Функциональное программирование

ЛЕКЦИЯ 2. ТЕМА 2. БЕСТИПОВЫЕ АРИФМЕТИЧЕСКИЕ ВЫРАЖЕНИЯ

План лекции

1. Термы.
2. Синтаксис.
3. Индукция на термах.
4. Семантические стили.

НЕОБХОДИМОСТЬ ИЗУЧЕНИЯ ТЕОРЕТИЧЕСКИХ ОСНОВ ФП

- Базовые характеристики языков программирования (во многом определяются системой типов в ФП)
- Четкие, ясные, точные механизмы описания синтаксиса и семантики программ

Пример: язык Haskell

Поскольку Haskell представляет собой чисто функциональный язык,

- все вычисления осуществляются с помощью исчисления выражений (синтаксических термов),
- производящих значения (абстрактные сущности, которые мы рассматриваем как ответы).

Каждое значение имеет связанный с ним тип (на интуитивном уровне мы можем рассматривать типы как множества значений).

Налицо построенная строгая система типов закономерности и свойства которой необходимо учитывать, уметь формально обращаться с ней

ГРАММАТИКА ПСЕВДО-ЯЗЫКА

```
t ::=                          {- термы: -}
    true                 {- константа «истина» -}
    false                {- константа «ложь» -}
    if t then t else t   {- условное выражение -}
    0                {- константа «ноль» -}
    succ t           {- следующее число -}
    pred t           {- предыдущее число -}
    iszero t         {- проверка на ноль -}
```

ОПРЕДЕЛЕНИЯ И ПОЯСНЕНИЯ

t (c p до u)– метапеременная (служит для описания), служит заместителем какого-либо терма

Выражением будут называться любые синтактические объекты

Термы – выражения, которые представляют собой вычисления

Программа – терм, построенный на основе конструкций соответствующей приведенной грамматике

Числа - конструкции вида succ(succ(succ(0)))

Результаты вычислений – это либо булевы константы, либо числа - это все термы

Такие термы будем называть **значениями**

ДРУГИЕ ОПРЕДЕЛЕНИЯ СИНТАКСИСА

Определение 1 [термы через индукцию]

Множество термов – это наименьшее множество *T* такое , что

1. {true; false; 0} □ ⊆ T ;
2. если t1 ∈ T , то {succ t1; pred t1; iszero t1} □ ⊆T ;
3. если t1 ∈ T , t2 ∈ T , t3 ∈ T , то if t1 then t2 else t3 ∈ T .

ДРУГИЕ ОПРЕДЕЛЕНИЯ СИНТАКСИСА

Определение 2 [термы через правила вывода]: множество термов определяется следующим образом

$$true \in \mathcal{T} \qquad false \in \mathcal{T} \qquad 0 \in \mathcal{T}$$

$$\frac{t_1 \in \mathcal{T}}{succ\ t_1 \in \mathcal{T}} \qquad \frac{t_1 \in \mathcal{T}}{pred\ t_1 \in \mathcal{T}} \qquad \frac{t_1 \in \mathcal{T}}{iszero\ t_1 \in \mathcal{T}}$$

$$\frac{t_1 \in \mathcal{T} \qquad t_2 \in \mathcal{T} \qquad t_3 \in \mathcal{T}}{if\ t_1\ then\ t_2\ else\ t_3 \in \mathcal{T}}$$

ДРУГИЕ ОПРЕДЕЛЕНИЯ СИНТАКСИСА

- Определение 3 [термы через правила вывода]: множество термов определяется объединением множеств $S=\bigcup_i S_i$ для каждого $i \in N$, где

$S_0=\varnothing$

$S_{i+1}=$ {true, false,0}

$\cup$ {succ t_1, pred t_1, iszero t_1 | $t_1 \in S_i$}

$\cup$ { if t_1 then t_2 else t_3 | $t_1, t_2, t_3 \in$ Si}

$T=S$!!

ДОМАШНЕЕ ЗАДАНИЕ:

1. Сколько элементов содержит S_3?
2. Покажите $\forall\ i, S_i \subseteq S_{i+1}$
3. Покажите, что S - наименьшее множество

T=S !!

Проверим!!!!

Для S должны выполняться условия

1. {true; false; 0} $\subseteq$ S ;
2. если t1 $\in$ S , то {succ t1; pred t1; iszero t1} $\subseteq$ S ;
3. если t1 $\in$ S , t2 $\in$ S , t3 $\in$ S , то if t1 then t2 else t3 $\in$ S .

T=S !!

Докажем!!!

