Категорная модель модального лямбда-исчисления, основанного на интуиционистской логике.

Даня Рогозин

МГУ

Март, 2018

Мотивация. Функциональное программирование на языке Haskell.

- Обратимся в рамках мотивации к функциональному программированию на таких языках, как Haskell, Purescript, Elm или Idris;
- Без ограничения общности разделим типы в языке Haskell (или в любом другом из языков выше) на две части: простые типы и параметризованные;
- Простые типы (`Int`, `String`, `Char`, etc) – это привычные типы данных;
- Параметризованные типы (`List Int`, `Maybe Char`, `IO String`) используются для вычислений в рамках оговоренного вычислительного контекста;
- Аналогично можно и разделить функции.

Мотивация. Функтор.

Класс типов `Functor` – это общий интерфейс для “выполнения действия над параметризованным типом, обобщение функции `map` на списках”:

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

Motivation. Monad.

Согласно Hackage: “С точки зрения хаскеллиста лучше всего определять монаду как тип данных для произвольных действий”. В частности, вычисления в мире ввода-вывода – частный случай монадических вычислений.

(Старое) определение монады:

```
class Functor m => Monad m where
    return : a -> m a
    (»=) :: m a -> (a -> m b) -> m b.
```

Монадическая композиция (композиция действий):

```
(>=>) :: Monad m => (a -> m b) -> (b -> m c) -> a -> m c
```

Конечная цель: аппликативные функторы.

Аппликативные функторы сильнее функторов и слабее монад:

```
class Functor f => Applicative f where
    pure :: a -> f a
    (<*>) :: f (a -> b) -> f a -> f b
```

Используя аппликативный функтор, мы можем вложить значение в вычислительный контекст `f` с помощью `pure` и выполнить аппликацию внутри `f` применением `<*>`.

Использование:

- Обобщение `fmap` для функции произвольной арности:
  `pure f <*> a1 <*> ... <*> an`
- Парсинг;
- Монада в современном Haskell является наследником аппликатива.

Монадические вычисления в теории.

1) *Eugenio Moggi.* “Notions of computation and monads.” Inf. Comput., 93(1): 55–92, 1991.

2) *Frank Pfenning and Rowan Davies.* “A judgmental reconstruction of modal logic.” Mathematical. Structures in Comp. Sci. 11, 4 (August 2001), 511—540.

3) *Bierman, G., and De Paiva, V.*. On an Intuitionistic Modal Logic. Studia Logica: An International Journal for Symbolic Logic, 65(3), 2000. 383–416.
etc...

Аппликативные функторы.

К сожалению, аппликативный функтор является далеко не самой известной концепцией за вне сообщества хаскеллистов. Возможная причина: аппликативные функторы рассмотрены с программистской точки зрения, без теоретического рассмотрения, то есть теоретико-доказательного построения синтаксиса и алгебраической (категорной) модели.

Пример нескольких работ:

1) *Conor McBride and Ross Paterson.* "Applicative Programming with Effects." Journal of Functional Programming 18:1 (2008), pages 1–13.
2) *Ross Paterson.* "Constructing Applicative Functors." Mathematics of Program Construction, Madrid, 2012, Lecture Notes in Computer Science vol. 7342, pp. 300–323, Springer, 2012.

Белое пятно: стоит рассмотреть модальное лямбда-исчисление, которые могло бы аксиоматизировать вычисления с аппликативным функтором и имело хорошую алгебраическую модель.

Интуиционистская эпистемическая логика IEL^-.

Данную проблему удобно решать, если мы располагает некоторой конструктивной модальной логикой с хорошими аксиомами, по которой мы можем построить интересное нам модальное лямбда-исчисление:

Определение

Интуиционистская эпистемическая логика IEL^-:
1) Аксиомы IPC;
2) $\mathbf{K}(A \to B) \to (\mathbf{K}A \to \mathbf{K}B)$ (нормальность);
3) $A \to \mathbf{K}A$ (ко-рефлексия);
Правило: MP.

1) *Artemov S., Protopopescu T.* (2014, June). Intuitionistic epistemic logic. ArXiv, math.LO 1406.1582v1.
2) *Krupski V. N., Alexey Y.* "Sequent calculus for intuitionistic epistemic logic IEL" // Logical Foundations of Computer Science – Vol. 9537 of Lecture Notes in Computer Science. – Springer, 2016. – P. 187–201.

Натуральный вывод для IEL^-.

Определение

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

$$\frac{\Gamma \vdash A}{\Gamma \vdash \mathbf{K}A}\ \mathbf{K}_I$$

$$\frac{\Gamma \vdash \mathbf{K}A_1, \ldots, \Gamma \vdash \mathbf{K}A_n \qquad A_1, \ldots, A_n \vdash B}{\Gamma \vdash \mathbf{K}B}$$

Натуральный вывод для IEL$^-$.

Лемма

$\Gamma \vdash_{NIEL^-} A \Rightarrow IEL^- \vdash \bigwedge \Gamma \to A.$

Proof.

Индукция по построению вывода. Рассмотрим модальные случаи.

1) Если $\Gamma \vdash_{\mathrm{NIEL}^-} A$, тогда $\mathrm{IEL}^- \vdash \bigwedge \Gamma \to \mathbf{K}A$.

