

Теоретико-категорная семантика модальной теории типов, основанной на интуиционистской эпистемической логике

1 Предварительные замечания и определения

1.1 Глоссарий по основным конструкциям функционального языка программирования Haskell: функторы, монады, аппликативные функторы

Определение 1. Класс типов

Классом типов в языке Haskell – это реализация некоторого общего интерфейса для совокупности типов.

Представителем (или наследником) класса типов называется реализация данного класса для конкретного типа.

Определение 2. Функтор

Функтор – это однопараметрический класс типов, позволяющий пронести действие функции через значения, полученные в результате применения к их типу одноместного типового оператора.

Определение в стандартной библиотеке выглядит следующим образом:

```
class Functor f where
  fmap :: (a -> b) -> f a -> f b
```

Рассмотрим примеры:

- Список (неограниченная в длине последовательность) является функтором: Определение в стандартной библиотеке выглядит следующим образом:

```
instance Functor [] where
  fmap :: (a -> b) -> [a] -> [b]
  fmap f [] = []
  fmap f (x:xs) = (f x) : (fmap f xs)
```

Данный пример достаточно прост: реализация функтора для списка – это функция высшего порядка, которая, принимая на входе одноместную функцию из типа a в тип b и список элементов типа a , возвращает список элементов типа b , который получен применением функции к каждому элементу списка, полученного на вход.

- Пара (тип декартова произведения типов) также функтор:

```
instance Functor (b, ) where
  fmap :: (a -> c) -> (b, a) -> (b, c)
  fmap f (x, y) = (x, f y)
```

Конструктор пары является двухпараметрическим типовым оператором, но мы сделали из него однопараметрический оператор фиксацией первого параметра.

Данная реализация также довольно проста: на вход принимается функция из типа a в тип c и кортеж, в котором первая координата имеет тип b , а вторая – тип a . На выходе мы получаем кортеж типа (b, c) , применяя полученную на вход функцию ко второй координате пары.

- Тип *Maybe* – это однопараметрический типовой оператор, для обработки неопределенных значений:

```
data Maybe a = Nothing | Just a
```

Реализация функтора для типа *Maybe*:

```
instance Functor Maybe where
  fmap :: (a -> b) -> Maybe a -> Maybe b
  fmap f Nothing = Nothing
  fmap f (Just x) = Just (f x)
```

Если второй аргумент является неопределенным значением (на вход передан *Nothing*), то и возвращается *Nothing*. Если же значение определено, то есть оно имеет вид *Just x*, тогда мы применяем функцию функцию к x , а результат вычисления оборачиваем в конструктор *Just*.

1.2 Приложение А. Глоссарий по теории категорий.

Определение 3. Категория \mathcal{C} состоит из:

- Класса объектов $Ob_{\mathcal{C}}$;
- Для любых объекта $A, B \in Ob_{\mathcal{C}}$ определено множество стрелок (или морфизмов) из A в B $Hom_{\mathcal{C}}(A, B)$;
- Если $f \in Hom_{\mathcal{C}}(A, B)$ и $g \in Hom_{\mathcal{C}}(B, C)$, то $g \circ f \in Hom_{\mathcal{C}}(A, C)$;
- Для любого объекта $A \in Ob_{\mathcal{C}}$, определен тождественный морфизм $id_A \in Hom_{\mathcal{C}}(A, A)$;
- Для любой стрелки $f \in Hom_{\mathcal{C}}(A, B)$, для любой стрелки $g \in Hom_{\mathcal{C}}(B, C)$ и для любой стрелки $h \in Hom_{\mathcal{C}}(C, D)$, $h \circ (g \circ f) = (h \circ g) \circ f$.
- Для любой стрелки $f \in Hom_{\mathcal{C}}(A, B)$, $f \circ id_A = f$ и $id_B \circ f = f$.

Определение 4. Функтор

Пусть \mathcal{C}, \mathcal{D} – категории. Функтором называется отображение $F : \mathcal{C} \rightarrow \mathcal{D}$, такое, что:

- $F : A \mapsto FA$, где $A \in Ob_{\mathcal{C}}$;
- $F(g \circ f) = F(g) \circ F(f)$;
- $F(id_A) = id_{FA}$.

Определение 5. Естественное преобразование Пусть $\mathcal{F}, \mathcal{G} : \mathcal{C} \rightarrow \mathcal{D}$ – функторы. Естественным преобразованием $\alpha : \mathcal{F} \Rightarrow \mathcal{G}$ называется такое индексированное семейство стрелок $(\alpha_X)_{X \in Ob_{\mathcal{C}}}$, что для любых $A, B \in Ob_{\mathcal{C}}$, для любой стрелки $f \in Hom_{\mathcal{C}}(A, B)$, диаграмма коммутует:

$$\begin{array}{ccc} \mathcal{F}A & \xrightarrow{\mathcal{F}(f)} & \mathcal{F}B \\ \alpha_A \downarrow & & \downarrow \alpha_B \\ \mathcal{G}A & \xrightarrow{\mathcal{G}(f)} & \mathcal{G}B \end{array}$$

Определение 6. Моноидальная категория

Моноидальная категория – это категория \mathcal{C} с дополнительной структурой:

- Бифунктор $\otimes : \mathcal{C} \times \mathcal{C} \rightarrow \mathcal{C}$, который мы будем называть тензором;
- Единица $\mathbb{1}$;
- Изоморфизм, который мы будем называть ассоциатором: для любых $A, B, C \in Ob_{\mathcal{C}}$, $\alpha_{A,B,C} : (A \otimes B) \otimes C \cong A \otimes (B \otimes C)$;
- Изоморфизм $L_A : \mathbb{1} \otimes A \cong A$;
- Изоморфизм $R_A : A \otimes \mathbb{1} \cong A$;

- Первое условие когерентности (пятиугольник Маклейна) (данная диаграмма коммутрует):

$$\begin{array}{ccccc}
 & & (A \otimes (B \otimes C)) \otimes D & & \\
 & \nearrow^{\alpha_{A,B,C} \otimes id_D} & & \searrow^{\alpha_{A,B \otimes C,D}} & \\
 ((A \otimes B) \otimes C) \otimes D & & & & A \otimes ((B \otimes C) \otimes D) \\
 \downarrow^{\alpha_{A \otimes B,C,D}} & & & & \downarrow^{id_A \otimes \alpha_{B,C,D}} \\
 (A \otimes B) \otimes (C \otimes D) & \xrightarrow{\alpha_{A,B,C \otimes D}} & & & A \otimes (B \otimes (C \otimes D))
 \end{array}$$

- Второе условие когерентности (тождество треугольника):

$$\begin{array}{ccc}
 (A \otimes \mathbb{1}) \otimes B & \xrightarrow{\alpha_{A,\mathbb{1},B}} & A \otimes (\mathbb{1} \otimes B) \\
 \searrow^{R_A \otimes id_B} & & \swarrow_{id_A \otimes L_B} \\
 & A \otimes B &
 \end{array}$$

Определение 7. Декартово замкнутая категория

Декартово замкнутая категория – это категория с терминальным объектом, произведениями и экспоненцированием:

1) Объект $\mathbb{1}$ в категории \mathcal{C} называется терминальным, если для любого объекта $A \in Ob_{\mathcal{C}}$ и для любых морфизмов $f, g \in Hom_{\mathcal{C}}(A, \mathbb{1})$, $f = g$.

2) Пусть $A, B \in Ob_{\mathcal{C}}$, тогда произведением объектов A и B называется объект $A \times B$, такой, для любого $C \in Ob_{\mathcal{C}}$ и для любых морфизмов $f \in Hom_{\mathcal{C}}(C, A)$ и $g \in Hom_{\mathcal{C}}(C, B)$, что диаграмма коммутрует:

$$\begin{array}{ccccc}
 & & C & & \\
 & \swarrow g & \downarrow \langle f, g \rangle & \searrow f & \\
 A & \xleftarrow{\pi_1} & A \times B & \xrightarrow{\pi_2} & B
 \end{array}$$

Морфизм $\langle f, g \rangle$ называется парой морфизмов f и g , а морфизмы вида π_1 и π_2 – каноническими проекциями.

3) Пусть $A, B \in Ob_{\mathcal{C}}$, тогда экспонентой объектов A и B называется объект B^A , такой, что диаграмма коммутует для любого объекта $C \in Ob_{\mathcal{C}}$ и для любого морфизма $f \in Hom_{\mathcal{C}}(C \times A, B)$:

$$\begin{array}{ccc}
 B^A \times A & \xrightarrow{\epsilon_{A,B}} & B \\
 \uparrow \Lambda(f) \times id_A & \nearrow f & \\
 C \times A & &
 \end{array}$$

где $\Lambda(f) \times id_A = \langle \Lambda(f) \circ \pi_1, id_A \circ \pi_2 \rangle$

Морфизмы вида $\epsilon_{A,B}$ называются вычисляющими стрелками, а морфизмы вида $\Lambda(f)$ – каррированием стрелки f .

Определение 8. Моноидальный функтор

Пусть $\langle \mathcal{C}, \otimes_1, \mathbb{1}_{\mathcal{C}} \rangle$ и $\langle \mathcal{D}, \otimes_2, \mathbb{1}_{\mathcal{D}} \rangle$ моноидальные категории.

