Fpl – functional programming language

- Дополним язык Exp4 возможностью декларирования взаимно-рекурсивных функций вида:
  $\mathbf{f_1(x_1,\ldots x_{k1})\ \texttt{<=}\ e_1}$
  …
  $\mathbf{f_n(x_1,\ldots x_{kn})\ \texttt{<=}\ e_n}$

- Это список определений функций: $\mathbf{f_i}$ – имя функции, $\mathbf{e_i}$ – её тело, а $\mathbf{(x_1,\ldots x_{ki})}$ – список параметров.
- Новый язык будем называть Fpl .

Абстрактный синтаксис языка Fpl

1) Синтаксические категории

```
p  ∈  Prog          op ∈  Op
D  ∈  Dec           bop∈  BOp
e  ∈  Exp       x  ∈  Var
be ∈  BExp          bx ∈  BVar
f  ∈  FunVar    n  ∈  Num
```

2) Определения

$$
\begin{aligned}
&\mathbf{p} ::= \langle \mathbf{e},\mathbf{D} \rangle\\
&\mathbf{D} ::= \mathbf{f}(\mathbf{x}_1,\ldots\mathbf{x}_k) <= \mathbf{e} \mid \mathbf{f}(\mathbf{x}_1,\ldots\mathbf{x}_k) <= \mathbf{e},\ \mathbf{D}'\\
&\mathbf{op} ::= + \mid - \mid * \mid \mathbf{div}\\
&\mathbf{bop} ::= \mathbf{And} \mid \mathbf{Or}\\
&\mathbf{e} ::= \mathbf{x} \mid \mathbf{n} \mid \mathbf{e}'\ \mathbf{op}\ \mathbf{e}'' \mid \mathbf{let}\ \mathbf{x}=\mathbf{e}'\ \mathbf{in}\ \mathbf{e}'' \mid\\
&\quad \mathbf{if}\ \mathbf{be}\ \mathbf{then}\ \mathbf{e}'\ \mathbf{else}\ \mathbf{e}'' \mid\\
&\quad \mathbf{f}(\mathbf{e}_1,\ldots\mathbf{e}_k),\ \text{где } \mathbf{k} - \text{арность } \mathbf{f}\\
&\mathbf{be} ::= \mathbf{bx} \mid \mathbf{T} \mid \mathbf{F} \mid \mathbf{Not}\ \mathbf{be}' \mid \mathbf{Equal}\ (\mathbf{e},\mathbf{e}')\\
&\quad \mid \mathbf{be}'\ \mathbf{bop}\ \mathbf{be}''
\end{aligned}
$$

Ограничение

В каждой программе <e,D> должны выполняться условия:

1. Каждое имя функции, встречающееся в **e** должно иметь определение в **D**.
2. Каждое имя функции может иметь только одно определение в **D**.

Отношение $\Rightarrow_A$ для языка Fpl

- Записывается
  $\mathbf{D}, \boldsymbol{\rho} \vdash \mathbf{e} \Rightarrow_A \mathbf{v}$,
  а читается так:
  «Если продекларировано **D** , то в окружении **ρ** выражение **e** даст арифметический результат **v**»
- Имеет тип:
  $\Rightarrow_A$ **`: Dec → ENV → Exp → Num`**
- Декларации влияют и на вычисление булевых значений

Отношение $\Rightarrow_B$ для языка Fpl

- Декларации влияют и на вычисление булевых значений. Например, выражение **`Equal(f(e),g(e'))`** – булево и использует функции f и g. Следовательно отношение $\Rightarrow_B$ имеет тип:

  $\Rightarrow_B$ **`: Dec → ENV → BExp → {T,F}`**

Результат работы программы

Отношения $\Rightarrow_{\mathbf{A}}$ и $\Rightarrow_{\mathbf{B}}$ позволяют определить результат программы <e,D>.

$$\boldsymbol{\rho} \vdash \langle \mathbf{e}, \mathbf{D} \rangle \Rightarrow \mathbf{v}$$

если

$$\mathbf{D}, \boldsymbol{\rho} \vdash \mathbf{e} \Rightarrow_{\mathbf{A}} \mathbf{v}$$

Естественная семантика языка Fpl

- Все правила аналогичны правилам языка Exp4. Добавлено только одно новое правило FunR

✔ Правило FunR

$$\frac{\mathbf{D},\rho \vdash \mathbf{e}_i \Rightarrow_A \mathbf{v}_i,\ 1 \le i \le k \qquad \mathbf{D},\rho[\mathbf{x}_1/\mathbf{v}_1,\ldots \mathbf{x}_k/\mathbf{v}_k] \vdash \mathbf{e} \Rightarrow_A \mathbf{v}}{\mathbf{D},\rho \vdash \mathbf{f}(\mathbf{e}_1,\ldots \mathbf{e}_k) \Rightarrow_A \mathbf{v}} \quad [\,\mathbf{f}(\mathbf{x}_1,\ldots \mathbf{x}_k) <= \mathbf{e} \in \mathbf{D}]$$

Способы передачи параметров

По правилу FunR для вычисления **f(e1,…ek)** сначала нужно вычислить параметры **e1,…ek**, а потом тело функции из определения **f** в окружении, в которое добавлены связи формальных параметров с действительными. Это ***передача параметров по значению (call by value).***

Но можно передавать параметры не вычисляя, просто подставляя выражения в тело функции. Это ***передача параметров по имени (call by name).***

Передача параметров по значению используется в ***строгих*** функциональных языках, а передача параметров по имени – в ***ленивых***.