(1)	$\bigwedge \Gamma \to A$	предположение индукции
(2)	$A \to \mathbf{K}A$	ко-рефлексия
(3)	$(\bigwedge \Gamma \to A) \to ((A \to \mathbf{K}A) \to (\bigwedge \Gamma \to \mathbf{K}A))$	теорема IPC
(4)	$(A \to \mathbf{K}A) \to (\bigwedge \Gamma \to \mathbf{K}A)$	из (1), (3) и MP
(5)	$\bigwedge \Gamma \to \mathbf{K}A$	из (2), (4) и MP

$\square$

Натуральный вывод для IEL^-.

Proof.

2) Если $\Gamma \vdash_{\mathsf{NIEL}^-} \mathbf{K}\vec{A}$ и $\vec{A} \vdash B$, то $\mathsf{IEL}^- \vdash \bigwedge \Gamma \to \mathbf{K}B$.

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

$\square$

Натуральный вывод для IEL$^-$.

Лемма

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

Proof.

Построение выводов для модальных аксиом в NIEL$^-$. $\square$

Модальное лямбда-исчисление по IEL^-

Определение

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

$$\frac{\Gamma \vdash M : A}{\Gamma \vdash \textbf{pure}\ \ M : \mathbf{K}A}\ \mathbf{K}_I$$

$$\frac{\Gamma \vdash \vec{M} : \mathbf{K}\vec{A} \qquad \vec{x} : \vec{A} \vdash N : B}{\Gamma \vdash \textbf{let pure}\ \vec{x} = \vec{M}\ \textbf{in}\ N : \mathbf{K}B}\ let_{\mathbf{K}}$$

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

Примеры деревьев вывода

$$\dfrac{\dfrac{x : A \vdash x : A}{x : A \vdash \mathbf{pure}\, x : \mathbf{K}A}}{\vdash (\lambda x.\mathbf{pure}\, x) : A \to \mathbf{K}A}$$

$$\dfrac{\dfrac{f : \mathbf{K}(A \to B) \vdash f : \mathbf{K}(A \to B) \qquad x : \mathbf{K}A \vdash x : \mathbf{K}A \qquad \dfrac{g : A \to B \vdash g : A \to B \qquad y : A \vdash y : A}{g : A \to B, y : A \vdash gy : B}}{\dfrac{f : \mathbf{K}(A \to B), x : \mathbf{K}A \vdash \mathbf{let\ pure}\, g, y = f, x\, \mathbf{in}\, gy : \mathbf{K}B}{f : \mathbf{K}(A \to B) \vdash \lambda x.\mathbf{let\ pure}\, g, y = f, x\, \mathbf{in}\, gy : \mathbf{K}A \to \mathbf{K}B}}\,\mathrm{let}_{\mathbf{K}}}{\vdash \lambda f.\lambda x.\mathbf{let\ pure}\, g, y = f, x\, \mathbf{in}\, gy : \mathbf{K}(A \to B) \to \mathbf{K}A \to \mathbf{K}B}$$

Подстановка

Определение

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) $(\textbf{pure}\ M)[x := P] = \textbf{pure}\ (M[x := P])$;
7) $(\textbf{let pure}\ \vec{x} = \vec{M}\ \textbf{in}\ N)[y := P] = \textbf{let pure}\ \vec{x} = (\vec{M}[y := P])\ \textbf{in}\ N$.

Редукция

Определение

Правила β-редукции и η-редукции:

1) $(\lambda x.M)N \to_\beta M[x := N]$;

2) $\pi_1\langle M, N\rangle \to_\beta M$;

3) $\pi_2\langle M, N\rangle \to_\beta N$;

4) $$\begin{array}{l}\textbf{let pure } \vec{x}, y, \vec{z} = \vec{M}, \textbf{let pure } \vec{w} = \vec{N} \textbf{ in } Q, \vec{P} \textit{ in } R \to_\beta \\ \textbf{let pure } \vec{x}, \vec{w}, \vec{z} = \vec{M}, \vec{N}, \vec{P} \textbf{ in } R[y := Q]\end{array}$$

5) $\textbf{let pure } \vec{x} = \textbf{pure } \vec{M} \textbf{ in } N \to_\beta \textbf{pure } N[\vec{x} := \vec{M}]$

6) $\textbf{let pure } \_\_ = \_\_ \textbf{ in } M \to_\beta \textbf{pure } M$, где $\_\_$ – это пустая последовательность термов.

7) $\lambda x.fx \to_\eta f$;

8) $\langle \pi_1 P, \pi_2 P\rangle \to_\eta P$;

9) $\textbf{let pure } x = M \textbf{ in } x \to_\eta M$;

Метатеоретические свойства системы

Теорема

Редукция субъекта
Если $\Gamma \vdash M : A$ *и* $M \twoheadrightarrow_{\beta\eta} N$, *тогда* $\Gamma \vdash N : A$

Теорема

Отношение $\twoheadrightarrow_{\beta}$ *сильно нормализуемо;*

Теорема

Отношение $\twoheadrightarrow_{\beta}$ *конфлюентно.*

Теорема

Нормальная форма $\lambda_{\mathbf{K}}$ *со стратегией вычисления с вызовом по имени обладает свойством подформульности: если* M *в нормальной форме, то всего его подтермы также в нормальной форме.*