rangle \\ \langle \text{Значение} \rangle &::= ('a' \dots 'z') \{ '0' \dots '9' \}^* \{ '_ ' \}^* \\ &\quad \mid \text{'F'} \mid \text{'T'} \mid \text{'If'} \mid \text{'Y'} \mid \text{'Plus'} \mid ('0' \dots '9')^+ \\ &\quad \mid \text{'<'} \langle \text{Значение} \rangle \text{'>'} \mid \text{'PrL'} \mid \text{'PrR'} \\ &\quad \mid \text{'InL'} \mid \text{'InR'} \mid \text{'Case'} \end{aligned}$$

Аналогично простым лямбда-выражениям, если подряд указаны какие-либо идентификаторы или ключевые слова let и in, то они будут разделяться пробелом. Любые другие пробелы — а также начальные и конечные пробелы в строке — должны игнорироваться. Символы табуляции, возврата каретки и перевода строки должны трактоваться как пробелы.

Требуется нормализовать выражение. Для переменных особого вида следует применять следующие правила редукции:

Исходное выражение	Бета-редукция выражения
'If T t e'	't'
'If F t e'	'e'
'Plus x y'	$(x + y) \% 2^{63}$, если x,y — числа, иначе — ошибка
'Y f'	'f (Y f)'
'PrL <a,b>'	'a'
'PrL <a,b>'	'b'
'Case (PrL x) l r'	'l x'
'Case (PrR y) l r'	'r y'

Частичное применение переменных особого вида допускается (в этом случае имеет место классический карринг). Если какое-то применение переменной особого вида отсутствует в списке (например, ‘If 0 1 2’) — то его редукция должна приводить к ошибке.

В качестве примеров к программе должны быть приложены выражения для вычисления:

- по числу n ($0 \leq n \leq 91$) вычислить n -е число Фибоначчи F_n (подсказка: $F_{91} = 7540113804746346429$)
- по числу n вернуть упорядоченную пару $\langle a, b \rangle$, где a — число цифр в двоичном разложении n (без ведущих нулей), а b — сумма цифр в двоичном разложении.

9. Алгоритм W .

На вход программе передается файл task13.in, содержащий расширенное лямбда-выражение. Требуется применить алгоритм W и выдать в выходной файл результирующий тип и контекст в следующей грамматике:

$$\begin{aligned} \langle \text{Ответ} \rangle &::= \langle \text{Тип} \rangle \backslash \mathbf{n} \langle \text{Контекст} \rangle \mid \text{‘Лямбда-выражение не имеет типа.’} \\ \langle \text{Контекст} \rangle &::= \{ \langle \text{Переменная} \rangle \text{ ‘=’ } \langle \text{Тип} \rangle \text{ ‘\mathbf{n}’} \}^* \\ \langle \text{Тип} \rangle &::= \langle \text{Тип} \rangle \mid \langle \text{Дизъюнкция} \rangle \text{ ‘->’ } \langle \text{Тип} \rangle \mid \text{‘(’ } \langle \text{Тип} \rangle \text{ ‘)’} \\ \langle \text{Дизъюнкция} \rangle &::= \langle \text{Дизъюнкция} \rangle \text{ ‘|’ } \langle \text{Конъюнкция} \rangle \mid \langle \text{Конъюнкция} \rangle \\ \langle \text{Конъюнкция} \rangle &::= \langle \text{Конъюнкция} \rangle \text{ ‘\&’ } \langle \text{Имя-типа} \rangle \mid \langle \text{Имя-типа} \rangle \\ \langle \text{Переменная} \rangle &::= \text{‘a’} \dots \text{‘z’} \{ \text{‘0’} \dots \text{‘9’} \}^* \{ \text{‘,’} \}^* \\ \langle \text{Имя-типа} \rangle &::= \langle \text{Переменная} \rangle \mid \text{‘Bool’} \mid \text{‘Int’} \end{aligned}$$

У выражений особого вида должны быть следующие особые типы:

Имя	Тип
T,F	$Bool$
If	$Bool \rightarrow a \rightarrow a \rightarrow a$
$0, 1, \dots, 2^{63} - 1$	Int
Inc	$Int \rightarrow Int$
$\langle x^a, y^b \rangle$	$a \& b$
PrL	$a \& b \rightarrow a$
PrR	$a \& b \rightarrow b$
InL	$a \rightarrow a b$
InR	$a \rightarrow a b$
Case	$a b \rightarrow (a \rightarrow c) \rightarrow (b \rightarrow c) \rightarrow c$
Y	$(a \rightarrow a) \rightarrow a$

Убедитесь, что решения для примеров из предыдущей задачи правильно типизируются.