Передача параметров по имени

- ✔ Правило $FunR_{le}$

$$\frac{D, \rho \vdash e[x_1/e_1, \ldots x_k/e_k] \Rightarrow_A v}{D, \rho \vdash f(e_1, \ldots e_k) \Rightarrow_A v} \quad [\, f(x_1, \ldots x_k) <= e \in D]$$

Теоремы

- Для языка Fpl нельзя доказать теорему, что для каждого выражения **`e`** $\in$ **`Fpl`** можно найти результат, поскольку если **`f(x) <= f(x+1)`** $\in$ **`D`**, то выражение, содержащее вызов **`f`** будет вычисляться бесконечно.
- Теорема об уникальности результата справедлива.
- Для каждого выражения **`e`** $\in$ **`Fpl,`** окружения ρ и программы **`p=<e,D>`**, из $\rho \vdash$ **`p`** $\Rightarrow$ **`n`** и $\rho \vdash$ **`p`** $\Rightarrow$ **`n'`** следует, что **`n=n'.`**
  В доказательстве метод структурной индукции неприменим. Нужно использовать метод индукции по дереву вывода.

Случай EqR

$$\frac{\mathbf{D,\rho} \vdash \mathbf{e} \Rightarrow_{\mathbf{A}} \mathbf{v} \quad \mathbf{D,\rho} \vdash \mathbf{e'} \Rightarrow_{\mathbf{A}} \mathbf{v}}{\mathbf{D,\rho} \vdash \mathbf{Equal(e,e')} \Rightarrow_{\mathbf{B}} \mathbf{T}}$$

По индукции предположим:

$\mathbf{P_A(\rho,e,v)}$, $\mathbf{P_A(\rho,e',v)}$. **(1),(2)**

Требуется доказать:

$\mathbf{P_B(\rho,Equal(e,e'),T)}$.

То есть что из:

$\mathbf{D,\rho} \vdash \mathbf{Equal(e,e')} \Rightarrow_{\mathbf{B}} \mathbf{bv'}$

следует **bv'=T** .

Для вывода **bv'** можно было использовать только правило EqR. Поэтому должны существовать два числа w и w' такие, что : $\mathbf{D,\rho} \vdash \mathbf{e} \Rightarrow_{\mathbf{A}} \mathbf{w}$, $\mathbf{D,\rho} \vdash \mathbf{e'} \Rightarrow_{\mathbf{A}} \mathbf{w'}$.

Случай EqR

Для вывода **bv'** можно было использовать только правило EqR. Поэтому должны существовать два числа w и w' такие, что :

$$\mathbf{D,\rho} \vdash \mathbf{e} \Rightarrow_{\mathbf{A}} \mathbf{w} \quad ,$$

$$\mathbf{D,\rho} \vdash \mathbf{e'} \Rightarrow_{\mathbf{A}} \mathbf{w'}.$$

Из свойств (1) и (2) следует соответственно **w=v** и **w′=v**.

Далее по правилу **EqR** имеем **bv′=T**.

Случай LocR

$$\frac{\mathbf{D},\rho \vdash \mathbf{e} \Rightarrow_A \mathbf{v} \quad \mathbf{D},\rho[\mathbf{v}/\mathbf{x}] \vdash \mathbf{e}' \Rightarrow_A \mathbf{w}'}{\mathbf{D},\rho \vdash \mathbf{let\ x{=}e\ in\ e}' \Rightarrow_A \mathbf{w}'}$$

По индукции предположим:

$\mathbf{P}_A(\rho,\mathbf{e},\mathbf{v})$, (1)

$\mathbf{P}_A(\rho[\mathbf{v}/\mathbf{x}],\mathbf{e}',\mathbf{w}')$. (2)

Требуется доказать:

$\mathbf{P}_A(\rho,\mathbf{let\ x{=}e\ in\ e}',\mathbf{w}')$.

То есть что из:

$\mathbf{D},\rho \vdash \mathbf{let\ x{=}e\ in\ e}' \Rightarrow_A \mathbf{v}'$

следует $\mathbf{v}'=\mathbf{w}'$.

Случай LocR

Для вывода **v'** можно было использовать только правило LocR. Поэтому должно существовать число w такое, что :

$$\mathbf{D,\rho \vdash e \Rightarrow_A w}\ , \qquad (3)$$

$$\mathbf{D,\rho[w/x] \vdash e' \Rightarrow_A v'}. \qquad (4)$$

Из свойства (1) и (3) получим **w=v**.

Из свойства (2) и (4) получим **v′=w′**.

Случай FunR

$$\frac{D,\rho \vdash e_i \Rightarrow_A v_i,\ 1 \le i \le k \qquad D,\rho[v/x] \vdash e' \Rightarrow_A w'}{D,\rho \vdash f(e_1,\ldots e_k) \Rightarrow_A v}$$

По индукции предположим:

$P_A(\rho, e_i, v_i), 1 \le i \le k$ (1)

$P_A(\rho[v_1/x_1, \ldots v_k/x_k], e, v).$ (2)

Требуется доказать:

$P_A(\rho, f(e_1, \ldots e_k), v).$

Случай FunR

Допустим, что:

$\mathbf{D,\rho \vdash f(e_1,...e_k) \Rightarrow_A v'}$

тогда достаточно доказать, что $\mathbf{v=v'}$.

Для вывода $\mathbf{v'}$ можно было использовать только правило FunR. Поэтому должны существовать числа $\mathbf{v_1',...v_k'}$ такие, что : $\mathbf{D,\rho \vdash e_i \Rightarrow_A v_i'}$, $1 \le i \le k$,
и при этом

$\mathbf{D,\rho[v_1'/x_1,...v_k'/x_k] \vdash e \Rightarrow_A v'}$ **(3).**

Из свойств (1) следует $\mathbf{v_i' = v_i}$,

а из (2) и (3) получим $\mathbf{v'=v}$.