Соответствие Карри–Ховарда. Лекция вторая

А. Н. Непейвода
ИПС им. А.К. Айламазяна РАН,
МГТУ им. Н.Э. Баумана

MathClub,
25 февраля 2022, Иннополис

Аксиомы комбинаторной логики

$$\mathbf{S}(\mathbf{K}\mathbf{K}) = \mathbf{S}(\mathbf{S}(\mathbf{K}\mathbf{S})(\mathbf{S}(\mathbf{K}\mathbf{K})(\mathbf{S}(\mathbf{K}\mathbf{S})\mathbf{K})))(\mathbf{K}\mathbf{K})$$

$$\mathbf{S}(\mathbf{K}\mathbf{S})(\mathbf{S}(\mathbf{K}\mathbf{K})) = \mathbf{S}(\mathbf{K}\mathbf{K})(\mathbf{S}(\mathbf{S}(\mathbf{K}\mathbf{S})(\mathbf{S}(\mathbf{K}\mathbf{K})\mathbf{I}))(\mathbf{K}\mathbf{I}))$$

$$\begin{aligned}&\mathbf{S}(\mathbf{K}(\mathbf{S}(\mathbf{K}\mathbf{S})))(\mathbf{S}(\mathbf{K}\mathbf{S})(\mathbf{S}(\mathbf{K}\mathbf{S})))\\ &\quad = \mathbf{S}(\mathbf{S}(\mathbf{K}\mathbf{S})(\mathbf{S}(\mathbf{K}\mathbf{K})(\mathbf{S}(\mathbf{K}\mathbf{S})(\mathbf{S}(\mathbf{K}(\mathbf{S}(\mathbf{K}\mathbf{S})))\mathbf{S}))))(\mathbf{K}\mathbf{S})\end{aligned}$$

$$\mathbf{I} = \mathbf{S}(\mathbf{S}(\mathbf{K}\mathbf{S})\mathbf{K})(\mathbf{K}\mathbf{I})$$

Упражнение

Известно, что композиция $\lambda x\,y\,z.x\;(y\;z)$ — это `fmap` для функтора `Reader`. Выразить её в бесточечном виде по алгоритму и по логике.

Упражнение

Известно, что композиция $\lambda x\, y\, z.x\ (y\ z)$ — это `fmap` для функтора `Reader`. Выразить её в бесточечном виде по алгоритму и по логике.

```
newtype MyRead r a = Rd run :: r -> a
instance Applicative (MyRead r) where
  pure v = Rd (\_ -> v)
  Rd f <*> Rd g = Rd (\x -> f x (g x))
```

Упражнение

Известно, что композиция $\lambda x\,y\,z.x\;(y\;z)$ — это `fmap` для функтора `Reader`. Выразить её в бесточечном виде по алгоритму и по логике.

```
newtype MyRead r a = Rd run :: r -> a
instance Applicative (MyRead r) where
  pure v = Rd (\_ -> v)
  Rd f <*> Rd g = Rd (\x -> f x (g x))
```

```
fmap f (Rd x)
    = (Rd ((pure (<*>)) <*> pure) f x)
```

Вычисления «от противного»

Проблема термов в типизированном λ-исчислении (и чистого функционального стиля): «что упало, то пропало». Если вычислено $(M\ N)$, то M и N по отдельности уже потеряны навсегда.

Вычисления «от противного»

Проблема термов в типизированном λ-исчислении (и чистого функционального стиля): «что упало, то пропало». Если вычислено $(M\ N)$, то M и N по отдельности уже потеряны навсегда.

Логичный выход — возвраты, если стало ясно, что вычисления зашли в тупик. Аналог в доказательствах — работа с отрицанием.

$$\frac{\Gamma \vdash \tau,\ \Gamma \vdash \neg\,\tau}{\Gamma \vdash \sigma}$$ (из лжи следует всё что угодно)

$$\frac{\Gamma \vdash \neg\,\neg\,\tau}{\Gamma \vdash \tau}$$ (снятие двойного отрицания)

Теорема Гливенко

Пусть Φ — пропозициональная формула. Тогда $\vdash \Phi$ в классической логике $\Leftrightarrow \vdash \neg\neg\Phi$ в конструктивной интуиционистской логике.

Здесь под конструктивной интуиционистской логикой понимаем минимальную логику + «из лжи следует всё что угодно».

