Лекция 13

Инсерционное моделирование
Функциональное и алгебраическое программирование

Функциональное программирование

Синтаксис

$$f_i(x_1, x_2, \ldots) = E_i(f, x),$$
$$i = 1, \ldots n, x = (x_1, x_2, \ldots)$$

$$T_\Omega(C, V)$$
$$T_\Omega(C)$$

Типы данных

$$T_{\Xi,\Omega}(C, V)$$
$$C = (C_\xi)_{\xi \in \Xi}$$
$$V = (V_\xi)_{\xi \in \Xi}$$
$$f : (\xi_1, \ldots, \xi_n) \to \xi$$

Денотационная семантика

Область D

Семантика константных выражений: $[[A]] \in D$

Семантика функций: $[[f_i]] \in D^{n_i} \to D$

Наименьшая неподвижная точка функционала

$$(f_1, \ldots, f_n) \mapsto (F_1(f), \ldots, F_n(f))$$

$$F_i(f) = \lambda x E_i(f, x)$$

Функциональные языки высших степеней

Язык функциональных выражений обогащается введением λ-абстракции. Семантика Дана Скотта, Хенкина и др. Функциональные типы данных.

Операционная семантика

Подстановка: $E' := E[x_1 := E_1, x_2 := E_2, \ldots]$

Упрощение: $simpl(E)$,

$2*2=4, x+0=x, \ldots$

Вызов функции

z входит один раз

Переходы:

$$\frac{F = G[z := f_i(t_1, t_2, \ldots)]}{F \rightarrow simpl(G[z := E_i[x_1 := t_1, x_2 := t_2, \ldots]])}$$

$$F \rightarrow simpl(F)$$

Теорема

$$[[A_1]] = d_1, [[A_2]] = d_2, \ldots \Rightarrow$$
$$\Rightarrow [[f_i]](A_1, A_2, \ldots) = d \Leftrightarrow f_i(A_1, A_2, \ldots) \vec{\text{¿}} B, [[B]] = d$$

Используется теорема о неподвижной точке:

$$f_i(x_1, x_2, \ldots) = \text{¿} f_i^{(n)}(x)$$
$$f_i^{(0)}(x) = \text{¿}$$
$$f_i^{(n+1)}(x) = E_i(f^{(n)}, x)$$

Стратегии вычислений
(семантики и языки)

- Call-by-value (eager)
- Call-by name (lazy)
- Parallel,…

Эффективность и полнота

Операционная семантика

параллельного функционального программирования

$$\frac{t = G[\,x_1 := G_1, x_2 := G_2, \ldots\,]}{t \rightarrow simpl\left(G[\,x_1 := G_1^{'}, x_2 := G_2^{'}, \ldots\,]\right)}$$

$$\frac{t = G[\,x_1 := G_1, x_2 := G_2, \ldots\,], G_1 \rightarrow F_1, G_2 \rightarrow F_2, \ldots}{t \rightarrow simpl\left(G[\,x_1 := F_1^{'}, x_2 := F_2^{'}, \ldots\,]\right)}$$

E **– вызов,** ***E'*** **– развертывание**

$G_i \rightarrow F_i$ **– одношаговый переход**

для внутренних вызовов (рекурсивно)

Алгебраическое программирование

Синтаксис

$$T_{\Omega}(C,V)$$

$$T_{\Omega}(C)$$

$$rs(x_1, x_2, \ldots)(s_1 = t_1, s_2 = t_2, \ldots)$$

Операционная семантика

$$t \rightarrow t' \qquad t \leftrightarrow t'$$

x входит один раз

$$\frac{t = G[x := s],\ s = s_i[x_1 := u_1, x_2 := u_2, \ldots]}{t \rightarrow G[x := t_i[x_1 := u_1, x_2 := u_2, \ldots]]}$$

