Курс: Функциональное программирование
Практика 2. Рекурсия и редукция

Разминка

▶ Найдите WHNF и NF для

$$\omega\ 2$$
$$\omega\ 3$$
$$\omega\ n$$

Напоминание: $\omega \equiv \lambda x.\, x\, x$. Замечание: мы больше не пишем черту над числами Черча.

Каррирование

Если функция двух аргументов задана в традиционном стиле $f\,(\mathtt{pair}\ x\ y)$ (на паре, т.е. декартовом произведении), то перейти к стандартной записи можно *каррированием*:

$$\mathtt{curry} \ \equiv\ \lambda f\, x\, y.\, f\,(\mathtt{pair}\ x\ y)$$

▶ Реализуйте обратную процедуру, `uncurry`.

Функция предшествования для чисел Чёрча

Вспомогательные функции

$$\begin{aligned} \mathtt{zp} &\equiv \mathtt{pair}\ 0\ 0 \\ \mathtt{sp} &\equiv \lambda p.\, \mathtt{pair}\ (\mathtt{snd}\ p)\ (\mathtt{succ}\ (\mathtt{snd}\ p)) \end{aligned}$$

Вторая работает так

$$\mathtt{sp}\ (\mathtt{pair}\ i\ j) \ =\ \mathtt{pair}\ j\ (j+1)$$

$$\begin{aligned} \mathtt{sp}^0\ (\mathtt{zp}) &= \mathtt{pair}\ 0\ 0 \\ \mathtt{sp}^m\ (\mathtt{zp}) &= \mathtt{pair}\ (m-1)\ m \end{aligned}$$

(здесь $m > 0$). Тогда функция предшествования:

$$\mathtt{pred} \ \equiv\ \lambda m.\, \mathtt{fst}\ (m\ \mathtt{sp}\ \mathtt{zp})$$

▶ Какая у неё временная сложность?
▶ Что нужно поменять, чтобы вышел факториал?

Числа Чёрча: **примитивная рекурсия**.
Обобщим предыдущую схему

$$
\begin{aligned}
\mathtt{xz} &\equiv \lambda x.\, \mathtt{pair}\ x\ 0 \\
\mathtt{fs} &\equiv \lambda f\, p.\, \mathtt{pair}\ (f\ (\mathtt{fst}\ p)\ (\mathtt{snd}\ p))\ (\mathtt{succ}\ (\mathtt{snd}\ p)) \\
\mathtt{rec} &\equiv \lambda m\, f\, x.\, \mathtt{fst}\ (m\ (\mathtt{fs}\ f)\ (\mathtt{xz}\ x))
\end{aligned}
$$

$$
\begin{tikzcd}
 & & & \langle & x, \arrow[dl] & 0\,\rangle \arrow[dl] \arrow[d, "\mathtt{succ}"] \\
 & & \langle f & x \arrow[dl] & 0, & 1\,\rangle \arrow[dl] \arrow[d, "\mathtt{succ}"] \\
 & \langle f\ (f & x \arrow[dl] & 0) & 1, & 2\,\rangle \arrow[dl] \arrow[d, "\mathtt{succ}"] \\
\langle f\ (f\ (f & x & 0) & 1) & 2, & 3\,\rangle
\end{tikzcd}
$$

В частности,

$$\mathtt{pred} \equiv \lambda m.\, \mathtt{rec}\ m\ (\lambda x\, y.\, y)\ 0$$

► Реализуйте факториал через комбинатор примитивной рекурсии `rec`.
► Реализуйте функцию суммирования чисел от 1 до n.
► Реализуйте функцию нахождения n-ой частичной суммы ряда $\sum_{k=1}^{n} f(k)$.

Конструкторы **списков** можно определить так:

$$
\begin{aligned}
\mathtt{nil} &\equiv \lambda c\, n.\, n \\
\mathtt{cons} &\equiv \lambda e\, l\, c\, n.\, c\, e\, (l\, c\, n)
\end{aligned}
$$

Например,

$$
\begin{aligned}
[\,] &\equiv \mathtt{nil} = \lambda c\, n.\, n \\
[5, 3, 2] &\equiv \mathtt{cons}\ 5\ (\mathtt{cons}\ 3\ (\mathtt{cons}\ 2\ \mathtt{nil})) = \lambda c\, n.\, c\ 5\ (c\ 3\ (c\ 2\ n))
\end{aligned}
$$

Функция, определяющая пуст ли список

$$\mathtt{empty} \equiv \lambda l.\, l\ (\lambda h\, t.\, \mathtt{fls})\ \mathtt{tru}$$

▶ Проверьте правильность работы `empty`.
▶ Попробуйте найти более «короткую» версию `empty`.
▶ Постройте функцию `head`, возвращающую голову списка, например

$$\texttt{head}\ [5,3,2] \ = \ 5$$

Комбинаторы неподвижной точки

▶ Используя комбинатор неподвижной точки Карри

$$\mathbf{Y} \ \equiv \ \lambda f.\,(\lambda x.\,f\,(x\,x))(\lambda x.\,f\,(x\,x))$$

найдите G, такой что $\forall X.\ G\,X \ =_\beta \ X\,(X\,G)$.