Теорема Гливенко

Пусть Φ — пропозициональная формула. Тогда $\vdash \Phi$ в классической логике $\Leftrightarrow \vdash \neg\neg\Phi$ в конструктивной интуиционистской логике.

$$
\begin{array}{l}
* \neg(\neg\neg A \Rightarrow A) \\
\quad \vert \; *A \\
\quad \vert \quad \vert \; *\neg\neg A \\
\quad \vert \quad \vert \quad \vert \; A \\
\quad \vert \quad \vert \; \neg\neg A \Rightarrow A \\
\quad \vert \quad \vert \; \bot \\
\quad \vert \; \neg A \\
\quad \vert \; *\neg\neg A \\
\quad \vert \quad \vert \; \bot \\
\quad \vert \quad \vert \; A \\
\quad \vert \; \neg\neg A \Rightarrow A \\
\quad \vert \; \bot \\
\neg\neg(\neg\neg A \Rightarrow A)
\end{array}
$$

Теорема Гливенко

Пусть Φ — пропозициональная формула. Тогда $\vdash \Phi$ в классической логике $\Leftrightarrow \vdash \neg\neg\Phi$ в конструктивной интуиционистской логике.

$$
\begin{array}{l}
*\neg(\neg\neg A \Rightarrow A) \\
\mid\ *A \\
\mid\ \mid\ *\neg\neg A \\
\mid\ \mid\ \mid\ A \\
\mid\ \mid\ \neg\neg A \Rightarrow A \\
\mid\ \mid\ \bot \\
\mid\ \neg A \\
\mid\ *\neg\neg A \\
\mid\ \mid\ \bot \\
\mid\ \mid\ A \\
\mid\ \neg\neg A \Rightarrow A \\
\mid\ \bot \\
\neg\neg(\neg\neg A \Rightarrow A)
\end{array}
$$

$$
\begin{array}{l}
*\neg(((A \Rightarrow B) \Rightarrow A) \Rightarrow A) \\
\mid\ *A \\
\mid\ \mid\ *(A \Rightarrow B) \Rightarrow A \\
\mid\ \mid\ \mid\ A \\
\mid\ \mid\ ((A \Rightarrow B) \Rightarrow A) \Rightarrow A \\
\mid\ \mid\ \bot \\
\mid\ \neg A \\
\mid\ *(A \Rightarrow B) \Rightarrow A \\
\mid\ \mid\ *A \\
\mid\ \mid\ \mid\ B \\
\mid\ \mid\ A \Rightarrow B \\
\mid\ \mid\ A \\
\mid\ ((A \Rightarrow B) \Rightarrow A) \Rightarrow A \\
\mid\ \bot \\
\neg\neg(((A \Rightarrow B) \Rightarrow A) \Rightarrow A)
\end{array}
$$

Переход к относительному отрицанию

«Вычисления зашли в тупик»$\Rightarrow$ приводят к нежелательному результату. Интерпретация $\neg_R \Phi = (\Phi \Rightarrow R)$.

Рассмотрим предыдущие два вывода с точки зрения извлечения термов из их конструкции.

Переход к относительному отрицанию

«Вычисления зашли в тупик»$\Rightarrow$ приводят к нежелательному результату. Интерпретация $\neg_R\Phi = (\Phi \Rightarrow R)$.

$*(\neg_R \neg_R A \Rightarrow A) \Rightarrow R$ (тип x)

$*A$ (тип y)

$*\neg_R \neg_R A$ (тип z)

A (терм y)

$\neg_R \neg_R A \Rightarrow A$ (терм $\lambda z.y$)

R (терм $x\ (\lambda z.y)$)

$A \Rightarrow R$ (терм $\lambda y.x\ (\lambda z.y)$)

$*(A \Rightarrow R) \Rightarrow R$ (тип w)

R (терм $w\ (\lambda y.x\ (\lambda z.y))$)

A (??!!!)