1. Пусть S’ удовлетворяет условиям наименьшего множества
2. При помощи полной индукции по i докажем что $S_i \subseteq S'$

- Для случая i=0
- Для случая i=j+1>0, пусть $S_j \subseteq S'$

ИНДУКЦИЯ НА ТЕРМАХ

Из определения 1 если $t \in T$

1. t является константой;

2. t результатом succ t_1; pred t_1; iszero t_1 t_1<t;

3. t результатом if t_1 then t_2 else t_3 t_1, t_2, t_3 <t

ОПРЕДЕЛЕНИЯ

Определение 1 *Множество констант, встречающихся в терме t (записывается $Consts(t)$), определяется так:*

$$
\begin{aligned}
Consts(\mathtt{true}) &= \{\mathtt{true}\} \\
Consts(\mathtt{false}) &= \{\mathtt{false}\} \\
Consts(\mathtt{0}) &= \{\mathtt{0}\} \\
Consts(\mathtt{succ}\ t_1) &= Consts(t_1) \\
Consts(\mathtt{pred}\ t_1) &= Consts(t_1) \\
Consts(\mathtt{iszero}\ t_1) &= Consts(t_1) \\
Consts(\mathtt{if}\ t_1\ \mathtt{then}\ t_2\ \mathtt{else}\ t_3) &= Consts(t_1) \cup Consts(t_2) \cup Consts(t_3)
\end{aligned}
$$

ОПРЕДЕЛЕНИЯ

Определение 2 Размер *(size)* терма t *(записывается* $size(t)$*)* определяется так:

$$
\begin{aligned}
size(\mathtt{true}) &= 1 \\
size(\mathtt{false}) &= 1 \\
size(\mathtt{0}) &= 1 \\
size(\mathtt{succ}\ t_1) &= size(t_1) + 1 \\
size(\mathtt{pred}\ t_1) &= size(t_1) + 1 \\
size(\mathtt{iszero}\ t_1) &= size(t_1) + 1 \\
size(\mathtt{if}\ t_1\ \mathtt{then}\ t_2\ \mathtt{else}\ t_3) &= size(t_1) + size(t_2) + size(t_3) + 1
\end{aligned}
$$

ОПРЕДЕЛЕНИЯ

Определение 3 глубина *(depth)* терма t *(записывается* $depth(t)$*)*

определяется так:

$$
\begin{aligned}
depth(\texttt{true}) &= 1 \\
depth(\texttt{false}) &= 1 \\
depth(\texttt{0}) &= 1 \\
depth(\texttt{succ}\ t_1) &= depth(t_1) + 1 \\
depth(\texttt{pred}\ t_1) &= depth(t_1) + 1 \\
depth(\texttt{iszero}\ t_1) &= depth(t_1) + 1 \\
depth(\texttt{if}\ t_1\ \texttt{then}\ t_2\ \texttt{else}\ t_3) &= \max(depth(t_1), depth(t_2), depth(t_3)) + 1
\end{aligned}
$$

ПРИМЕР СООТНОШЕНИЯ МЕЖДУ ЧИСЛОМ КОНСТАНТ В ТЕММЕ И ЕГО РАЗМЕРОМ

Лемма 1 *Число различных констант в терме* t *не больше его размера* (*т. е.,* $|Consts(t)| \leqslant size(t)$).

ПРИНЦПЫ ИНДУКЦИИ ПО ТЕРМАМ

Индукция по глубине

Пусть P — предикат,
определенный на множестве термов.

Если для каждого терма s,

предполагая, что $P(r)$ для всех r, таких, что
$depth(r) < depth(s)$,
можно доказать, что $P(s)$,

то $P(s)$ выполняется для всех s.

ПРИНЦПЫ ИНДУКЦИИ ПО ТЕРМАМ

Индукция по размеру

Пусть P — *предикат,*
определенный на множестве термов.

Если для каждого терма s,
предполагая, что $P(r)$ *для всех* r, *таких, что* $size(r) < size(s)$,
можно доказать, что $P(s)$,
то $P(s)$ *выполняется для всех* s.

ПРИНЦПЫ ИНДУКЦИИ ПО ТЕРМАМ

Структурная индукция

Пусть P — предикат,

определенный на множестве термов.

Если для каждого терма s,

предполагая, что $P(r)$ для всех непосредственных подтермов s, можно доказать, что $P(s)$,

то $P(s)$ выполняется для всех s.

