Языки программирования. Семантика и система типов

Теоретическое задание. Тема 3

Бронников Егор

Задание 1. $(\lambda f.\,(f\,\{\mathrm{succ}\,0,\ \mathrm{iszero}\,(f\,\{0,\ \mathrm{false}\}).a\}).b)\,(\lambda t.\,\{a = t.1,\ b = t.2\})$

1. Дополнение пропущенных аннотаций.
$(\lambda f : \boldsymbol{\{a : Nat,\ b : Bool\}}.\,(f\,\{\mathrm{succ}\,0,\ \mathrm{iszero}\,(f\,\{0,\ \mathrm{false}\}).a\}).b)\,(\lambda t : \boldsymbol{Nat} \times \boldsymbol{Bool}.\,\{a = t.1,\ b = t.2\})$

2. Дерево вывода типа.

$$
\begin{prooftree}
\AxiomC{}
\UnaryInfC{$f\colon \mathit{\{a\colon Nat,\ b\colon Bool\}} \vdash f : \boldsymbol{\{a\colon Nat,\ b\colon Bool\}}$}
\AxiomC{}
\UnaryInfC{$f\colon \mathit{\{a\colon Nat,\ b\colon Bool\}} \vdash \mathrm{succ}\ 0 : \boldsymbol{Nat}$}
\AxiomC{}
\UnaryInfC{$f\colon \mathit{\{a\colon Nat,\ b\colon Bool\}} \vdash f : \boldsymbol{\{a\colon Nat,\ b\colon Bool\}}$}
\AxiomC{}
\UnaryInfC{$f\colon \mathit{\{a\colon Nat,\ b\colon Bool\}} \vdash 0 : \boldsymbol{Nat}$}
\AxiomC{}
\UnaryInfC{$f\colon \mathit{\{a\colon Nat,\ b\colon Bool\}} \vdash \mathrm{false} : \boldsymbol{Bool}$}
\BinaryInfC{$f\colon \mathit{\{a\colon Nat,\ b\colon Bool\}} \vdash \{0,\ \mathrm{false}\} : \boldsymbol{\{a\colon Nat,\ b\colon Bool\}}$}
\BinaryInfC{$f\colon \mathit{\{a\colon Nat,\ b\colon Bool\}} \vdash f\,\{0,\ \mathrm{false}\} : \boldsymbol{\{a\colon Nat,\ b\colon Bool\}}$}
\UnaryInfC{$f\colon \mathit{\{a\colon Nat,\ b\colon Bool\}} \vdash (f\,\{0,\ \mathrm{false}\}).a : \boldsymbol{Nat}$}
\UnaryInfC{$f\colon \mathit{\{a\colon Nat,\ b\colon Bool\}} \vdash \mathrm{iszero}\ (f\,\{0,\ \mathrm{false}\}).a : \boldsymbol{Bool}$}
\BinaryInfC{$f\colon \mathit{\{a\colon Nat,\ b\colon Bool\}} \vdash \{\mathrm{succ}\ 0,\ \mathrm{iszero}\ (f\,\{0,\ \mathrm{false}\}).a\} : \boldsymbol{\{a\colon Nat,\ b\colon Bool\}}$}
\BinaryInfC{$f\colon \mathit{\{a\colon Nat,\ b\colon Bool\}} \vdash (f\,\{\mathrm{succ}\ 0,\ \mathrm{iszero}\ (f\,\{0,\ \mathrm{false}\}).a\}).b : \boldsymbol{Bool}$}
\UnaryInfC{$\vdash \lambda f\colon \mathit{\{a\colon Nat,\ b\colon Bool\}}\,.\,(f\,\{\mathrm{succ}\ 0,\ \mathrm{iszero}\ (f\,\{0,\ \mathrm{false}\}).a\}).b : \boldsymbol{\{a\colon Nat,\ b\colon Bool\} \to Bool}$}
\AxiomC{}
\UnaryInfC{$t\colon \mathit{Nat} \times \mathit{Bool} \vdash t.1 : \boldsymbol{Nat}$}
\AxiomC{}
\UnaryInfC{$t\colon \mathit{Nat} \times \mathit{Bool} \vdash t.2 : \boldsymbol{Bool}$}
\BinaryInfC{$t\colon \mathit{Nat} \times \mathit{Bool} \vdash \{a = t.1,\ b = t.2\} : \boldsymbol{\{a\colon Nat,\ b\colon Bool\}}$}
\UnaryInfC{$\vdash (\lambda t\colon \mathit{Nat} \times \mathit{Bool}\,.\,\{a = t.1,\ b = t.2\}) : \boldsymbol{(Nat \times Bool) \to \{a\colon Nat,\ b\colon Bool\}}$}
\BinaryInfC{$\vdash (\lambda f\colon \mathit{\{a\colon Nat,\ b\colon Bool\}}\,.\,(f\,\{\mathrm{succ}\ 0,\ \mathrm{iszero}\ (f\,\{0,\ \mathrm{false}\}).a\}).b)\ (\lambda t\colon \mathit{Nat} \times \mathit{Bool}\,.\,\{a = t.1,\ b = t.2\}) : \boldsymbol{(Nat \times Bool) \to Bool}$}
\end{prooftree}
$$