$\neg_R \neg_R A \Rightarrow A$

R

$((\neg_R \neg_R A \Rightarrow A) \Rightarrow R) \Rightarrow R$

Переход к относительному отрицанию

«Вычисления зашли в тупик»$\Rightarrow$ приводят к нежелательному результату. Интерпретация $\neg_R \Phi = (\Phi \Rightarrow R)$.

$$
\begin{array}{ll}
*(((A \Rightarrow B) \Rightarrow A) \Rightarrow A) \Rightarrow R & \text{(тип } x\text{)} \\
\quad *A & \text{(тип } y\text{)} \\
\quad\quad *(A \Rightarrow B) \Rightarrow A & \text{(тип } z\text{)} \\
\quad\quad\quad A & \text{(терм } y\text{)} \\
\quad\quad ((A \Rightarrow B) \Rightarrow A) \Rightarrow A & \text{(терм } \lambda z.y\text{)} \\
\quad\quad R & \text{(терм } x\ (\lambda z.y)\text{)} \\
\quad A \Rightarrow R & \text{(терм } \lambda y.x\ (\lambda z.y)\text{)} \\
\quad *(A \Rightarrow B) \Rightarrow A & \text{(тип } u\text{)} \\
\quad\quad *A & \text{(тип } w\text{)} \\
\quad\quad\quad B & \text{??!!!} \\
\quad\quad A \Rightarrow B & \\
\quad\quad A & \\
\quad ((A \Rightarrow B) \Rightarrow A) \Rightarrow A & \\
\quad R & \\
((((A \Rightarrow B) \Rightarrow A) \Rightarrow A) \Rightarrow R) \Rightarrow R &
\end{array}
$$

Переход к относительному отрицанию

«Вычисления зашли в тупик»$\Rightarrow$ приводят к нежелательному результату. Интерпретация $\neg_R \Phi = (\Phi \Rightarrow R)$.

$*(((A \Rightarrow B) \Rightarrow A) \Rightarrow A) \Rightarrow B$ (тип x)

$*A$ (тип y)

$*(A \Rightarrow B) \Rightarrow A$ (тип z)

A (терм y)

$((A \Rightarrow B) \Rightarrow A) \Rightarrow A$ (терм $\lambda z.y$)

B (терм $x\ (\lambda z.y)$)

$A \Rightarrow B$ (терм $\lambda y.x\ (\lambda z.y)$)

$*(A \Rightarrow B) \Rightarrow A$ (тип u)

$*A$ (тип w)

B (терм $(\lambda y.x\ (\lambda z.y))\ w$)

$A \Rightarrow B$ (терм $\lambda y.x\ (\lambda z.y)$)

A (терм $u\ (\lambda y.x\ (\lambda z.y))$)

$((A \Rightarrow B) \Rightarrow A) \Rightarrow A$ (терм $\lambda u.u\ (\lambda y.x\ (\lambda z.y))$)

B (терм $x\ (\lambda u.u\ (\lambda y.x\ (\lambda z.y)))$)

$((((A \Rightarrow B) \Rightarrow A) \Rightarrow A) \Rightarrow B) \Rightarrow B$

(терм $(\lambda x.x\ (\lambda u.u\ (\lambda y.x\ (\lambda z.y))))$)

Навешивание двойного отрицания

Видно, что использование «возвратных» термов (переход от типа Φ к типу $\neg_R \neg_R \Phi$) расширяет возможности языка. Возникает вопрос, в каких подформулах лучше это делать?

- Преобразование Колмогорова:
  $\tau(A) = \neg_R \neg_R A \quad \tau(\Phi \Rightarrow \Psi) = \neg_R \neg_R(\tau(\Phi) \Rightarrow \tau(\Psi))$
- Вариант более слабого преобразования (в стиле Куроды) — это $\neg_R \neg_R \tau'(\Phi)$, где:
  $\tau'(A) = A \quad \tau'(\Phi \Rightarrow \Psi) = \tau'(\Phi) \Rightarrow \neg_R \neg_R \tau'(\Psi))$

В примерах для краткости $(\Phi \Rightarrow R) \Rightarrow R$ переобозначим как Φ'.

Переход по Колмогорову

$*(A' \Rightarrow ((A' \Rightarrow B')' \Rightarrow B')') \Rightarrow R$

> $*(A \Rightarrow R) \Rightarrow R$
>
> > $*((A' \Rightarrow B')' \Rightarrow B') \Rightarrow R$
> >
> > > $*((A' \Rightarrow B') \Rightarrow R) \Rightarrow R$
> > >
> > > > $*B \Rightarrow R$
> > > >
> > > > > $*A' \Rightarrow B'$
> > > > >
> > > > > > $*A \Rightarrow R$
> > > > > >
> > > > > > > R
> > > > > >
> > > > > > $(A \Rightarrow R) \Rightarrow R$
> > > > > >
> > > > > > $(B \Rightarrow R) \Rightarrow R$
> > > > > >
> > > > > > R
> > > > >
> > > > > $(A' \Rightarrow B') \Rightarrow R$
> > > > >
> > > > > R
> > > >
> > > > $(B \Rightarrow R) \Rightarrow R$
> > >
> > > $(A' \Rightarrow B')' \Rightarrow B'$
> > >
> > > R
> >
> > $(((A' \Rightarrow B')' \Rightarrow B') \Rightarrow R) \Rightarrow R$
>
> $A' \Rightarrow ((A' \Rightarrow B')' \Rightarrow B')'$
>
> R