Моноидальный функтор $\mathcal{F} : \langle \mathcal{C}, \otimes_1, \mathbb{1} \rangle \rightarrow \langle \mathcal{D}, \otimes_2, \mathbb{1}' \rangle$ это функтор $\mathcal{F} : \mathcal{C} \rightarrow \mathcal{D}$ с дополнительными естественными преобразованиями:

- $u : \mathbb{1}_{\mathcal{D}} \rightarrow \mathcal{F}\mathbb{1}_{\mathcal{C}}$;
- $*_{A,B} : \mathcal{F}A \otimes_{\mathcal{D}} \mathcal{F}B \rightarrow \mathcal{F}(A \otimes_{\mathcal{C}} B)$.

и условиями когерентности:

- Ассоциативность:

$$\begin{array}{ccc}
(\mathcal{F}A \otimes_{\mathcal{D}} \mathcal{F}B) \otimes_{\mathcal{D}} \mathcal{F}C & \xrightarrow{\alpha_{\mathcal{F}A, \mathcal{F}B, \mathcal{F}C}^{\mathcal{D}}} & \mathcal{F}A \otimes_{\mathcal{D}} (\mathcal{F}B \otimes_{\mathcal{D}} \mathcal{F}C) \\
\downarrow *_{A, B} \otimes_{\mathcal{D}} id_{\mathcal{F}B} & & \downarrow id_{\mathcal{F}A} \otimes_{\mathcal{D}} *_{B, C} \\
\mathcal{F}(A \otimes_{\mathcal{C}} B) \otimes_{\mathcal{D}} \mathcal{C} & & \mathcal{F}A \otimes_{\mathcal{D}} \mathcal{F}(B \otimes_{\mathcal{C}} C) \\
\downarrow *_{A \otimes_{\mathcal{C}} B, C} & & \downarrow *_{A, B \otimes_{\mathcal{C}} C} \\
\mathcal{F}((A \otimes_{\mathcal{C}} B) \otimes_{\mathcal{C}} C) & \xrightarrow{\mathcal{F}(\alpha_{A, B, C}^{\mathcal{C}})} & \mathcal{F}(A \otimes_{\mathcal{C}} (B \otimes_{\mathcal{C}} C))
\end{array}$$

- Свойство левой единицы:

$$\begin{array}{ccc}
\mathbb{1}_{\mathcal{D}} \otimes_{\mathcal{D}} \mathcal{F}A & \xrightarrow{u \otimes id_{\mathcal{F}A}} & \mathcal{F}\mathbb{1}_{\mathcal{C}} \otimes_{\mathcal{D}} \mathcal{F}A \\
\downarrow L_{\mathcal{F}A}^{\mathcal{D}} & & \downarrow *_{\mathbb{1}_{\mathcal{C}}, A} \\
\mathcal{F}A & \xleftarrow{\mathcal{F}(L_A^{\mathcal{C}})} & \mathcal{F}(\mathbb{1}_{\mathcal{C}} \otimes_{\mathcal{C}} A)
\end{array}$$

- Свойство правой единицы:

$$\begin{array}{ccc}
\mathcal{F}A \otimes_{\mathcal{D}} \mathbb{1}_{\mathcal{D}} & \xrightarrow{id_{\mathcal{F}A} \otimes u} & \mathcal{F}A \otimes_{\mathcal{D}} \mathcal{F}\mathbb{1}_{\mathcal{C}} \\
\downarrow R_{\mathcal{F}A}^{\mathcal{D}} & & \downarrow *_{A, \mathbb{1}_{\mathcal{C}}} \\
\mathcal{F}A & \xleftarrow{\mathcal{F}(R_A^{\mathcal{C}})} & \mathcal{F}(A \otimes_{\mathcal{C}} \mathbb{1}_{\mathcal{C}})
\end{array}$$

Определение 9. Аппликативный функтор

Аппликативный функтор – это тройка $\langle \mathcal{C}, \mathcal{K}, \eta \rangle$, где \mathcal{C} – это моноидальная категория, \mathcal{K} – это моноидальный эндифунктор и $\eta : Id_{\mathcal{C}} \Rightarrow \mathcal{K}$ – это естественное преобразование, такое, что:

- $u = \eta_{\mathbb{1}}$;
- $*_{A, B} \circ (\eta_A \otimes \eta_B) = \eta_{A \otimes B}$, то есть диаграмма коммутует:

$$\begin{array}{ccc}
A \otimes B & \xrightarrow{\eta_A \otimes \eta_B} & \mathcal{K}A \otimes \mathcal{K}B \\
& \searrow \eta_{A \otimes B} & \downarrow *_{A, B} \\
& & \mathcal{K}(A \otimes B)
\end{array}$$

По умолчанию мы будем рассматривать ниже аппликативный функтор над декартово замкнутой категорией.

2 Введение

2.1 Обзор имеющихся результатов

Модальная теория типов – относительно молодая область современной математической логики, изучающая конструктивные модальные логики с точки зрения программных вычислений, в котором каждому объекту, участвующему в вычислительной процедуре приписан свой тип данных, представляющий тот или иной объект.

Если в обычной теории типов, рассматриваются системы типов, соответствующие по Карри-Говарду, интуиционистским логикам (исчисление высказываний, исчисление высказываний второго порядка, исчисления предикатов первого или высших порядков, и т.д.), то в модальной теории типов изучаются системы типов, изоморфные по Карри-Говарду, интуиционистским модальным логикам.

2.2 Задача исследования

2.2.1 Логика IEL^-

Модальная интуиционистская логика IEL^- была предложена С. Артемовым и Т. Протопопеску [1]. IEL^- предлагает свою формальную теорию интуиционистских убеждений, согласанную с ВНК-семантикой интуиционистской логики.

Неформально \mathbf{KA} означает, что A верифицировано интуиционистски.

Логика IEL^- следующими схемами аксиом и правилами вывода:

Определение 10. *Интуиционистская модальная логика IEL^- :*

- 1) Аксиомы интуиционистского исчисления высказываний;
 - 2) $\mathbf{K}(A \rightarrow B) \rightarrow (\mathbf{KA} \rightarrow \mathbf{KB})$ (нормальность);
 - 3) $A \rightarrow \mathbf{KA}$ (ко-рефлексия);
- Правило вывода: MP .

Далее мы будем обозначать модальность как \square .

Легко видеть, что правило усиления в этой логике является производным.

В. Крупский и А. Ятманов построили секвенциальное исчисление для логики IEL ($IEL^- + \mathbf{KA} \rightarrow \neg\neg A$) и показали, что задача поиска вывода в данной логике PSPACE-полна [2].

2.2.2 Мотивация из функционального программирования

Функциональные языки программирования, такие, как Haskell [3], Idris [4], Purescript [5] или Elm [6] содержат специальные классы типов ¹ для вычислений с типами, вложенных в вычислительных контекст. Основные и наиболее интересные нам классы типов: `Functor` и `Applicative` ²:

```
class Functor f where
  fmap :: (a -> b) -> f a -> f b

class Functor f => Applicative f where
  pure :: a -> f a
```

¹Класс типов – это общий интерфейс (наличие одних и тех же методов) для специальной группы типов данных.

²Читатель может более подробно узнать о данной разновидности вычислений в стандартной библиотеке языка [7] или в следующей книге [8]

$$(<*>) :: f (a \rightarrow b) \rightarrow f a \rightarrow f b$$

Вычислительным контекстом (или *контейнером*) мы называем оператор над типами f , где f – это “функция” из $*$ в $*$: типовой оператор берет простой тип (имеющий сорт $*$) и возвращает другой простой тип сорта $*$. Более подробное описание системы типов с сортами, используемая для этих целей в Haskell, описана здесь [12].

Applicative functor allows to generalize the action of a functor for functions with arbitrary number of arguments, for instance:

```
liftA2 :: Applicative f => (a -> b -> c) -> f a -> f b -> f c
liftA2 f x y = ((pure f) <*> x) <*> y
```

It’s not difficult to see that modal axioms in IEL^- and types of the methods of Applicative class in Haskell-like languages (which is described below) are syntactically similar and we are going to show that this coincidence has a non-trivial computational meaning.

We investigate the relationship between intuitionistic epistemic logic IEL^- and applicative programming with side-effects by constructing the type system (which is called $\lambda_{\mathbf{K}}$) which is Curry-Howard isomorphic to IEL^- . So we will consider \mathbf{K} -modality as an arbitrary applicative functor and we prove that obtained type system is sound and complete for applicative functor on cartesian closed category (using the categorical definition proposed by Paterson [26]).

$\lambda_{\mathbf{K}}$ consists of the rules for simply typed lambda-calculus and special typing rules for lifting types into the applicative functor \mathbf{K} . We assume that our type system will axiomatize the simplest case of computation with effects with one container. We provide a proof-theoretical view at this kind of computations in functional programming and prove strong normalization and confluence.

3 Модальное λ -исчисление, основанное на исчислении IEL^-

Определим натуральное исчисление для IEL^- :

Определение 11. *Натуральное исчисление $NIEL^-$ для интуиционистской эпистемической логики IEL^- – это расширение натурального исчисления для интуиционистской логики высказываний с добавлением следующих правил вывода для модальности:*

$$\frac{\Gamma \vdash A}{\Gamma \vdash \Box A} \Box_I \qquad \frac{\Gamma \vdash \Box A_1, \dots, \Gamma \vdash \Box A_n \quad A_1, \dots, A_n \vdash B}{\Gamma \vdash \Box B}$$

Первое правило позволяет выводить ко-рефлексию. Второе модальное правило – это аналог для правила \Box_I в натуральном исчислении для конструктивной К (see [25]) без \Diamond .