Примечание. Рекомендуется изучить исходный Excel-документ `source.xlsx`.

Задание 2.

Условие. В нетипизированном λ-исчисления, пары термов могут быть представлены при помощи кодировки Чёрча. Можно ли использовать это представление, чтобы представить пары как производную форму поверх простого типизированного λ-исчисления с логическими и арифметическими выражениями?

Ответ. Да, можно.
Для начала стоит напомнить определение пары в кодировке Чёрча в нетипизированном λ-исчислении:

$$
\begin{gathered}
pair := \lambda x.\,\lambda y.\,\lambda z.\,z\,x\,y \\
first := \lambda p.\,p\,(\lambda x.\,\lambda y.\,x) \\
second := \lambda p.\,p\,(\lambda x.\,\lambda y.\,y)
\end{gathered}
$$

Теперь перейдём к описанию определение пары в кодировке Чёрча в простом типизированном λ-исчислении:

$$
\begin{gathered}
pair := \lambda x : T_1.\,\lambda y : T_2.\,\lambda z : (T_1 \to T_2 \to T)\,.\,z\,x\,y \\
first := \lambda p : (T_1 \times T_2)\,.\,p\,(\lambda x : T_1\,.\,\lambda y : T_2\,.\,x) \\
second := \lambda p : (T_1 \times T_2)\,.\,p\,(\lambda x : T_1\,.\,\lambda y : T_2\,.\,y) \\
(T_1 \times T_2) := (T_1 \to T_2 \to T \to T)
\end{gathered}
$$

a. Функция раскрытия сокращения.

- *Пара*

  $e(pair) = e(\lambda x : T_1 . \lambda y. : T_2\, \lambda z : (T_1 \to T_2 \to T) \,.\, z\, x\, y) = \lambda x : T_1 . \lambda y : T_2 . \lambda z : (T_1 \to T_2 \to T) \,.\, e(z)\, e(x)\, e(y)$

- *Проекции*

  $e(first) = e(\lambda p : (T_1 \times T_2) \,.\, p\, (\lambda x : T_1 \,.\, \lambda y : T_2 \,.\, x)) = \lambda p : (T_1 \times T_2) \,.\, p\, (\lambda x : T_1 \,.\, \lambda y : T_2 \,.\, e(x))$
  $e(second) := e(\lambda p : (T_1 \times T_2) \,.\, p\, (\lambda x : T_1 \,.\, \lambda y : T_2 \,.\, y)) = \lambda p : (T_1 \times T_2) \,.\, p\, (\lambda x : T_1 \,.\, \lambda y : T_2 \,.\, e(y))$

- *Тип-произведения*

  $e(T_1 \times T_2) = e(T_1 \to T_2 \to T \to T) = e(T_1) \to e(T_2) \to e(T) \to e(T)$

b. Сохранение вычисления и типизации

- *Сохранение вычисления:* Да, функция раскрытия сокращений сохраняет вычисление, поскольку просто переводит выражения из одного представления в другое, не меняя семантику.

- *Сохранение типизации:* Да, сохраняется. Каждое выражение в новом представлении имеет тип, который зависит от типов $first$ и $second$, а также типов проекций, что отражается в определении новых функций.

Таким образом, функция раскрытия сокращений сохраняет и вычисление, и типизацию. Если бы это не работало, это могло бы быть из-за несоответствия типов или неправильной интерпретации функций проекции. Например, если бы мы неправильно определили тип $T_1 \times T_2$ или проекции $first$ и $second$, это могло бы привести к нарушению типизации.