$((A' \Rightarrow ((A' \Rightarrow B')' \Rightarrow B')') \Rightarrow R) \Rightarrow R$

Извлечённый терм: $\lambda k_0.k_0\ (\lambda x k_1.k_1\ (\lambda y k_2.y\ (\lambda k_3.k_3\ (\lambda k_4.x\ k_4)\ k_2)))$

Слабый переход по Куроде

$$
\begin{array}{l}
*(A \Rightarrow ((A \Rightarrow B') \Rightarrow B')') \Rightarrow R \\
\mid\; *A \\
\mid\;\mid\; *((A \Rightarrow B') \Rightarrow B') \Rightarrow R \\
\mid\;\mid\;\mid\; *A \Rightarrow B' \\
\mid\;\mid\;\mid\;\mid\; (B \Rightarrow R) \Rightarrow R \\
\mid\;\mid\;\mid\;\mid\; *B \Rightarrow R \\
\mid\;\mid\;\mid\;\mid\;\mid\; R \\
\mid\;\mid\;\mid\;\mid\; (B \Rightarrow R) \Rightarrow R \\
\mid\;\mid\;\mid\; ((A \Rightarrow B') \Rightarrow B' \\
\mid\;\mid\;\mid\; R \\
\mid\;\mid\; (((A \Rightarrow B') \Rightarrow B') \Rightarrow R) \Rightarrow R \\
\mid\; A \Rightarrow ((A \Rightarrow B') \Rightarrow B')' \\
\mid\; R \\
((A \Rightarrow ((A \Rightarrow B') \Rightarrow B')') \Rightarrow R) \Rightarrow R
\end{array}
$$

Извлечённый терм: $\lambda k_0.k_0\ (\lambda x k_1.k_1\ (\lambda y k_2.y\ x\ k_2))$.

Система передачи продолжений

Система передачи продолжений в CBN-стиле:

- $\tau_N(\mathrm{const}) = \lambda k.k\ \mathrm{const}$
- $\tau_N(x) = \lambda k.x\ k$
- $\tau_N(\lambda x.M) = \lambda k.k\ (\lambda x.\tau_N(M))$
- $\tau_N(M\ N) = \lambda k.\tau_N(M)\ (\lambda f.f\ \tau_N(N)\ k)$

Система передачи продолжений

Система передачи продолжений в CBN-стиле:

- $\tau_N(\mathrm{const}) = \lambda k.k\ \mathrm{const}$
- $\tau_N(x) = \lambda k.x\ k$
- $\tau_N(\lambda x.M) = \lambda k.k\ (\lambda x.\tau_N(M))$
- $\tau_N(M\ N) = \lambda k.\tau_N(M)\ (\lambda f.f\ \tau_N(N)\ k)$

Система передачи продолжений в CBV-стиле:

- $\tau_V(\mathrm{const}) = \lambda k.k\ \mathrm{const}$
- $\tau_V(x) = \lambda k.k\ x$
- $\tau_V(\lambda x.M) = \lambda k.k\ (\lambda x.\tau_V(M))$
- $\tau_V(M\ N) = \lambda k.\tau_V(M)\ (\lambda f.\tau_V(N)\ (\lambda a.f\ a\ k))$

Применение CPS-преобразования

- $M \to^* \mathit{const}$ при CBV-стратегии $\Leftrightarrow$ $\tau_V(M)\ \mathit{id} \to^* \mathit{const}$ при какой угодно стратегии.
- $M \to^* \mathit{const}$ при CBN-стратегии $\Leftrightarrow$ $\tau_N(M)\ \mathit{id} \to^* \mathit{const}$ при какой угодно стратегии.