Мы будем обозначать $\Gamma \vdash \Box A_1, \dots, \Gamma \vdash \Box A_n$ и $A_1, \dots, A_n \vdash B$ соответственно как $\Gamma \vdash \mathbf{K}\vec{A}$ и $\vec{A} \vdash B$ для краткости.

Лемма 1. $\Gamma \vdash_{NIEL^-} A \Rightarrow IEL^- \vdash \bigwedge \Gamma \rightarrow A$.

Доказательство. Индукция по построению вывода. Рассмотрим модальные случаи.

1) Если $\Gamma \vdash_{NIEL^-} A$, тогда $IEL^- \vdash \bigwedge \Gamma \rightarrow \Box A$.

- | | |
|---|------------------------|
| (1) $\bigwedge \Gamma \rightarrow A$ | предположение индукции |
| (2) $A \rightarrow \Box A$ | ко-рефлексия |
| (3) $(\bigwedge \Gamma \rightarrow A) \rightarrow ((A \rightarrow \Box A) \rightarrow (\bigwedge \Gamma \rightarrow \Box A))$ | теорема IEL^- |
| (4) $(A \rightarrow \Box A) \rightarrow (\bigwedge \Gamma \rightarrow \Box A)$ | из (1), (3) и МР |
| (5) $\bigwedge \Gamma \rightarrow \Box A$ | из (2), (4) и МР |

2) Если $\Gamma \vdash_{NIEL^-} \Box \vec{A}$ и $\vec{A} \vdash B$, то $IEL^- \vdash \bigwedge \Gamma \rightarrow \Box B$.

- | | |
|---|----------------------------------|
| (1) $\bigwedge \Gamma \rightarrow \bigwedge_{i=1}^n \Box A_i$ | предположение индукции |
| (2) $\bigwedge_{i=1}^n \Box A_i \rightarrow \Box \bigwedge_{i=1}^n A_i$ | теорема IEL^- |
| (3) $\bigwedge \Gamma \rightarrow \Box \bigwedge_{i=1}^n A_i$ | по (1), (2) и правилу силлогизма |
| (4) $\bigwedge_{i=1}^n A_i \rightarrow B$ | предположение индукции |
| (5) $(\bigwedge_{i=1}^n A_i \rightarrow B) \rightarrow \Box(\bigwedge_{i=1}^n A_i \rightarrow B)$ | ко-рефлексия |
| (6) $\Box(\bigwedge_{i=1}^n A_i \rightarrow B)$ | из (4), (5) и МР |
| (7) $\Box \bigwedge_{i=1}^n A_i \rightarrow \Box B$ | по (6) и по нормальности |
| (8) $\bigwedge \Gamma \rightarrow \Box B$ | по (3), (7) и правилу силлогизма |

□

Лемма 2. *Если $IEL^- \vdash A$, то $NIEL^- \vdash A$.*

Доказательство. Построение выводов для модальных аксиом в $NIEL^-$. Мы рассмотрим эти выводы ниже с использованием термов. □

Далее мы построим типизированное λ -исчисление по фрагменту $NIEL^-$ с правилами для импликации, конъюнкции и модальности. Данный фрагмент эквивалентен IEL^- без аксиом для отрицания и

дизъюнкции, что элементарно проверяется аналогично.

Определим термы и типы:

Определение 12. *Множество термов:*

Пусть \mathbb{V} счетное множество переменных. Термы $\Lambda_{\mathbf{K}}$ порождается следующей грамматикой:

$$\begin{aligned} \Lambda_{\mathbf{K}} ::= & \mathbb{V} \mid (\lambda \mathbb{V}. \Lambda_{\mathbf{K}}) \mid (\Lambda_{\mathbf{K}} \Lambda_{\mathbf{K}}) \mid (\Lambda_{\mathbf{K}}, \Lambda_{\mathbf{K}}) \mid (\pi_1 \Lambda_{\mathbf{K}}) \mid (\pi_2 \Lambda_{\mathbf{K}}) \mid \\ & (\text{pure } \Lambda_{\mathbf{K}}) \mid (\text{let pure } \mathbb{V}^* = \Lambda_{\mathbf{K}}^* \text{ in } \Lambda_{\mathbf{K}}) \end{aligned}$$

Где \mathbb{V}^* и $\Lambda_{\mathbf{K}}^*$ обозначают множество всех конечных последовательностей переменных $\bigcup_{i=0}^{\infty} \mathbb{V}^i$ и множество всех конечных последовательностей термов $\bigcup_{i=0}^{\infty} \Lambda_{\mathbf{K}}^i$. Последовательность переменных \vec{x} и последовательность термов \vec{M} в терме вида **let pure** должны иметь одинаковую длину. Иначе терм не будет правильно построенным.

Определение 13. *Множество типов:*

Пусть \mathbb{T} – это счетное множество атомарных типов. Типы $\mathbb{T}_{\mathbf{K}}$ с типовым оператором \square порождается следующей грамматикой:

$$\mathbb{T}_{\mathbf{K}} ::= \mathbb{T} \mid (\mathbb{T}_{\mathbf{K}} \rightarrow \mathbb{T}_{\mathbf{K}}) \mid (\mathbb{T}_{\mathbf{K}} \times \mathbb{T}_{\mathbf{K}}) \mid (\square \mathbb{T}_{\mathbf{K}}) \quad (1)$$

Контекст, его домен и кодомен определены стандартно [11][12].

Наша система состоит из следующих правил типизации в стиле Карри:

Определение 14. *Модальное λ -исчисление, основанное на исчислении IEL^- :*

$$\begin{array}{c} \frac{}{\Gamma, x : A \vdash x : A} \text{ax} \\[10pt] \frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x. M : A \rightarrow B} \rightarrow_i \qquad \frac{\Gamma \vdash M : A \rightarrow B \quad \Gamma \vdash N : A}{\Gamma \vdash MN : B} \rightarrow_e \\[10pt] \frac{\Gamma \vdash M : A \quad \Gamma \vdash N : B}{\Gamma \vdash \langle M, N \rangle : A \times B} \times_i \qquad \frac{\Gamma \vdash M : A_1 \times A_2}{\Gamma \vdash \pi_i M : A_i} \times_e, i \in \{1, 2\} \\[10pt] \frac{\Gamma \vdash M : A}{\Gamma \vdash \text{pure } M : \mathbf{K}A} \square_I \qquad \frac{\Gamma \vdash \vec{M} : \square \vec{A} \quad \vec{x} : \vec{A} \vdash N : B}{\Gamma \vdash \text{let pure } \vec{x} = \vec{M} \text{ in } N : \square B} \text{let}_{\square} \end{array}$$

Правило типизации \square аналогично правилу \bigcirc_I в монадическом метаязыке [17].

\square_I позволяет вкладывать объект типа A в текущий вычислительный контекст, изменяя его тип на $\square A$.

Правило типизации let_{\square} аналогично правилу \square -правилу в модальном λ -исчислении для интуиционистской минимальной нормальной модальной логики **IK** [19].

$\Gamma \vdash \vec{M} : \square \vec{A}$ – это синтаксический сахар для $\Gamma \vdash M_1 : \square A_1, \dots, \Gamma \vdash M_n : \square A_n$ и $\vec{x} : \vec{A} \vdash N : B$ – это краткая форма для $x_1 : A_1, \dots, x_n : A_n \vdash N : B$. **let pure** $\vec{x} = \vec{M}$ **in** N – это мгновенное локальное связывание в терме N . Мы будем использовать такую краткую форму вместо **let pure** $x_1, \dots, x_n = M_1, \dots, M_n$ **in** N .

Примеры выводов:

$$\begin{array}{c}
\frac{x : A \vdash x : A}{x : A \vdash \mathbf{pure} x : \Box A} \\
\hline
\vdash (\lambda x. \mathbf{pure} x) : A \rightarrow \Box A
\end{array}$$

$$\begin{array}{c}
\frac{f : \Box(A \rightarrow B) \vdash f : \Box(A \rightarrow B) \quad x : \Box A \vdash x : \Box A \quad \frac{g : A \rightarrow B \vdash g : A \rightarrow B \quad y : A \vdash y : A}{g : A \rightarrow B, y : A \vdash gy : B} \rightarrow_e}{\frac{f : \Box(A \rightarrow B), x : \Box A \vdash \mathbf{let pure} g, y = f, x \mathbf{ in } gy : \Box B}{f : \Box(A \rightarrow B) \vdash \lambda x. \mathbf{let pure} g, y = f, x \mathbf{ in } gy : \Box A \rightarrow \Box B} \mathbf{let_K}} \\
\hline
\vdash \lambda f. \lambda x. \mathbf{let pure} g, y = f, x \mathbf{ in } gy : \Box(A \rightarrow B) \rightarrow \Box A \rightarrow \Box B
\end{array}$$

Определим свободные переменные, подстановку, β -редукцию и η -редукцию. Многошаговая β -редукция и $\beta\eta$ -эквивалентность определены стандартно:

Определение 15. Множество свободных переменных $FV(M)$ для произвольного термина M :

- 1) $FV(x) = \{x\}$;
- 2) $FV(\lambda x. M) = FV(M) \setminus \{x\}$;
- 3) $FV(MN) = FV(M) \cup FV(N)$;
- 4) $FV(\langle M, N \rangle) = FV(M) \cup FV(N)$;
- 5) $FV(\pi_i M) \subseteq FV(M)$, $i \in \{1, 2\}$;
- 6) $FV(\mathbf{pure} M) = FV(M)$;
- 7) $FV(\mathbf{let pure} \vec{x} = \vec{M} \mathbf{ in } N) = \bigcup_{i=1}^n FV(M_i)$, где $n = |\vec{M}|$.