$1)\ t \overset{\rightarrow}{¿} t'$

Семантика переписывания

$2)\ t \overset{\rightarrow}{¿} t'$

Эквациональная семантика

Денотационная семантика

Каноническая форма = единственность

Нормализация

$t \vec{¿}\, t'$ ¿⇔

Факторизация

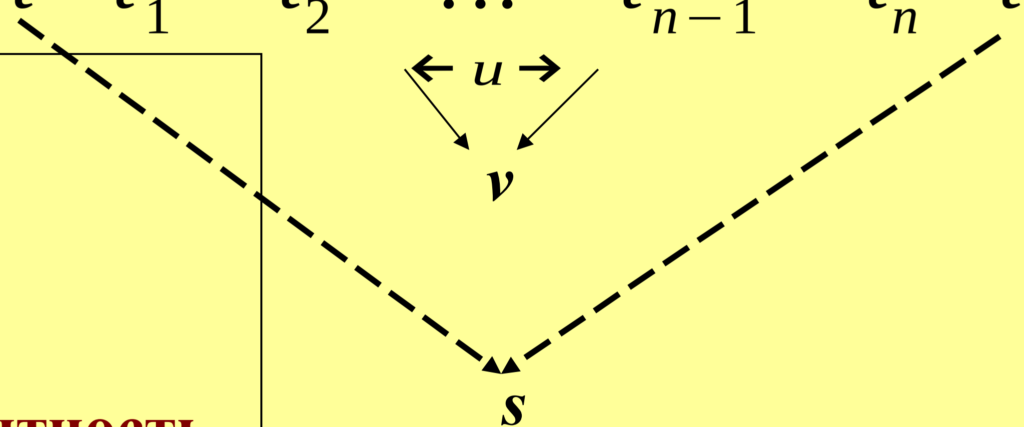
$t \sim t' (R) \Leftrightarrow t \vec{¿}\, t'$

Теорема Черча –Россера

Всякий терм имеет каноническую форму=>

$$t \vec{¿}\, t' \Longleftrightarrow t \vec{¿}\, s \vec{¿}\, t'$$

$$t = t_1 \leftrightarrow t_2 \leftrightarrow \ldots \leftrightarrow t_{n-1} \leftrightarrow t_n = t'$$



$$\begin{tikzcd} & t \arrow[dl, "*"'] \arrow[dr, "*"] & \\ t_1 \arrow[dr, dashed, "*"'] & & t_2 \arrow[dl, dashed, "*"] \\ & s & \end{tikzcd}$$

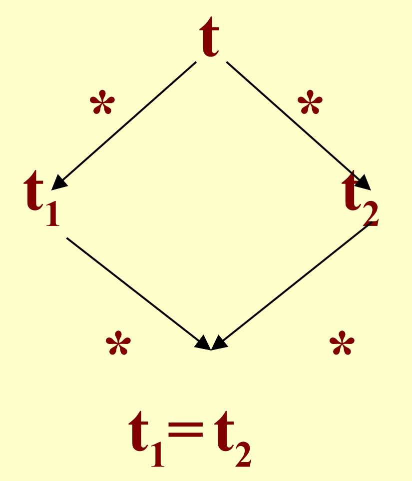
конфлюэнтность

Канонические системы

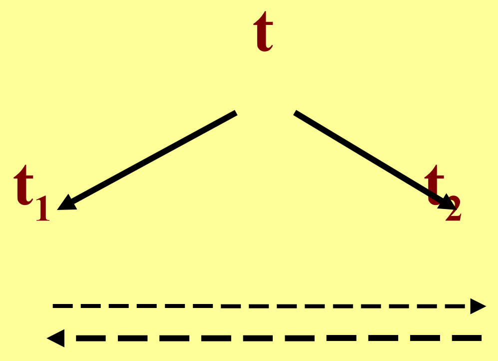
Конфлюэнтность =>
н.ф. существует => единственна

Нетеровость=> н.ф. существует

Каноничность<=>
Нетеровость + конфлюэнтность



Алгоритм Кнута-Бендикса



Мат. логика в программировании, М. 1991

Примеры

$$
\begin{aligned}
&rs(x_1, x_2, \ldots)(\\
&f_1(x_1, x_2, \ldots) = E_1,\\
&f_2(x_1, x_2, \ldots) = E_2,\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
&);
\end{aligned}
$$

$$
\begin{aligned}
&rs(x, y, \ldots)(\\
&\ldots\ldots\ldots\ldots\ldots\ldots\\
&if(1, x, y) = x,\\
&if(0, x, y) = y\\
&);
\end{aligned}
$$

$$n \Rightarrow s^n(0)$$

$$
\begin{aligned}
&rs(x, y)(\\
&s(x) + y = s(x + y),\\
&0 + x = x\\
&);
\end{aligned}
$$

Расширения

- Условные правила
  - $p(x,y,\ldots)\to(t(x,y,\ldots) = s(x,y,\ldots))$
  - **Сопоставление** $u = t(x,y,\ldots)$, $x = a$, $y = b,\ldots$
  - **Упрощение** $p(a,b,\ldots) = 1,0,\ldots$
- Несвободная алгебра $T_\Omega(C)$
  - **Решение уравнений** $u = t(x,y,\ldots)$, $x = a$, $y = b,\ldots$
- Упрощения после переписывания
- Ассоциативно коммутативное переписывание
- Типы (ML)

Система APS

Основные особенности:

- Интеграция основных парадигм
- Переписывание с последующей канонизацией
- Использование стратегий переписывания
- Алгебраические структуры данных
  (графовые термы)
- Развитие:
  - инсерционное программирование
  - автоматическое доказательство теорем
  - инструмент для верификации софтвера

Язык APLAN

описания имен и отметок
начальные присваивания
инклуды

Алгебраические программы
Алгебраические выражения
Переписывающие правила
Процедуры
Базовые операторы
Вычисление значений
Приведение к канонической форме

присваивание
установка
while, if
Вызовы
compile