Монада Cont

```
newtype MyCont r a = Mc  runCont :: (a -> r) -> r
instance Functor (MyCont r) where
  fmap f (Mc cps_a) = ...
instance Applicative (MyCont r) where
  pure v = ...
  Mc cps_f <*> Mc cps_a = ...
```

Монада Cont

```
newtype MyCont r a = Mc  runCont :: (a -> r) -> r
instance Functor (MyCont r) where
  fmap f (Mc cps_a) =  Mc (\cps -> cps_a (cps . f))
instance Applicative (MyCont r) where
  pure v = ...
  Mc cps_f <*> Mc cps_a = ...
```

Монада Cont

```
newtype MyCont r a = Mc  runCont :: (a -> r) -> r
instance Functor (MyCont r) where
  fmap f (Mc cps_a) =  Mc (\cps -> cps_a (cps . f))
instance Applicative (MyCont r) where
  pure v = Mc (\r -> r v)
  Mc cps_f <*> Mc cps_a = ...
```

Монада Cont

```
newtype MyCont r a = Mc  runCont :: (a -> r) -> r
instance Functor (MyCont r) where
  fmap f (Mc cps_a) =  Mc (\cps -> cps_a (cps . f))
instance Applicative (MyCont r) where
  pure v = Mc (\r -> r v)
  Mc cps_f <*> Mc cps_a =
      Mc (\cps_b -> cps_f (\f -> cps_a (cps_b . f)))
```

Упражнение

Пользуясь соответствием Карри–Ховарда, построить явным образом терм, реализующий bind-оператор >>= для монады Cont.
Тип >>= :: MyCont a -> (a -> MyCont b) -> MyCont b, и это всё, что нужно знать для решения задачи.

Прерывания вычислений

- Жёсткое прерывание — полный выход из контекста ($\neg_{\mathrm{R}} \neg_{\mathrm{R}} A \Rightarrow A$);
- Мягкое прерывание — возвращает текущий контекст вычислений ($(\neg_{\mathrm{R}} A \Rightarrow A) \Rightarrow A$) — закон Пирса, а также тип оператора call-with-current-continuation.

Вывод = конструкция

$*((((A \Rightarrow B) \Rightarrow B) \Rightarrow A) \Rightarrow B)$

- $*B \Rightarrow A$
  - $*(A \Rightarrow B) \Rightarrow B$
    - $*A \Rightarrow B$
      - B
      - A
    - $(A \Rightarrow B) \Rightarrow A$
    - A Закон Пирса!
  - $((A \Rightarrow B) \Rightarrow B) \Rightarrow A$
  - B
- $(B \Rightarrow A) \Rightarrow B$
- B Закон Пирса!

$((((A \Rightarrow B) \Rightarrow B) \Rightarrow A) \Rightarrow B) \Rightarrow B$

Вывод = конструкция

$*((((A \Rightarrow B) \Rightarrow B) \Rightarrow A) \Rightarrow B)$ (тип терма x)

$*B \Rightarrow A$ (тип терма y)

$*(A \Rightarrow B) \Rightarrow B$ (тип терма z)

$*A \Rightarrow B$ (тип терма w)

B (терм $z\,w$)

A (терм $y\,(z\,w)$)

$(A \Rightarrow B) \Rightarrow A$ (терм $\lambda w.y\,(z\,w)$)

A ($T_1 = \mathrm{callcc}\ \lambda w.y\,(z\,w)$)

$((A \Rightarrow B) \Rightarrow B) \Rightarrow A$ (терм $\lambda z.T_1$)

B (терм $x\,(\lambda z.T_1)$)

$(B \Rightarrow A) \Rightarrow B$ (терм $\lambda y.x\,(\lambda z.T_1)$)

B ($T_2 = \mathrm{callcc}\ \lambda y.x\,(\lambda z.T_1)$)

$((((A \Rightarrow B) \Rightarrow B) \Rightarrow A) \Rightarrow B) \Rightarrow B$ терм $\lambda x.T_2$

Определение монады

```
class Applicative M => Monad M where
  return :: a -> M a
  (>>=) :: M a -> (a -> M b) -> M b
```

Дополнительные монадические операторы:

```
join :: M M a -> M a
fmap :: (a -> b) -> M a -> M b
```

Соответствие Карри–Ховарда

Propositional Lax Logic — монадическая логика с единственным модальным оператором $\bigcirc$.

Правила вывода (+ стандартные правила естественного вывода в интуиционистской логике):

- $\dfrac{A}{\bigcirc A}$ (Return)
- $\dfrac{A \Rightarrow \bigcirc B}{\bigcirc A \Rightarrow \bigcirc B}$ (Bind)

Сохраняет возможность извлечь монадический оператор из вывода типов! Обратим внимание на инвертированный порядок аргументов.