Во избежание лишних коллизий мы будем полагать, что если терм вида $\mathbf{let pure} x = (\mathbf{let pure} \vec{y} = \vec{N} \mathbf{ in } P) \mathbf{ in } M$ типизируется, то x не содержится в \vec{y} (свободные переменные термов M и \vec{N} не должны пересекаться), то есть последовательное применение локальных связываний требует на каждом шаге различных переменных.

Определение 16. Подстановка:

- 1) $x[x := N] = N$, $x[y := N] = x$;
- 2) $(MN)[x := N] = M[x := N]N[x := N]$;
- 3) $(\lambda x. M)[x := N] = \lambda x. M[y := N]$, $y \in FV(M)$;
- 4) $(M, N)[x := P] = (M[x := P], N[x := P])$;
- 5) $(\pi_i M)[x := P] = \pi_i(M[x := P])$, $i \in \{1, 2\}$;
- 6) $(\mathbf{pure} M)[x := P] = \mathbf{pure}(M[x := P])$;
- 7) $(\mathbf{let pure} \vec{x} = \vec{M} \mathbf{ in } N)[y := P] = \mathbf{let pure} \vec{x} = (\vec{M}[y := P]) \mathbf{ in } N$.

Определение 17. Правила β -редукции и η -редукции:

- 1) $(\lambda x. M)N \rightarrow_\beta M[x := N]$;
- 2) $\pi_1 \langle M, N \rangle \rightarrow_\beta M$;
- 3) $\pi_2 \langle M, N \rangle \rightarrow_\beta N$;
- 4) $\mathbf{let pure} \vec{x}, y, \vec{z} = \vec{M}, \mathbf{let pure} \vec{w} = \vec{N} \mathbf{ in } Q, \vec{P} \mathbf{ in } R \rightarrow_{\beta\Box} \mathbf{let pure} \vec{x}, \vec{w}, \vec{z} = \vec{M}, \vec{N}, \vec{P} \mathbf{ in } R[y := Q]$
- 5) $\mathbf{let pure} \vec{x} = \mathbf{pure} \vec{M} \mathbf{ in } N \rightarrow_{\beta\Box\mathbf{pure}} \mathbf{pure} N[\vec{x} := \vec{M}]$
- 6) $\mathbf{let pure} _ = _ \mathbf{ in } M \rightarrow_{\beta\mathbf{nec}} \mathbf{pure} M$, где $_$ – это пустая последовательность термов.
- 7) $\lambda x. fx \rightarrow_\eta f$;
- 8) $\langle \pi_1 P, \pi_2 P \rangle \rightarrow_\eta P$;
- 9) $\mathbf{let pure} x = M \mathbf{ in } x \rightarrow_{\Box id} M$;

Мы будем писать $M \rightarrow_r N$, если терм M редуцируется к терму N по одному из перечисленных выше правил.

По умолчанию мы используем стратегию вычисления с вызовом по имени.

Докажем стандартные леммы о контекстах ³:

Лемма 3. *Инверсия отношения типизации \Box_I .*

Пусть $\Gamma \vdash \mathbf{pure} M : \Box A$, тогда $\Gamma \vdash M : A$;

Доказательство. Очевидно. □

Лемма 4. *Базовые леммы.*

- Если $\Gamma \vdash M : A$ и $\Gamma \subseteq \Delta$, тогда $\Delta \vdash M : A$;
- Если $\Gamma \vdash M : A$, тогда $\Delta \vdash M : A$, где $\Delta = \{x_i : A_i \mid (x_i : A_i) \in \Gamma \ \& \ x_i \in FV(M)\}$
- Если $\Gamma, x : A \vdash M : B$ и $\Gamma \vdash N : A$, где $\Gamma \vdash M[x := N] : B$.

Рассмотрим случаи для правила \mathbf{let}_{\Box} .

Доказательство.

1) Пусть вывод заканчивается следующим правилом:

$$\frac{\Gamma \vdash \vec{M} : \Box \vec{A} \quad \vec{x} : \vec{A} \vdash N : B}{\Gamma \vdash \mathbf{let pure} \vec{x} = \vec{M} \mathbf{in} N : \Box B} \mathbf{let}_{\Box}$$

По предположению индукции $\Delta \vdash \vec{M} : \mathbf{K} \vec{A}$, тогда $\Delta \vdash \mathbf{let pure} \vec{x} = \vec{M} \mathbf{in} N : \Box B$.

Случаи 2)–3) рассматриваются аналогично. □

Теорема 1. *Редукция субъекта*

Если $\Gamma \vdash M : A$ и $M \rightarrow_{\beta\eta} N$, тогда $\Gamma \vdash N : A$

Доказательство. Индукция по выводу $\Gamma \vdash M : A$ и по порождению $\rightarrow_{\beta\eta}$.

Случаи с функцией и парами рассмотрены здесь [12] [13].

1) Если $\Gamma \vdash \mathbf{let pure} \vec{x}, y, \vec{z} = \vec{M}, \mathbf{let pure} \vec{w} = \vec{N} \mathbf{in} Q, \vec{P} \mathbf{in} R : \Box B$, тогда $\Gamma \vdash \mathbf{let pure} \vec{x}, \vec{w}, \vec{z} = \vec{M}, \vec{N}, \vec{P} \mathbf{in} R[y := Q] : \Box B$ по правилу 4).

2) Если $\Gamma \vdash \mathbf{let pure} x = M \mathbf{in} x : \Box A$, тогда $\Gamma \vdash M : \mathbf{K} A$ по правилу 9).

Рассмотрено здесь [19].

3) Пусть вывод заканчивается применением следующего правила

$$\frac{\Gamma \vdash \mathbf{pure} \vec{M} : \Box \vec{A} \quad \vec{x} : \vec{A} \vdash N : B}{\Gamma \vdash \mathbf{let pure} \vec{x} = \mathbf{pure} \vec{M} \mathbf{in} N : \Box B}$$

Тогда $\Gamma \vdash \vec{M} : \vec{A}$ по инверсии отношения типизации для \Box_I и $\Gamma \vdash N[\vec{x} := \vec{M}] : B$ по лемме 4, часть 3.

Тогда мы можем преобразовать данный вывод в следующий:

³Мы не будем рассматривать случаи для стандартных связок, так как они уже доказаны для просто типизированного λ -исчисления [11] [12]. Мы будем рассматривать только модальные случаи

$$\frac{\Gamma \vdash N[\vec{x} := \vec{M}] : B}{\Gamma \vdash \mathbf{pure} N[\vec{x} := \vec{M}] : \Box B} \Box_I$$

4) Пусть вывод заканчивается применением правила \mathbf{let}_{\Box} для типового объявления, выводимого из пустого контекста:

$$\frac{\vdash M : A}{\vdash \mathbf{let pure} _ = _ \mathbf{in} M : \Box A}$$

Тогда, если $\vdash M : A$, тогда $\vdash \mathbf{pure} M : \Box A$.

Данное рассуждение действует также и в обратную сторону. \square

Теорема 2.

\rightarrow_r сильно нормализуемо;

Доказательство.

Построим отображение из $\lambda_{\mathbf{K}}$ в просто типизированное λ -исчисление с типами \rightarrow , \times и выделенным типом натуральных чисел \mathbb{N} , для которого есть следующие правила типизации и редукции:

$$\begin{array}{c} \overline{\Gamma \vdash 0 : \mathbb{N}} \\ \frac{\Gamma \vdash n : \mathbb{N}}{\Gamma \vdash \mathbf{succ} n : \mathbb{N}} \\ \frac{\Gamma \vdash n : \mathbb{N} \quad \Gamma \vdash m : \mathbb{N}}{\Gamma \vdash n + m : \mathbb{N}} \end{array}$$

- $n + 0 \rightarrow_{\beta} n$;
- $(n + \mathbf{succ} m) \rightarrow_{\beta} \mathbf{succ} (n + m)$

Определим перевод $|\cdot|$ между данными исчислениями отдельно на типах, и на термах

Определение 18. *Интерпретация типов*

- $A \in \mathbb{T} \Rightarrow |A| = A$;
- $|A \rightarrow B| = |A| \rightarrow |B|$;
- $|A \times B| = |A| \times |B|$;
- $|\Box A| = \mathbb{N} \times |A|$.

Определение 19. *Интерпретация термов*

- $x \in \mathbb{V} \Rightarrow |x| = x$;
- $|\lambda x.M| = \lambda x.|M|$;
- $|(MN)| = |M||N|$;
- $|\langle M, N \rangle| = \langle |M|, |N| \rangle$;
- $|\pi_i M| = \pi_i |M|$, $i \in \{1, 2\}$;
- $|\mathbf{pure} M| = \langle 0, |M| \rangle$;

- $|\mathbf{let\ pure\ } \vec{x} = \vec{N} \mathbf{in\ } M| = \langle \sum_{i=1}^n \pi_1 |N|, |M|[\vec{x} := \pi_2 \vec{N}] \rangle$

Для удобства положим $|\mathbf{let\ pure\ } _ = _ \mathbf{in\ } M| = \langle 0, M \rangle$.