Хотя для комбинатора $\mathbf{Y}$ выполняется характеристическое уравнение $\mathbf{Y}\,F \ =_\beta \ F\,(\mathbf{Y}\,F)$, но неверно ни $\mathbf{Y}\,F \ \twoheadrightarrow_\beta \ F\,(\mathbf{Y}\,F)$, ни $F\,(\mathbf{Y}\,F) \twoheadrightarrow_\beta \ \mathbf{Y}\,F$:

$$\begin{aligned} \mathbf{Y}\,F \quad &\equiv \quad (\lambda f.\,(\lambda x.\,f\,(x\,x))(\lambda x.\,f\,(x\,x)))\,F \\ &\to_\beta \quad (\lambda x.\,F\,(x\,x))(\lambda x.\,F\,(x\,x)) \\ &\to_\beta \quad F((\lambda x.\,F\,(x\,x))(\lambda x.\,F\,(x\,x))) \ \to_\beta \ \ldots \end{aligned}$$

▶ Проверьте, что комбинатор неподвижной точки Тьюринга $\mathbf{\Theta}$

$$A \equiv \lambda x\,y.\,y\,(x\,x\,y), \ \ \mathbf{\Theta} \equiv A\,A$$

обладает одним из приведенных выше свойств редуцируемости.
▶ Найдите G, такой что $\forall X.\ G\,X \ \twoheadrightarrow_\beta \ X\,(X\,G)$.

▶ Докажите, что терм Ψ является комбинатором неподвижной точки тогда и только тогда, когда он представляет собой неподвижную точку комбинатора $\mathbf{S}\,\mathbf{I} = \lambda y\,z.\,z\,(y\,z)$.

Домашнее задание

▶ (1 балл) Приведите пример замкнутого чистого λ-терма находящегося
– в слабой головной нормальной форме, но не в головной нормальной форме;
– в головной нормальной форме, но не в нормальной форме.

Следующие три задачи нужно решить не используя комбинаторы неподвижной точки.

▶ (1 балл) В лямбда-исчислении реализуйте функции: `monus`, вычитающую числа Чёрча усеченным образом (truncated subtraction), `equals`, сравнивающую два числа Чёрча на предмет равенства, а также всевозможные неравенства, строгие и нестрогие, `lt`, `gt`, `le`, `ge`.

▶ (1 балл) В лямбда-исчислении реализуйте функции:
– `sum` суммирующую элементы списка, например

$$\texttt{sum}\ [5, 3, 2] \ = \ \mathsf{10}$$

– `length` вычисляющую длину списка, например

$$\texttt{length}\ [5, 3, 2] \ = \ \mathsf{3}$$

▶ (3 балла) В лямбда-исчислении реализуйте функцию `tail`, возвращающую хвост списка, например

$$\texttt{tail}\ [5, 3, 2] \ = \ [3, 2]$$

▶ (2 балла) Используя Y-комбинатор, сконструируйте
– «пожирателя», то есть такой терм `eater`, который для любого M обеспечивает $\texttt{eater}\,\mathsf{M} = \texttt{eater}$;
– «бюрократа», то есть такой терм `bureaucrat`, который любых термов M и N обеспечивает $\texttt{bureaucrat}\,\mathsf{M}\,\mathsf{N} = \texttt{bureaucrat}\,\mathsf{N}\,\mathsf{M}$;
– терм `qwerty` таким образом, чтобы для любых термов M и N выполнялось $\texttt{qwerty}\,\mathsf{M}\,\mathsf{N} = \mathsf{N}\,\texttt{qwerty}\,(\mathsf{N}\,\mathsf{M}\,\texttt{qwerty})$.

► (2 балла) Пусть имеются взаимно-рекурсивное определение термов `foo` и `bar`. В общем виде его можно записать так

$$
\begin{aligned}
\texttt{foo} &= \mathsf{P}\ \texttt{foo}\ \texttt{bar} \\
\texttt{bar} &= \mathsf{Q}\ \texttt{foo}\ \texttt{bar}
\end{aligned}
$$

Здесь P и Q — некоторые термы, не содержащие ни `foo`, ни `bar`. Используя $\mathbf{Y}$-комбинатор, найдите нерекурсивные определения для `foo` и `bar`. Постарайтесь найти максимально «компактное» решение, с наименьшим количеством $\mathbf{Y}$-комбинаторов.