Тьюринг-полнота нетипизированного лямбда-исчисления

Машина Поста — машина Тьюринга на бинарном алфавите. Закодируем описание и состояние машины лямбда-термами:

- Будем кодировать алфавит с помощью `True` и `False`.
- Состояния $q \in \{\bot\} \cup Q$ — нумералами $0, \dots, |Q|$ (терминальное состояние $\bot$ кодируется нулём).
- Таблицу состояний $\delta : Q \times \{0, 1\} \to (\{\bot\} \cup Q) \times \{0, 1\} \times \{\mathtt{L}, \mathtt{S}, \mathtt{R}\}$ — списком пар длины $|Q|$, где каждая компонента каждой пары — `Pair` q (`Pair` b n), где q — номер нового состояния, b — новый символ, n — $\lambda xyz.x$, $\lambda xyz.y$ или $\lambda xyz.z$).
- Бесконечную ленту и головку — парой списков символов, где в первом списке перечислены символы слева от головки (читая от головки), а во втором — справа (списки конечные, поскольку в любой момент на ленте записано только конечное количество символов).

Постройте замкнутый терм `Interpret` такой, что для любого описания таблицы δ и начального состояния

$$s = \mathtt{Pair}\ q_0\ (\mathtt{Pair}\ [b_1, \dots, b_n]\ [b'_1, \dots, b'_m])$$

про $R = \mathtt{Interpret}\ \delta\ s$ верно следующее:

- Если у R нет нормальной формы, соответствующая машина Поста не завершает свою работу на данном входе;
- Если R нормализуется, то $\mathrm{nf}(R)$ — это состояние ленты в результате работы соответствующей машины.

P.S. При желании можете вводить более удобную Вам нотацию для лямбда-термов (инфиксные операторы, кортежи), главное — объясните, как она устроена.

P.P.S. Также, при желании, можете взять любую другую более удобную Вам кодировку; опять же, главное — объясните, как она устроена.

решение

Базовые комбинаторы:

- $\mathtt{True} = \mathtt{T} = \lambda xy.\ x$
- $\mathtt{False} = \mathtt{F} = \lambda xy.\ y$
- $\langle\ \rangle = \lambda xyP.\ P\ x\ y$
  - $[\,x_1,\ \dots,\ x_n\,] = \langle x_1,\ \langle \dots,\ \langle x_n,\ \mathtt{F}\rangle\rangle\rangle$
- $\mathtt{get} = \lambda il.\ (i\ (\lambda l.\ l\ \mathtt{F})\ l)\ \mathtt{T}$ — n-ный элемент списка l
- $\mathtt{isnil} = \lambda l.\ l\ (\lambda xyz.\ \mathtt{F})\ \mathtt{T}$

Будем кодировать

- таблицу состояний машины δ как $[\,\langle\langle q,\ \langle b,\ n\rangle\rangle\rangle,\ \langle\langle q,\ \langle b,\ n\rangle\rangle\rangle\,]$
- состояние ленты t как $\langle [\,b_0,\ \dots,\ b_n\,],\ [\,b'_1,\ \dots,\ b'_m\,]\rangle$, где b_0 — ячейка под головкой, b_1 — ячейка слева от головки, b'_1 — ячейка справа от головки, и т. д. Будем считать, что за пределами списков все ячейки `False`.
- индексы состояний как натуральные числа: $q_0 = \lambda Sx.\ x$ и $q_n = \lambda Sx.\ S\ q_{n-1}$
- положение дел s как $\langle q_i,\ t\rangle$

Итак.

- $\mathtt{rewr} = \lambda tb.\ \langle\langle b,\ t\ \mathtt{T}\ \mathtt{F}\rangle,\ t\ \mathtt{F}\rangle$
  - получив состояние ленты t и значение b, возвращает t, где вместо первого элемента первого списка (ячейки под головкой) записано b.
- $\mathtt{shift} = \lambda tn.\ \langle n\ (t\ \mathtt{T}\ \mathtt{F})\ (t\ \mathtt{T})\ (\langle t\ \mathtt{F}\ \mathtt{T},\ t\ \mathtt{T}\rangle),\ n\ (\langle t\ \mathtt{T}\ \mathtt{T},\ t\ \mathtt{F}\rangle)\ (t\ \mathtt{F})\ (t\ \mathtt{F}\ \mathtt{F})\rangle$
  - получив состояние ленты t и сдвиг $n = \lambda xyz.\ x/y/z$, возвращает сдвинутую ленту
    - $\langle [\,b_1,\ \dots,\ b_n\,],\ [\,b_0,\ b'_1,\ \dots,\ b'_m\,]\rangle$, если $n = \lambda xyz.\ x$

    * $\langle [b_0, \dots, b_n], [b'_1, \dots, b'_m] \rangle$, если $n = \lambda xyz.\ y$
    * $\langle [b'_1, b_0 \dots, b_n], [b'_2, \dots, b'_m] \rangle$, если $n = \lambda xyz.\ z$
  - в результате сдвига один из списков может оказаться пустым, так что введем еще и `xtnd`, добавляющий в список еще одну (пустую) ячейку

- $\texttt{xtnd} = \lambda t.\ \langle \texttt{isnil}\ (t\ \texttt{T})\ \langle \texttt{F}, \texttt{F} \rangle\ (t\ \texttt{T}), \texttt{isnil}\ (t\ \texttt{F})\ \langle \texttt{F}, \texttt{F} \rangle\ (t\ \texttt{F}) \rangle$
  - принимает состояние ленты — пару списков и, если один из списков пуст (F), заменяет его на список из одного `F` ($\langle \texttt{F}, \texttt{F} \rangle$)

- $\texttt{b} = \lambda qt.\ q\ (t\ \texttt{T}\ \texttt{T})\ \texttt{F}\ \texttt{T}$
  $\texttt{n} = \lambda qt.\ q\ (t\ \texttt{T}\ \texttt{T})\ \texttt{F}\ \texttt{F}$
  $\texttt{qi} = \lambda qt.\ q\ (t\ \texttt{T}\ \texttt{T})\ \texttt{T}$
  - достают из состояния новое значение (b), сдвиг (n) и индекс нового состояния (q_i), соответвующие значению в текущей ячейке

- $\texttt{oper} = \lambda qt.\ \langle \texttt{qi}\ q\ t, \texttt{xtnd}\ (\texttt{shift}\ (\texttt{rewr}\ t\ (\texttt{b}\ q\ t))\ (\texttt{n}\ q\ t)) \rangle$
  - получив состояние машины q и состояние ленты t, возвращает новое положение дел с перезаписанным значением под головкой и сдвигом

- $\texttt{step} = \lambda \delta s.\ \texttt{oper}\ (\texttt{get}\ (s\ \texttt{T})\ \delta)\ (s\ \texttt{F})$
  - шаг машины: преобразует таблицу состояний машины и положение дел в состояние машины и состояние ленты и возвращает новое положение дел

- $\texttt{eval} = \lambda e \delta s.\ (s\ \texttt{T})\ (e\ \delta\ (\texttt{step}\ \delta\ s))\ (s\ \texttt{F})$
  - шаг рекурсии: возвращает текущее состояние ленты, если машина в терминальном состоянии ($q_i = 0 = \texttt{F}$), или вычисляет новое состояние ленты и отправляет его дальше

- $\texttt{Interpret} = \lambda \delta s_0.\ (\lambda x.\ \texttt{eval}\ (x\ x)\ \delta\ s_0)\ (\lambda x.\ \texttt{eval}\ (x\ x)\ \delta\ s_0)$
  - обычный оператор неподвижной точки, зацикливающий `eval`