О СИНТАКСИСЕ ПСЕВДО-ЯЗЫКА (ПОВТОР)

```
t ::=                          {- термы: -}
    true                 {- константа «истина» -}
    false                {- константа «ложь» -}
    if t then t else t    {- условное выражение -}
    0                {- константа «ноль» -}
    succ t           {- следующее число -}
    pred t           {- предыдущее число -}
    iszero t         {- проверка на ноль -}
```

О СИНТАКСИСЕ ПСЕВДО-ЯЗЫКА (ПОВТОР)

Множество термов – это наименьшее множество *T* такое , что

1. {true; false; 0} □ ⊆ T ;
2. если t1 ∈ T , то {succ t1; pred t1; iszero t1} □ ⊆T ;
3. если t1 ∈ T , t2 ∈ T , t3 ∈ T , то if t1 then t2 else t3 ∈ T .

Множество термов определяется объединением множеств $S=\bigcup_i S_i$ для каждого i∈N, где

S0=∅

Si+1= {true, false,0}

∪{succ t1, pred t1, iszero t1 | t1∈ Si}

∪{ if t1 then t2 else t3 | t1, t2, t3∈ Si}

СЕМАНТИЧЕСКИЕ СТИЛИ

Для определения смысла программы необходим её математический эквивалент, а для его построения нам надо в точности знать свойства языка, на котором написана программа.

Проблема сводится к отысканию способа построения математических эквивалентов всех конструкций языка - к формализации его семантики.

СЕМАНТИЧЕСКИЕ СТИЛИ

Семантика языка — это смысловое значение слов. В программировании — начальное смысловое значение операторов, основных конструкций языка и т. п.

1) `i=0; while(i<5){i++;}`

2) `i=0; do{i++;}while(i<4);`

СЕМАНТИЧЕСКИЕ СТИЛИ

Различные подходы к формализации семантики:

- Операционная семантика
- Денотационая семантика
- Аксиоматическая семантика
- Интерпретационная семантика
- Трансляционная семантика
- Трансформационная семантика

СЕМАНТИЧЕСКИЕ СТИЛИ

Операционная семантика

Специфицирует поведение языка определяя простую абстрактную машину (использует термы языка, а не машинные команды)

Состояние машины – терм

Поведение определяется **функцией перехода**.

Смысл терма *t* – конечное состояние

СЕМАНТИЧЕСКИЕ СТИЛИ

Трансляционная семантика

Описание операционной семантики конструкций в терминах языков программирования высокого уровня.

С помощью этого способа можно изучать язык, схожий с уже известным программисту.

СЕМАНТИЧЕСКИЕ СТИЛИ

Трансформационная семантика

Описание операционной семантики конструкций языка в терминах этого же языка. Трансформационная семантика является основой метапрограммирования.

СЕМАНТИЧЕСКИЕ СТИЛИ

Денотационная семантика

Смысл терма - математические объекты (число, функция, их величины)

Построение семантики:

1) нахождение набора семантических доменов (СД)
2) Определение функции интерпретации – (соответствие СД и термов)

СЕМАНТИЧЕСКИЕ СТИЛИ

Аксиоматическая семантика

Семантика каждой синтаксической конструкции языка определяется как некий набор аксиом или правил вывода, который можно использовать для вывода результатов выполнения этой конструкции.

Чтобы понять смысл всей программы, эти аксиомы и правила вывода следует использовать так же, как при доказательстве обычных математических теорем.

Смысл терма, то что можно о нем доказать.

Когда программа выполнена, получаем доказательство - что вычисленные результаты удовлетворяют необходимым ограничениям на их значения относительно входных значений. То есть, доказано, что выходные данные представляют значения соответствующей функции, вычисленной по значениям входных данных.

СЕМАНТИЧЕСКИЕ СТИЛИ

Интерпретационная семантика

описание операционной семантики конструкций в терминах языков программирования низкого уровня (язык ассемблера, машинный код).

Позволяет выявлять медленно выполняемые участки программы, и зачастую используется в целях оптимизации кода программ.

ВЫЧИСЛЕНИЯ

$\mathbb{B}$ *(бестиповое)*

Синтаксис

t	::=	*термы:*
	true	*константа «истина»*
	false	*константа «ложь»*
	if t then t else t	*условное выражение*
v	::=	*значения:*
	true	*константа «истина»*
	false	*константа «ложь»*

Вычисление $t \to t'$

$$\texttt{if true then } t_2 \texttt{ else } t_3 \to t_2 \qquad \text{(E-IfTrue)}$$

$$\texttt{if false then } t_2 \texttt{ else } t_3 \to t_3 \qquad \text{(E-IfFalse)}$$