Пример разбора задачи на монадическое соответствие

Построим оператор `join` из операторов >>= и `return`.

$* \bigcirc \bigcirc A$ (тип терма $\mathtt{x}$)

(Тут нужно придумать, как объединить контейнеры, оборачивающие $\mathtt{x}$)

$\bigcirc A$

$\bigcirc \bigcirc A \Rightarrow \bigcirc A$ (тип терма $\lambda \mathtt{x}. \ldots$)

Единственный оператор, умеющий заглядывать внутрь контейнера — это `bind`. Он требует два аргумента: контейнерный типа `M a` и монадическую функцию типа `a -> M b`. Поскольку нужно заглянуть внутрь только одной из двух контейнерных оболочек, в роли типа `a` выступает также монада `M a′`. Значит, нам нужна стрелка типа `M a′ -> M a′`. Это функция `id`, вывод которой мы уже умеем строить.

Пример разбора задачи на монадическое соответствие

$* \bigcirc \bigcirc A$ (тип терма x)

$* \bigcirc A$ (тип терма y)

$\bigcirc A$ (просто возвращаем сам y)

$\bigcirc A \Rightarrow \bigcirc A$ (тип терма $\lambda y.y$)

(Аргументы `bind`*-оператора построены, осталось применить их в правильном порядке)*

$\bigcirc A$

$\bigcirc \bigcirc A \Rightarrow \bigcirc A$ (тип терма $\lambda x.\ldots$)

Пример разбора задачи на монадическое соответствие

$* \bigcirc \bigcirc A$ (тип терма x)

$* \bigcirc A$ (тип терма y)

$\bigcirc A$ (просто возвращаем сам y)

$\bigcirc A \Rightarrow \bigcirc A$ (тип терма $\lambda y.y$)

(Аргументы `bind`*-оператора построены, осталось применить их в правильном порядке)*

$\bigcirc A$ (тип терма $x >>= (\lambda y.y)$)

$\bigcirc \bigcirc A \Rightarrow \bigcirc A$ (тип терма $\lambda x.(x >>= (\lambda y.y))$)

Пример разбора задачи на монадическое соответствие

$$
\dfrac{\dfrac{* \bigcirc \bigcirc A \ \text{(тип терма } x\text{)} \quad \dfrac{* \bigcirc A \ \text{(тип терма } y\text{)} \quad \bigcirc A \ \text{(просто возвращаем сам } y\text{)}}{\bigcirc A \Rightarrow \bigcirc A \ \text{(тип терма } \lambda y.y\text{)}}}{\bigcirc A \ \text{(тип терма } x{>>=}(\lambda y.y)\text{)}}}{\bigcirc \bigcirc A \Rightarrow \bigcirc A \ \text{(тип терма } \lambda x.(x{>>=}(\lambda y.y))\text{)}}
$$

(Аргументы `bind`-оператора построены, осталось применить их в правильном порядке)

С помощью построения вывода в модальной логике мы просто решили уравнение в ФВП, получив ответ:
`join x = x >>= id`.

Задачи

- Построить вывод $(A \Rightarrow B) \Rightarrow \bigcirc A \Rightarrow \bigcirc B$.
- Построить вывод $\bigcirc(A \Rightarrow B) \Rightarrow \bigcirc A \Rightarrow \bigcirc B$.

Задачи

- Построить вывод $(A \Rightarrow B) \Rightarrow \bigcirc A \Rightarrow \bigcirc B$.
- Построить вывод $\bigcirc(A \Rightarrow B) \Rightarrow \bigcirc A \Rightarrow \bigcirc B$.

liftM — аналог fmap для монад, стандартно определяется через bind и return. Как?
А как определить через них <*>?

Задачи

- Построить вывод $(A \Rightarrow B) \Rightarrow \bigcirc A \Rightarrow \bigcirc B$.
- Построить вывод $\bigcirc(A \Rightarrow B) \Rightarrow \bigcirc A \Rightarrow \bigcirc B$.

liftM — аналог fmap для монад, стандартно определяется через bind и return. Как?

```
liftM f m = m >>= \i -> return (f i)
```

А как определить через них <*>?

```
f <*> a
   = f >>= (\xf -> a >>= (\xa -> return (xf xa)))
```

Всё!

Спасибо за внимание!