# Простые атрибутные типы (черновик 2)

Михайлов Анатолий Андреевич (601-41)

17 марта 2025 г.

**Аннотация**

В работе описывается система «$\lambda \to A$», основывающаяся на просто типизированном лямбда-исчислении, имеющая свойства, полезные в области формальных доказательств. Приведены примеры использования в доказательстве теорем в арифметике Пеано.

**Определения** Тип – последовательность функциональных атрибутов (ФА) и предикатных атрибутов (ПА), при этом в последовательности может не быть ФА или ПА, но длинна последовательности $\geq 1$. Синтаксис в расширенной форме Бэкуса-Наура:

$$
\begin{aligned}
A &::= ID \\
AL &::= A \text{ “,” } AL \mid A \\
F &::= (AL \to AL) \\
FL &::= F, FL \\
T &::= FL \text{ “,” } AL \mid FL \mid AL
\end{aligned}
$$

Примеры ПА: $a \quad b \quad c \quad Nat \quad Even \quad Odd$

Примеры ФА: $a \to b \quad c, d \to e, f, g \quad Nat, Even \to Nat, Odd$

**Основная идея** Выражение может иметь несколько типов одновременно. В таком случае, типы можно воспринимать как атрибуты этого выражения. Пример: утверждение «$a : A, B$» значит «выражение $a$ имеет атрибуты $A$ и $B$». Но, это еще не значит, что выражение $a$ не имеет атрибута $C$. Это значит лишь то, что для выражения $a$ не доказан атрибут $C$. Его можно получить, применив, например $a$ к $f : A \to C$.

**Проверка типов** Пусть дана функция $f : (A, B) \to C$. Тогда применение этой функции к аргументу $a$ считается возможным, если $a : A, B$, при этом $a$ может иметь любое количество дополнительных типов, главное чтобы были типы $A$ и $B$.

**Несколько типов функции** Эта особенность позволяет элегантно (по мнению автора) реализовать варианты функции. Отличный пример – определение функции $S$ в примерах с арифметикой Пеано. Механизм выбора варианта прост – выбирается самый левый тип функции, подходящий по атрибутам аргументу. Соответственно, слева нужно размещать самые частные случаи, а справа самые общие.

**Типы как утверждения** Эта интерпретация также применима к описываемому формализму. Пример: $x : Nat, Even$ означает, что $x$ – натуральное четное число. Переводя на язык логики предикатов – $Nat(x) \wedge Even(x)$. Каждое утверждение о типизации можно тривиально перевести на язык логики предикатов.

**Описание в системе натурального вывода** Этот параграф требует значительной доработки/переработки. Если a,b – термы, A,B,C,D – типы, то:

Абстракция:

$$\frac{a:\ A, ... \quad b = \lambda x:\ (B, ...).\, a}{b:\ (B, ...) \rightarrow (A, ...)}$$

Применение:

$$\frac{a:\ A, ... \quad b:\ (A \rightarrow (C, ...)), (B \rightarrow (C, D, ...))}{(ba):\ C, ...}$$

$$\frac{a:\ B, ... \quad b:\ (A \rightarrow (C, ...)), (B \rightarrow (C, D, ...))}{(ba):\ C, D, ...}$$

**Доказательство четности числа 4 в арифметике Пеано** Оно вдохновлялось доказательством в системе Twelf, но получилось значительно короче и дополнительно включает в себя определение нечетных чисел.
Аксиомы:

$$0:\ Nat,\ Even$$

$$S:\ (Nat, Even) \rightarrow (Nat, Odd), (Nat, Odd) \rightarrow (Nat, Even)$$

Построения:

$$1 = (S0) \qquad 2 = (S1) \qquad 3 = (S2) \qquad 4 = (S3)$$

Термы 1, 3 будут иметь атрибуты $(Nat, Odd)$
Термы 0, 2, 4 будут иметь атрибуты $(Nat, Even)$

**Доказательство четности/нечетности числа n+2 в арифметике Пеано** Аксиомы:

$$0 : Nat, \ Even$$

$$S : (Nat, Even) \to (Nat, Odd), (Nat, Odd) \to (Nat, Even)$$

Построения:

$$S' = \lambda n : Nat. (S(Sn))$$

$S'$ имеет атрибуты $(Nat, Even) \to (Nat, Even), (Nat, Odd) \to (Nat, Odd)$

**Расширение** Уверен, что существуют системы зависимых атрибутных типов, подобных тем, что есть в лямбда кубе, но я не могу осилить их описание и изучение с текущим уровнем знаний.

**О применении в языках программирования** Это изначальная цель разработки. Есть идеи о реализации языка программирования или системы автоматизированных доказательств (или того и другого, как в Coq), основывающегося на этом формализме. Есть опыт по реализации простых формализмов (типа $\lambda$-исчисления), но я понимаю, что мне не хватает знаний для чего-либо серьезного.