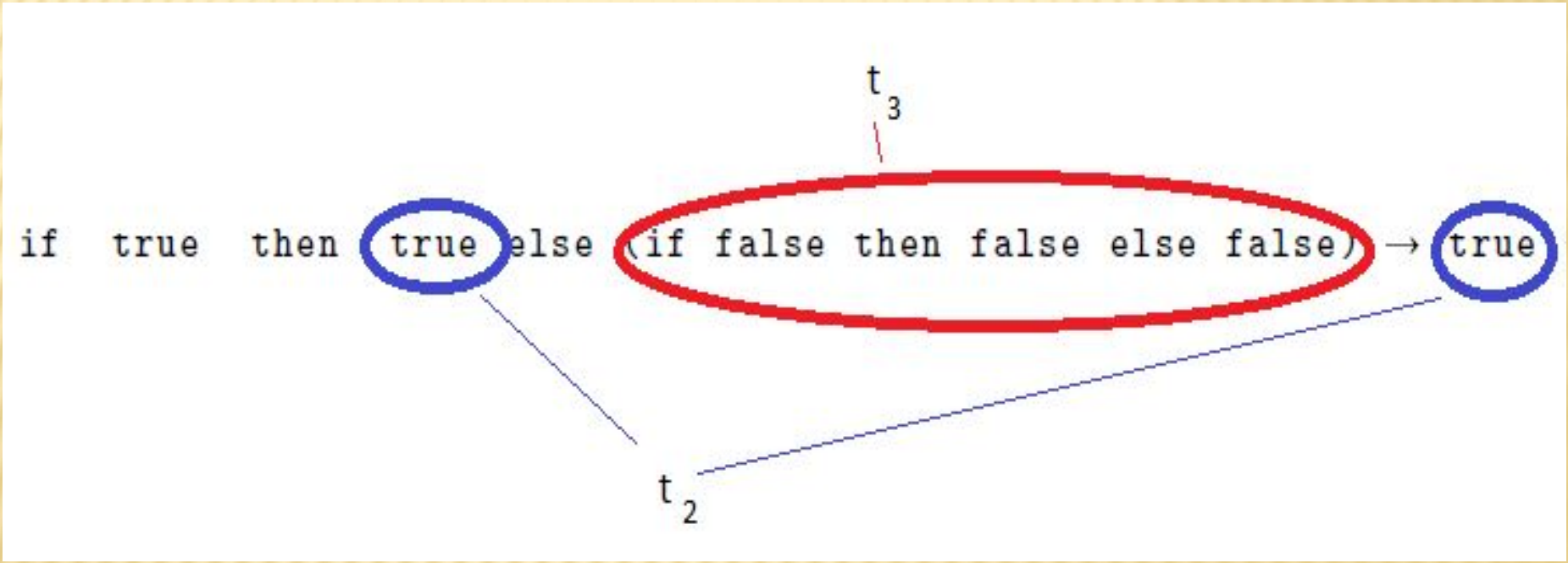
$$\frac{t_1 \to t_1'}{\texttt{if } t_1 \texttt{ then } t_2 \texttt{ else } t_3 \to \texttt{if } t_1' \texttt{ then } t_2 \texttt{ else } t_3} \qquad \text{(E-If)}$$

ВЫЧИСЛЕНИЯ (СЛУЧАЙ ТОЛЬКО БУЛЕВЫХ ВЫРАЖЕНИЙ)

1) $t \rightarrow t'$ *понимается как* t *за 1 шаг вычисляется как* t'
2) *Константы ни во что не вычисляются*
3) *E-IfTrue и E-IfFalse – рабочие правила*
4) *E-If – правило соответствия*
5) *Определение: Экземпляр правила вывода получается при замене каждой метапеременной в заключении правила и во всех его предпосылках*

ВЫЧИСЛЕНИЯ

Пример



ВЫЧИСЛЕНИЯ (СЛУЧАЙ ТОЛЬКО БУЛЕВЫХ ВЫРАЖЕНИЙ)

1) *Определение: Правило выполняется на отношении, если для каждого экземпляра правила его заключение является элементом отношения, либо одна из его предпосылок не является таковой*

ВЫЧИСЛЕНИЯ (СЛУЧАЙ ТОЛЬКО БУЛЕВЫХ ВЫРАЖЕНИЙ)

2. *Определение: Одношаговое отношение вычисления → есть наименьшее бинарное отношение на термах, на котором выполняются все три правила*

Если пара (t → t') является элементом отношения вычисления, то говорят, что утверждение о вычислении t → t' выводимо

