1 Немного истории.

2 Введение в лямбда-исчисление: определения и базовые результаты.

Рассмотрим мотивирующий пример. Когда мы пишем, что «функция отображает аргумент x в M», где M — это метапеременная, в которой лежит тело функции, то мы используем следующую нотацию $x \mapsto M$, тогда как запись $\lambda x.M$ следует читать точно также в содержательном смысле. Расширим наш мотивирующий и не совсем формальный пример, заменив метапеременную M на более понятное арифметическое выражение : $x \mapsto x^2+6x+9$ и $\lambda x.x^2 + 6x + 9$.

Теперь же перейдем к формальным определениям. Базовое понятие в λ-исчислении — это *предтерм.* Предположим, у нас есть бесконечный алфавит:

$$\Lambda = v_0, v_1, v_2, v_3, ... \tag{1}$$

Предтермами мы будем называть конечные строки над алфавитом Λ, порожденные следующей грамматикой:

$$\Lambda_{term} ::= \Lambda \mid (\Lambda_{term}\Lambda_{term}) \mid \lambda\Lambda.\Lambda_{term} \tag{2}$$

Примеры конечных строк, порожденный заданной грамматикой:
1) $((v_3\ v_5)\ v_8)$;
2) $\lambda v_6.v_5 v_6$
3) $\lambda v_0.v_0$
4) $\lambda v_{05091995}.\lambda v_{38}.v_4$
Как мы видим из определения грамматики, предтермы бывают трех видов.

Зададим классификацию предтермов в соответсвии с грамматикой:

1) Предтермы первого вида (это просто элементы Λ) называются *переменными*, которые мы будем обозначать тремя предпоследними буквами латинского алфавита $x, y, z, ...$ (возможно с индексами);

3 Комбинаторная логика и ее связь с лямбда-исчислением.

4 Простое типизированное лямбда-исчисление: типизация по Карри и по Черчу.

5 Типизированные комбинаторы.

6 Практическая реализация лямбда-исчисления, комбинаторной логики и теории типов.