Рассмотрим интерпретацию последнего терма с помощью интерпретации правила типизации:

$$\frac{|\Gamma \vdash \vec{N} : \Box \vec{A}| = |\Gamma| \vdash |\vec{N}| : \mathbb{A} \times |\vec{A}| \quad |\vec{x} : \vec{A} \vdash M : B| = \vec{x} : |\vec{A}| \vdash |M| : |B|}{|\Gamma \vdash \mathbf{let\ pure\ } \vec{x} = \vec{N} \mathbf{in\ } M : \Box B| = |\Gamma| \vdash \langle \sum_{i=1}^n \pi_1 |N|, |M|[\vec{x} := \pi_2 \vec{N}] \rangle : \mathbb{N} \times |B|} \mathbf{let}_{\Box}$$

Лемма 5. Интерпретация сохраняет подстановку:

$$|M[x := N]| = |M|[x := |N|] \text{ для произвольного терма } M.$$

Доказательство. Несложная индукция по длине M . □

Лемма 6. $M \rightarrow_r N \Rightarrow |M| \rightarrow_{\beta\eta} |N|$

Доказательство. Рассмотрим случаи с β_{\Box} , $\beta_{\Box\mathbf{pure}}$ и $\Box id$.

1)

$$\begin{aligned} |\mathbf{let\ pure\ } x = (\mathbf{let\ pure\ } y = N \mathbf{in\ } P) \mathbf{in\ } M| &= \\ &\text{По интерпретации} \\ \langle \pi_1 |N|, |M|[x := |P|[y := \pi_2 |N|]] \rangle & \\ |\mathbf{let\ pure\ } y = N \mathbf{in\ } M[x := P]| &= \\ \langle \pi_1 |N|, |M|[x := |P|][y := \pi_2 |N|] \rangle &\equiv \\ &\text{По лемме Барендрегта по подстановке} \\ \langle \pi_1 |N|, |M|[y := \pi_2 |N|][x := |P|[y := \pi_2 |N|]] \rangle &\equiv \\ &\text{По нашему соглашению, } y \notin FV(M) \\ \langle \pi_1 |N|, |M|[x := |P|[y := \pi_2 |N|]] \rangle & \end{aligned}$$

2)

$$\begin{aligned} |\mathbf{let\ pure\ } \vec{x} = \mathbf{pure\ } \vec{N} \mathbf{in\ } M| &= \\ &\text{По интерпретации} \\ \langle 0 + \dots + 0, |M|[\vec{x} := |\vec{N}|] \rangle \rightarrow_{\beta} & \\ &\text{Многошаговая редукция для натуральных чисел} \\ \langle 0, |M|[\vec{x} := |\vec{N}|] \rangle &= \\ &\text{По интерпретации} \\ |\mathbf{pure\ } M[\vec{x} := \vec{N}]| & \end{aligned}$$

3)

$$\begin{aligned} |\mathbf{let\ pure\ } x = M \mathbf{in\ } x| &= \\ &\text{По интерпретации} \\ \langle \pi_1 |M|, x[x := \pi_2 |M|] \rangle &= \\ &\text{Подстановка} \\ \langle \pi_1 |M|, \pi_2 |M| \rangle \rightarrow_{\eta} & \\ &\text{Правило } \eta\text{-редукции для пары} \\ |M| & \end{aligned}$$

Таким образом, мы показали, что $\lambda_{\mathbf{K}}$ корректно относительно $\lambda_{\rightarrow, \times, \mathbb{N}}$, тогда $\lambda_{\mathbf{K}}$ сильно нормализуемо, поскольку $\lambda_{\rightarrow, \times, \mathbb{N}}$ сильно нормализуемо. □

Теорема 3. *Свойство Черча-Россера*

\rightarrow_r *конфлюентно.*

Доказательство. По лемме Ньюмана, если отношение сильно нормализуемо и локально конфлюентно, то отношение конфлюентно.

Достаточно показать локальную конфлюентность.

Лемма 7. *Локальная конфлюентность.*

Если $M \rightarrow_r N$ и $M \rightarrow_r Q$, тогда найдется такой терм P , что $N \rightarrow_r P$ и $Q \rightarrow_r P$.

Доказательство. Рассмотрим данную критическую пару и покажем, что оба терма из данной пары редуцируются к одному и тому же терму:

$$\begin{array}{ccc}
 \text{let pure } x = (\text{let pure } \vec{y} = \text{pure } \vec{N} \text{ in } P) \text{ in } M & & \\
 \beta \square \downarrow & \searrow \beta \square \text{pure} & \\
 \text{let pure } \vec{y} = \text{pure } \vec{N} \text{ in } M[x := P] & & \text{let pure } x = \text{pure } P[\vec{y} := \vec{N}] \text{ in } M \\
 \text{let pure } \vec{y} = \text{pure } \vec{N} \text{ in } M[x := P] \rightarrow \beta \square \text{pure} & & \\
 \text{pure } M[x := P][\vec{y} = \vec{N}] & & \\
 \text{let pure } x = \text{pure } P[\vec{y} := \vec{N}] \text{ in } M \rightarrow \beta \square \text{pure} & & \\
 \text{pure } M[x := P[\vec{y} := \vec{N}]] & &
 \end{array}$$

□

По лемме о подстановке

$$\text{pure } M[x := P][\vec{y} = \vec{N}] \equiv \text{pure } M[\vec{y} = \vec{N}][x := P[\vec{y} := \vec{N}]]$$

По нашему соглашению, $x \notin \vec{y}$, тогда

$$M[\vec{y} = \vec{N}][x := P[\vec{y} := \vec{N}]] \equiv M[x := P[\vec{y} := \vec{N}]]$$

□

Теорема 4.

Нормальная форма λ_K со стратегией вычисления с вызовом по имени обладает свойством подформульности: если M в нормальной форме, то все его подтермы также в нормальной форме.

Доказательство. Индукция по структуре M .

Случай **let pure** $\vec{x} = \vec{M}$ **in** N рассмотрен Какутани [19] [20].

Пусть **pure** M в нормальной форме, тогда M в нормальной форме и все его подтермы также в нормальной форме по предположению индукции.

Тогда, если **pure** M в нормальной форме, то и все его подтермы также в нормальной форме. □

4 Теоретико-категорная семантика

Теорема 5. *Корректность*

Пусть $\Gamma \vdash M : A$ и $M =_{\beta\eta} N$, тогда $\llbracket \Gamma \vdash M : A \rrbracket = \llbracket \Gamma \vdash N : A \rrbracket$

Доказательство.

Определение 20. Семантическая трансляция из $\lambda_{\mathbf{K}}$ в аппликативный функтор $\langle \mathcal{C}, \Box, \eta \rangle$ над декартово замкнутой категорией \mathcal{C} , где \Box – это моноидальный эндифунктор и η – это естественное преобразование $Id_{\mathcal{C}} \Rightarrow \Box$:

- Интерпретация типов:

- $\llbracket A \rrbracket := \hat{A}, A \in \mathbb{T}$, где \hat{A} – это объект категории \mathcal{C} , полученный в результате некоторого присваивания;
- $\llbracket A \rightarrow B \rrbracket := \llbracket B \rrbracket^{\llbracket A \rrbracket}$;
- $\llbracket A \times B \rrbracket := \llbracket A \rrbracket \times \llbracket B \rrbracket$.

- Интерпретация для модальных типов:

- $\llbracket \Box A \rrbracket = \Box \llbracket A \rrbracket$;

- Интерпретация для контекстов:

- $\llbracket \quad \rrbracket = \mathbb{1}$, где $\mathbb{1}$ – это терминальный объект в заданной декартово замкнутой категории;
- $\llbracket \Gamma, x : A \rrbracket = \llbracket \Gamma \rrbracket \times \llbracket A \rrbracket$

- Интерпретация для типовых объявлений:

- $\llbracket \Gamma \vdash M : A \rrbracket := \llbracket M \rrbracket : \llbracket \Gamma \rrbracket \rightarrow \llbracket A \rrbracket$.