ВЫЧИСЛЕНИЯ

Пример

$$
\begin{aligned}
\mathtt{s} &\stackrel{\text{def}}{=} \mathtt{if\ true\ then\ false\ else\ false} \\
\mathtt{t} &\stackrel{\text{def}}{=} \mathtt{if\ s\ then\ true\ else\ true} \\
\mathtt{u} &\stackrel{\text{def}}{=} \mathtt{if\ false\ then\ true\ else\ true}
\end{aligned}
$$

$$\mathtt{if\ t\ then\ false\ else\ false} \rightarrow \mathtt{if\ u\ then\ false\ else\ false}$$

$$
\dfrac{\dfrac{}{\mathtt{s} \rightarrow \mathtt{false}}\ \text{E-IfTrue}}{\mathtt{t} \rightarrow \mathtt{u}}\ \text{E-If}
\qquad
\dfrac{\mathtt{t} \rightarrow \mathtt{u}}{\mathtt{if\ t\ then\ false\ else\ false} \rightarrow \mathtt{if\ u\ then\ false\ else\ false}}\ \text{E-If}
$$

ВЫЧИСЛЕНИЯ (СВОЙСТВА ОТНОШЕНИЯ ВЫЧИСЛЕНИЯ)

Теорема [теорема о детерминированности одношагового вычисления]

$$Если\ t \to t'\ \ и\ t \to t'',\ то\ t' = t''.$$

ВЫЧИСЛЕНИЯ (СВОЙСТВА КОНЕЧНОГО СОСТОЯНИЯ)

Результат вычисления – конечное состояние

Определение 3 Терм t находится в нормальной форме если к нему не применимо никакое правило вычисления $(\nexists\, t', \text{т.ч. } t \rightarrow t')$

Теорема 2. Всякое значение находится в нормальной форме.

Теорема 3. Всякая нормальная форма есть значение

ВЫЧИСЛЕНИЯ

Определение. Отношение многошагового вычисления $\rightarrow^*$ это наименьшее отношение, т.ч.

(1) Если $t \rightarrow t'$, то $t \rightarrow^* t'$

(2) Для всех t выполняется $t \rightarrow^* t$

(3) Если $t \rightarrow^* t'$ и $t' \rightarrow^* t''$, то $t \rightarrow^* t''$

ВЫЧИСЛЕНИЯ

Теорема [теорема о единственности нормальных форм]

Если $t \to^* u$ и $t \to^* u'$ нормальные формы, то $u=u'$

ВЫЧИСЛЕНИЯ

Каждый терм можно вычислить и получить значение

Теорема [завершение вычислений]

Для каждого терма t *существует нормальная форма* t' *такая что* и $t \rightarrow^* t'$

ВЫЧИСЛЕНИЯ

Домашнее задание: упражнение 3.5.13, стр. 59

ВЫЧИСЛЕНИЯ (АРИФМЕТИЧЕСКИЕ ВЫРАЖЕНИЯ)

Новые синтаксисические формы

t	::= ...	*термы:*
	0	*константа ноль*
	succ t	*следующее число*
	pred t	*предыдущее число*
	iszero t	*проверка на ноль*
v	::= ...	*значения:*
	nv	*числовое значение*
nv	::= ...	*числовые значения:*
	0	*нулевое значение*
	succ nv	*значение-последователь*

Новые правила вычисления $t \to t'$

$$\frac{t_1 \to t_1'}{\texttt{succ } t_1 \to \texttt{succ } t_1'} \quad \text{(E-Succ)}$$

$$\texttt{pred } 0 \to 0 \quad \text{(E-PredZero)}$$

$$\texttt{pred (succ } nv_1\texttt{)} \to nv_1 \quad \text{(E-PredSucc)}$$

$$\frac{t_1 \to t_1'}{\texttt{pred } t_1 \to \texttt{pred } t_1'} \quad \text{(E-Pred)}$$

$$\texttt{iszero } 0 \to \texttt{true} \quad \text{(E-IszeroZero)}$$

$$\texttt{iszero (succ } nv_1\texttt{)} \to \texttt{false} \quad \text{(E-IszeroSucc)}$$

$$\frac{t_1 \to t_1'}{\texttt{iszero } t_1 \to \texttt{iszero } t_1'} \quad \text{(E-IsZero)}$$

ВЫЧИСЛЕНИЯ (АРИФМЕТИЧЕСКИЕ ВЫРАЖЕНИЯ)

- Определение. Терм если он находиться в нормальной форме, но не является значением называется тупиковым термом

Пример: *succ false*

ВЫЧИСЛЕНИЯ

Домашнее задание: упражнение 3.5.16-3.5.18, стр. 61,62

Кто **УСПЕШНО** выступит в понедельник – 5 дополнительных баллов на экзамене