- Интерпретация для правил типизации:

$$\begin{array}{c}
 \overline{\llbracket \Gamma, x : A \vdash x : A \rrbracket = \pi_2 : \llbracket \Gamma \rrbracket \times \llbracket A \rrbracket \rightarrow \llbracket A \rrbracket} \\
 \\
 \frac{\llbracket \Gamma, x : A \vdash M : B \rrbracket = \llbracket M \rrbracket : \llbracket \Gamma \rrbracket \times \llbracket A \rrbracket \rightarrow \llbracket B \rrbracket}{\llbracket \Gamma \vdash (\lambda x.M) : A \rightarrow B \rrbracket = \Lambda(\llbracket M \rrbracket) : \llbracket \Gamma \rrbracket \rightarrow \llbracket B \rrbracket^{\llbracket A \rrbracket}} \\
 \\
 \frac{\llbracket \Gamma \vdash M : A \rightarrow B \rrbracket = \llbracket M \rrbracket : \llbracket \Gamma \rrbracket \rightarrow \llbracket B \rrbracket^{\llbracket A \rrbracket} \quad \llbracket \Gamma \vdash N : A \rrbracket = \llbracket N \rrbracket : \llbracket \Gamma \rrbracket \rightarrow \llbracket A \rrbracket}{\llbracket \Gamma \vdash (MN) : B \rrbracket = \llbracket \Gamma \rrbracket \xrightarrow{\langle \llbracket M \rrbracket, \llbracket N \rrbracket \rangle} \llbracket B \rrbracket^{\llbracket A \rrbracket} \times \llbracket A \rrbracket \xrightarrow{\epsilon} \llbracket B \rrbracket} \\
 \\
 \frac{\llbracket \Gamma \vdash M : A \rrbracket = \llbracket M \rrbracket : \llbracket \Gamma \rrbracket \rightarrow \llbracket A \rrbracket \quad \llbracket \Gamma \vdash N : B \rrbracket = \llbracket N \rrbracket : \llbracket \Gamma \rrbracket \rightarrow \llbracket B \rrbracket}{\llbracket \Gamma \vdash \langle M, N \rangle : A \times B \rrbracket = \langle \llbracket M \rrbracket, \llbracket N \rrbracket \rangle : \llbracket \Gamma \rrbracket \rightarrow \llbracket A \rrbracket \times \llbracket B \rrbracket} \\
 \\
 \frac{\llbracket \Gamma \vdash M : A_1 \times A_2 \rrbracket = \llbracket M \rrbracket : \llbracket \Gamma \rrbracket \rightarrow \llbracket A_1 \rrbracket \times \llbracket A_2 \rrbracket}{\llbracket \Gamma \vdash \pi_i M : A_i \rrbracket = \llbracket \Gamma \rrbracket \xrightarrow{\llbracket M \rrbracket} \llbracket A_1 \rrbracket \times \llbracket A_2 \rrbracket \xrightarrow{\pi_i} \llbracket A_i \rrbracket} \quad i \in \{1, 2\} \\
 \\
 \frac{\llbracket \Gamma \vdash M : A \rrbracket = \llbracket M \rrbracket : \llbracket \Gamma \rrbracket \rightarrow \llbracket A \rrbracket}{\llbracket \Gamma \vdash \mathbf{pure} M : \Box A \rrbracket := \llbracket \Gamma \rrbracket \xrightarrow{\llbracket M \rrbracket} \llbracket A \rrbracket \xrightarrow{\eta_{\llbracket A \rrbracket}} \Box \llbracket A \rrbracket}
 \end{array}$$

$$\frac{\begin{array}{l} \llbracket \Gamma \vdash \vec{M} : \Box \vec{A} \rrbracket = \langle \llbracket M_1 \rrbracket, \dots, \llbracket M_n \rrbracket \rangle : \llbracket \Gamma \rrbracket \rightarrow \prod_{i=1}^n \Box \llbracket A_i \rrbracket \quad \llbracket \vec{x} : \vec{A} \vdash N : B \rrbracket = \llbracket N \rrbracket : \prod_{i=1}^n \llbracket A_i \rrbracket \rightarrow \llbracket B \rrbracket \\ \llbracket \Gamma \vdash \text{let pure } \vec{x} = \vec{M} \text{ in } M : \Box B \rrbracket = \Box (\llbracket N \rrbracket) \circ (\llbracket A_1 \rrbracket * \dots * \llbracket A_n \rrbracket) \circ \langle \llbracket M_1 \rrbracket, \dots, \llbracket M_n \rrbracket \rangle : \llbracket \Gamma \rrbracket \rightarrow \Box \llbracket B \rrbracket \end{array}}{\quad}$$

Определение 21. *Одновременная подстановка*

Пусть $\Gamma = \{x_1 : A_1, \dots, x_n : A_n\}$, $\Gamma \vdash M : A$ и для любых $i \in \{1, \dots, n\}$, $\Gamma \vdash M_i : A_i$.

Одновременная подстановка $M[\vec{x} := \vec{M}]$ определяется рекурсивно:

- $x_i[\vec{x} := \vec{M}] = M_i$;
- $(\lambda x. M)[\vec{x} := \vec{M}] = \lambda x. (M[\vec{x} := \vec{M}])$;
- $(MN)[\vec{x} := \vec{M}] = (M[\vec{x} := \vec{M}]) (N[\vec{x} := \vec{M}])$;
- $\langle M, N \rangle = \langle (M[\vec{x} := \vec{M}]), (N[\vec{x} := \vec{M}]) \rangle$;
- $(\pi_i P)[\vec{x} := \vec{M}] = \pi_i (P[\vec{x} := \vec{M}])$;
- $(\text{pure } M)[\vec{x} := \vec{M}] = \text{pure } (M[\vec{x} := \vec{M}])$;
- $(\text{let pure } \vec{x} = \vec{M} \text{ in } N)[\vec{y} := \vec{P}] = \text{let pure } \vec{x} = (\vec{M}[\vec{y} := \vec{P}]) \text{ in } N$

Лемма 8.

$$\llbracket M[x_1 := M_1, \dots, x_n := M_n] \rrbracket = \llbracket M \rrbracket \circ \langle \llbracket M_1 \rrbracket, \dots, \llbracket M_n \rrbracket \rangle.$$

Доказательство.

1)

$$\llbracket \Gamma \vdash (\text{pure } M)[\vec{x} := \vec{M}] : \Box A \rrbracket = \llbracket \Gamma \vdash \text{pure } (M[\vec{x} := \vec{M}]) : \Box A \rrbracket$$

Определение подстановки

$$\eta_{\llbracket A \rrbracket} \circ \llbracket (M[\vec{x} := \vec{M}]) \rrbracket$$

интерпретация для **pure**

$$\eta_{\llbracket A \rrbracket} \circ (\llbracket M \rrbracket \circ \langle \llbracket M_1 \rrbracket, \dots, \llbracket M_n \rrbracket \rangle) =$$

предположение индукции*

$$(\eta_{\llbracket A \rrbracket} \circ \llbracket M \rrbracket) \circ \langle \llbracket M_1 \rrbracket, \dots, \llbracket M_n \rrbracket \rangle =$$

ассоциативность композиции

$$\llbracket \Gamma \vdash \text{pure } M : \Box A \rrbracket \circ \langle \llbracket M_1 \rrbracket, \dots, \llbracket M_n \rrbracket \rangle =$$

интерпретация для **pure**

2)

$$\begin{aligned}
& \llbracket \Gamma \vdash (\text{let pure } \vec{x} = \vec{M} \text{ in } N)[\vec{y} := \vec{P}] : \Box B \rrbracket = \\
& \quad \text{определение одновременной подстановки} \\
& \llbracket \Gamma \vdash \text{let pure } \vec{x} = (\vec{M}[\vec{y} := \vec{P}]) \text{ in } N : \Box B \rrbracket = \\
& \quad \text{интерпретация } let_{\Box} \\
& \Box(\llbracket N \rrbracket) \circ (\llbracket A_1 \rrbracket * \dots * \llbracket A_n \rrbracket) \circ \llbracket \Gamma \vdash (\vec{M}[\vec{y} := \vec{P}]) : \Box \vec{A} \rrbracket = \\
& \quad \text{предположение индукции} \\
& \Box(\llbracket N \rrbracket) \circ (\llbracket A_1 \rrbracket * \dots * \llbracket A_n \rrbracket) \circ (\llbracket \vec{M} \rrbracket \circ \langle \llbracket P_1 \rrbracket, \dots, \llbracket P_n \rrbracket \rangle) = \\
& \quad \text{ассоциативность композиции} \\
& (\Box(\llbracket N \rrbracket)) \circ (\llbracket A_1 \rrbracket * \dots * \llbracket A_n \rrbracket) \circ \llbracket \vec{M} \rrbracket \circ \langle \llbracket P_1 \rrbracket, \dots, \llbracket P_n \rrbracket \rangle = \\
& \quad \text{по интерпретации} \\
& \llbracket \Gamma \vdash \text{let pure } \vec{x} = \vec{M} \text{ in } N : \Box B \rrbracket \circ \langle \llbracket P_1 \rrbracket, \dots, \llbracket P_n \rrbracket \rangle
\end{aligned}$$

□

Лемма 9.

Пусть $\Gamma \vdash M : A$ и $M \rightarrow_{\beta\eta} N$, тогда $\llbracket \Gamma \vdash M : A \rrbracket = \llbracket \Gamma \vdash N : A \rrbracket$;

Доказательство.

Случаи с правилом β -редукции для let_{\Box} рассмотрены здесь [20]. Рассмотрим случаи с **pure**.

$$1) \llbracket \Gamma \vdash \text{let pure } \vec{x} = \text{pure } \vec{M} \text{ in } N : \Box B \rrbracket = \llbracket \Gamma \vdash \text{pure } N[\vec{x} := \vec{M}] : \Box B \rrbracket$$

$$\begin{aligned}
& \llbracket \Gamma \vdash \text{let pure } \vec{x} = \text{pure } \vec{M} \text{ in } N : \Box B \rrbracket = \\
& \quad \text{интепретация} \\
& \Box(\llbracket N \rrbracket) \circ (\llbracket A_1 \rrbracket * \dots * \llbracket A_n \rrbracket) \circ \langle \eta_{\llbracket A_1 \rrbracket} \circ \llbracket M_1 \rrbracket, \dots, \eta_{\llbracket A_n \rrbracket} \circ \llbracket M_n \rrbracket \rangle = \\
& \quad \text{свойство пары морфизмов} \\
& \Box(\llbracket N \rrbracket) \circ (\llbracket A_1 \rrbracket * \dots * \llbracket A_n \rrbracket) \circ (\eta_{\llbracket A_1 \rrbracket} \times \dots \times \eta_{\llbracket A_n \rrbracket}) \circ \langle \llbracket M_1 \rrbracket, \dots, \llbracket M_n \rrbracket \rangle = \\
& \quad \text{ассоциативность композиции} \\
& \Box(\llbracket N \rrbracket) \circ ((\llbracket A_1 \rrbracket * \dots * \llbracket A_n \rrbracket) \circ (\eta_{\llbracket A_1 \rrbracket} \times \dots \times \eta_{\llbracket A_n \rrbracket})) \circ \langle \llbracket M_1 \rrbracket, \dots, \llbracket M_n \rrbracket \rangle = \\
& \quad \text{по определению аппликативного функтора} \\
& \Box(\llbracket N \rrbracket) \circ \eta_{\llbracket A_1 \rrbracket \times \dots \times \llbracket A_n \rrbracket} \circ \langle \llbracket M_1 \rrbracket, \dots, \llbracket M_n \rrbracket \rangle = \\
& \quad \text{естественность } \eta \\
& \eta_{\llbracket B \rrbracket} \circ \llbracket N \rrbracket \circ \langle \llbracket M_1 \rrbracket, \dots, \llbracket M_n \rrbracket \rangle = \\
& \quad \text{ассоциативность композиции} \\
& \eta_{\llbracket B \rrbracket} \circ (\llbracket N \rrbracket \circ \langle \llbracket M_1 \rrbracket, \dots, \llbracket M_n \rrbracket \rangle) = \\
& \quad \text{по лемме об одновременной подстановке} \\
& \eta_{\llbracket B \rrbracket} \circ \llbracket N[\vec{x} := \vec{M}] \rrbracket \\
& \quad \text{интерпретация} \\
& \llbracket \Gamma \vdash \text{pure } (N[\vec{x} := \vec{M}]) : \Box B \rrbracket
\end{aligned}$$

$$2) \llbracket \vdash \text{let pure } _ = _ \text{ in } M : \Box A \rrbracket = \llbracket \vdash \text{pure } M : \Box A \rrbracket$$

$$\llbracket \vdash \text{let pure } _ = _ \text{ in } M : \Box A \rrbracket =$$

интерпретация

$$\Box(\llbracket M \rrbracket) \circ u_{\mathbb{1}} =$$

определение аппликативного функтора

$$\Box(\llbracket M \rrbracket) \circ \eta_{\mathbb{1}} =$$

естественность η

$$\eta_{\llbracket A \rrbracket} \circ \llbracket M \rrbracket =$$

интерпретация

$$\llbracket \vdash \text{pure } M : \Box A \rrbracket$$

□

□

Теорема 6. Полнота

Пусть $\llbracket \Gamma \vdash M : A \rrbracket = \llbracket \Gamma \vdash N : A \rrbracket$, тогда $M =_{\beta\eta} N$.

Доказательство.

Мы будем работать с термовой моделью для простого типизированного λ -исчисления с \times и \rightarrow , стандартно описанной здесь [22]:

Определение 22. Эквивалентность на парах вида переменная-терм:

Определим такое бинарное отношение $\sim_{A,B} \subseteq \mathbb{V} \times \Lambda_{\mathbf{K}}$, что:

$$(x, M) \sim_{A,B} (y, N) \Leftrightarrow x : A \vdash M : B \ \& \ y : A \vdash N : A \ \& \ M =_{\beta\eta} N[y := x].$$

Нетрудно заметить, что данное отношение является отношением эквивалентности.

Обозначим класс эквивалентности как $[x, M]_{A,B} = \{(y, N) \mid (x, M) \sim_{A,B} (y, N)\}$ (ниже мы будем опускать индексы).

Определение 23. Категория $\mathcal{C}(\lambda)$:

- $Ob_{\mathcal{C}} = \{\hat{A} \mid A \in \mathbb{T}\} \cup \{\mathbb{1}\}$;
- $Hom_{\mathcal{C}(\lambda)}(\hat{A}, \hat{B}) = (\mathbb{V} \times \Lambda_{\mathbf{K}}) / \sim_{A,B}$;
- Пусть $[x, M] \in Hom_{\mathcal{C}(\lambda)}(\hat{A}, \hat{B})$ и $[y, N] \in Hom_{\mathcal{C}(\lambda)}(\hat{B}, \hat{C})$, тогда $[y, M] \circ [x, M] = [x, N[y := M]]$;
- Тожественный морфизм $id_{\hat{A}} = [x, x] \in Hom_{\mathcal{C}(\lambda)}(\hat{A}, \hat{A})$;
- Терминальный объект $\mathbb{1}$;
- $\widehat{A \times B} = \hat{A} \times \hat{B}$;
- Каноническая проекция: $[x, \pi_i x] \in Hom_{\mathcal{C}(\lambda)}(\hat{A}_1 \times \hat{A}_2, \hat{A}_i)$ for $i \in \{1, 2\}$;
- $\widehat{A \rightarrow B} = \hat{B}^{\hat{A}}$;
- Вычисляющая стрелка $\epsilon = [x, (\pi_2 x)(\pi_1 x)] \in Hom_{\mathcal{C}(\lambda)}(\hat{B}^{\hat{A} \times \hat{A}}, \hat{B})$.

Определение 24. Определим эндифунктор $\Box : \mathcal{C}(\lambda) \rightarrow \mathcal{C}(\lambda)$ таким образом, что для любых $[x, M] \in Hom_{\mathcal{C}(\lambda)}(\hat{A}, \hat{B})$, $\Box([x, M]) = [y, \text{let pure } x = y \text{ in } M] \in Hom_{\mathcal{C}(\lambda)}(\Box \hat{A}, \Box \hat{B})$ (обозначения: $f\text{tar } f$ для произвольной стрелки f).

Достаточно показать, что \Box – это аппликативный функтор над $\mathcal{C}(\lambda)$.

Лемма 10. *Функториальность*

- $fmap (g \circ f) = fmap (g) \circ fmap (f);$
- $fmap (id_{\hat{A}}) = id_{\Box \hat{A}}.$

Доказательство.

1)

$$\begin{aligned}
 fmap (g \circ f) &= fmap([y, N] \circ [x, M]) = \\
 &\quad \text{По определению композиции} \\
 fmap ([x, N[y := M]]) &= \\
 &\quad \text{По определению fmap} \\
 [z, \text{let pure } x = z \text{ in } N[y := M]] \\
 \\
 fmap (g) \circ fmap (f) &= fmap ([y, N]) \circ fmap ([x, M]) = \\
 &\quad \text{По определению fmap} \\
 [y_1, \text{let pure } y = y_1 \text{ in } N] \circ [z, \text{let pure } x = z \text{ in } M] &= \\
 &\quad \text{По определению композиции} \\
 [z, \text{let pure } y = y_1 \text{ in } N[y_1 := \text{let pure } x = z \text{ in } M]] &= \\
 &\quad \text{Подстановка} \\
 [z, \text{let pure } y = (\text{let pure } x = z \text{ in } M) \text{ in } N] &= \\
 &\quad \text{Правило } \beta\Box \\
 [z, \text{let pure } x = z \text{ in } N[y := M]]
 \end{aligned}$$

2)

$$\begin{aligned}
 fmap (id_{\hat{A}}) &= \\
 &\quad \text{Определение тождественного морфизма} \\
 fmap [x, x] &= \\
 &\quad \text{По определению fmap} \\
 [z, \text{let pure } x = z \text{ in } x] \\
 &\quad \text{Правило } \Box id \\
 [z, z] &= id_{\Box \hat{A}}
 \end{aligned}$$

□

Определение 25. *Определим естественные преобразования:*

- $\eta : Id \Rightarrow \Box, \text{ такое, что } \forall \hat{A} \in Ob_{\mathcal{C}(\lambda)}, \eta_{\hat{A}} = [x, \text{pure } x] \in Hom_{\mathcal{C}(\lambda)}(\hat{A}, \Box \hat{A});$
- $*_{A, B} : \Box \hat{A} \times \Box \hat{B} \rightarrow \Box(\hat{A} \times \hat{B}), \text{ такое, что } \forall \hat{A}, \hat{B} \in Ob_{\mathcal{C}(\lambda)}, *_{\hat{A}, \hat{B}} = [p, \text{let pure } x, y = \pi_1 p, \pi_2 p \text{ in } \langle x, y \rangle] \in Hom_{\mathcal{C}(\lambda)}(\Box \hat{A} \times \Box \hat{B}, \Box(\hat{A} \times \hat{B})).$

Реализация $*$ в нашей термовой модели – это частный случай правила let_{\Box} :

$$\frac{\frac{p : \Box A \times \Box B \vdash p : \Box A \times \Box B}{p : \Box A \times \Box B \vdash \pi_1 p : \Box A} \quad \frac{p : \Box A \times \Box B \vdash p : \Box A \times \Box B}{p : \Box A \times \Box B \vdash \pi_2 p : \Box B} \quad \frac{x : A \vdash x : A \quad y : B \vdash y : B}{x : A, y : B \vdash \langle x, y \rangle : A \times B}}{p : \Box A \times \Box B \vdash \text{let pure } x, y = \pi_1 p, \pi_2 p \text{ in } \langle x, y \rangle : \Box(A \times B)}$$

Лемма 11.

\Box – моноидальный эндифунктор.

Доказательство.

Показывается аналогично [19]. □

Лемма 12. *Естественность и когерентность η :*

- $\text{fmap } f \circ \eta_A = \eta_B \circ f$;
- $*_{\hat{A}, \hat{B}} \circ (\eta_A \times \eta_B) = \eta_{\hat{A} \times \hat{B}}$;

Доказательство.

$$\text{i) } \text{fmap } f \circ \eta_{\hat{A}} = \eta_{\hat{B}} \circ f$$

$$\begin{aligned} \eta_{\hat{B}} \circ f &= \\ &\quad \text{по определению} \\ [y, \mathbf{pure } y] \circ [x, M] &= \\ &\quad \text{композиция} \\ [x, \mathbf{pure } y[y := M]] &= \\ &\quad \text{подстановка} \\ [x, \mathbf{pure } M] \end{aligned}$$

С другой стороны:

$$\begin{aligned} \text{fmap } f \circ \eta_{\hat{A}} &= \\ &\quad \text{по определению} \\ [z, \mathbf{let pure } x = z \mathbf{ in } M] \circ [x, \mathbf{pure } x] &= \\ &\quad \text{композиция} \\ [x, \mathbf{let pure } x = z \mathbf{ in } M[z := \mathbf{pure } x]] &= \\ &\quad \text{подстановка} \\ [x, \mathbf{let pure } x = \mathbf{pure } x \mathbf{ in } M] &= \\ &\quad \text{правило } \beta\text{-редукции} \\ [x, \mathbf{pure } M[x := x]] &= \\ &\quad \text{постановка} \\ [x, \mathbf{pure } M] \end{aligned}$$

$$\text{ii) } *_{\hat{A}, \hat{B}} \circ (\eta_{\hat{A}} \times \eta_{\hat{B}}) = \eta_{\hat{A} \times \hat{B}}$$

$$\begin{aligned}
& *_{\hat{A}, \hat{B}} \circ (\eta_{\hat{A}} \times \eta_{\hat{B}}) = \\
& \quad \text{раскрытие} \\
& [q, \text{let pure } x, y = \pi_1 q, \pi_2 q \text{ in } \langle x, y \rangle] \circ [p, \langle \text{pure } (\pi_1 p), \text{pure } (\pi_2 p) \rangle] = \\
& \quad \text{композиция} \\
& [p, \text{let pure } x, y = \pi_1 q, \pi_2 q \text{ in } \langle x, y \rangle [q := \langle \text{pure } (\pi_1 p), \text{pure } (\pi_2 p) \rangle]] = \\
& \quad \text{подстановка} \\
& [p, \text{let pure } x, y = \pi_1 (\langle \text{pure } (\pi_1 p), \text{pure } (\pi_2 p) \rangle), \pi_2 (\langle \text{pure } (\pi_1 p), \text{pure } (\pi_2 p) \rangle) \text{ in } \langle x, y \rangle] = \\
& \quad \text{правило } \beta\text{-редукции} \\
& [p, \text{let pure } x, y = \text{pure } (\pi_1 p), \text{pure } (\pi_2 p) \text{ in } \langle x, y \rangle] = \\
& \quad \text{правило } \beta\text{-редукции} \\
& [p, \text{pure } (\langle x, y \rangle [x := \pi_1 p, y := \pi_2 p])] = \\
& \quad \text{подстановка} \\
& [p, \text{pure } \langle \pi_1 p, \pi_2 p \rangle] = \\
& \quad \text{правило } \eta\text{-редукции} \\
& [p, \text{pure } p] = \\
& \quad \text{определение} \\
& \eta_{\hat{A} \times \hat{B}}
\end{aligned}$$

□

Определение 26.

$$u_{\mathbb{1}} = [\blacksquare, \text{let pure } _ = _ \text{ in } \blacksquare] \in \text{Hom}_{\mathcal{C}(\lambda)}(\mathbb{1}, \square \mathbb{1}).$$

Лемма 13.

$$u_{\mathbb{1}} = \eta_{\mathbb{1}}$$

Доказательство. Следует напрямую из правила $\beta\text{нес}$.

□

Лемма 14. \square – это аппликативный функтор.

Доказательство. Непосредственно следует из предыдущих лемм.

□

Аналогично [24], мы применяем трансляцию из $\lambda_{\mathbf{K}}$ к произвольной декартово замкнутой категории с аппликативным функтором \square , тогда мы имеем $\llbracket \Gamma \vdash M : A \rrbracket = [x, M[x_i := \pi_i x]]$, so $M =_{\beta\eta} N \Leftrightarrow \llbracket \Gamma \vdash M : A \rrbracket = \llbracket \Gamma \vdash N : A \rrbracket$.

□

Список литературы

- [1] Artemov S. and Protopopescu T., “Intuitionistic Epistemic Logic”, *The Review of Symbolic Logic*, 2016, vol. 9, no 2. pp. 266-298.
- [2] Krupski V. N. and Yatmanov A., “Sequent Calculus for Intuitionistic Epistemic Logic IEL”, *Logical Foundations of Computer Science: International Symposium, LFCS 2016, Deerfield Beach, FL, USA, January 4-7, 2016. Proceedings*, 2016, pp. 187-201.
- [3] Haskell Language. // URL: <https://www.haskell.org>. (Date: 1.08.2017)
- [4] Idris. A Language with Dependent Types.// URL:<https://www.idris-lang.org>. (Date: 1.08.2017)
- [5] Purescript. A strongly-typed functional programming language that compiles to JavaScript. URL: <http://www.purescript.org>. (Date: 1.08.2017)
- [6] Elm. A delightful language for reliable webapps. // URL: <http://elm-lang.org>. (Date: 1.08.2017)
- [7] Hackage, “The base package” // URL: <https://hackage.haskell.org/package/base-4.10.0.0> (Date: 1.08.2017)
- [8] Lipovaca M, “Learn you a Haskell for Great Good!”. //URL: <http://learnyouahaskell.com/chapters> (Date: 1.08.2017)
- [9] McBride C. and Paterson R., “Applicative programming with effects *Journal of Functional Programming*, 2008, vol. 18, no 01. pp 1-13.
- [10] McBride C. and Paterson R, “Functional Pearl. Idioms: applicative programming with effects”, *Journal of Functional Programming*, 2005. vol. 18, no 01. pp 1-20.
- [11] R. Nederpelt and H. Geuvers, “Type Theory and Formal Proof: An Introduction”. *Cambridge University Press*, New York, NY, USA, 2014. pp. 436.
- [12] Sorensen M. H. and Urzyczyn P, “Lectures on the Curry-Howard isomorphism”, *Studies in Logic and the Foundations of Mathematics*, vol. 149, *Elsevier Science*, 1998. pp 261.
- [13] Pierce B. C., “Types and Programming Languages”. *Cambridge, Mass: The MIT Press*, 2002. pp. 605.
- [14] Girard J.-Y., Taylor P. and Lafont Y, “Proofs and Types”, *Cambridge University Press*, New York, NY, USA, 1989. pp. 175.
- [15] Barendregt. H. P., “Lambda calculi with types”// Abramsky S., Gabbay Dov M., and S. E. Maibaum, “Handbook of logic in computer science (vol. 2), Osborne Handbooks Of Logic In Computer Science”, Vol. 2. *Oxford University Press, Inc.*, New York, NY, USA, 1993. pp 117-309.
- [16] Hindley J. Roger, “Basic Simple Type Theory”. *Cambridge University Press*, New York, NY, USA, 1997. pp. 185.
- [17] Pfenning F. and Davies R., “A judgmental reconstruction of modal logic”, *Mathematical Structures in Computer Science*, vol. 11, no 4, 2001, pp. 511-540.
- [18] H.P. Barendregt. The Lambda Calculus — Its Syntax and Semantics. *Studies in Logic and the Foundations of Mathematics*, vol. 103. Amsterdam: North-Holland, 1985.

- [19] Yoshihiko KAKUTANI, A Curry-Howard Correspondence for Intuitionistic Normal Modal Logic, Computer Software, Released February 29, 2008, Online ISSN , Print ISSN 0289-6540.
- [20] Kakutani Y. (2007) Call-by-Name and Call-by-Value in Normal Modal Logic. In: Shao Z. (eds) Programming Languages and Systems. APLAS 2007. Lecture Notes in Computer Science, vol 4807. Springer, Berlin, Heidelberg
- [21] T. Abe. Completeness of modal proofs in first-order predicate logic. Computer Software, JSSST Journal, 24:165 – 177, 2007.
- [22] Lambek, J. and Scott P.J. (1986) Introduction to Higher Order Categorical Logic, Cambridge Studies in Advanced Mathematics 7, Cambridge: Cambridge University Press.
- [23] Samuel Eilenberg and Max Kelly, Closed categories. Proc. Conf. Categorical Algebra (La Jolla, Calif., 1965).
- [24] Samson Abramsky and Nikos Tzevelekos, Introduction to Categories and Categorical Logic
- [25] G. A. Kavvos. The Many Worlds of Modal Λ -calculi: I. Curry-Howard for Necessity, Possibility and Time
- [26] Ross Paterson. in Mathematics of Program Construction, Madrid, 2012, Lecture Notes in Computer Science, vol. 7342, pp. 300–323, Springer, 2012.