Лекция С2
Машины Шёнфилда и частично вычислимые функции

Вадим Пузаренко

2 апреля 2020 г.

Мотивация

Основная идея доказательства того, что любая функция, вычилимая на машине Шёнфилда, является частично вычислимой, состоит в следующем: мы закодируем все вычисления на машинах Шёнфилда натуральными числами так, что все необходимые операции с полученными кодами (т. е. операции, имитирующие рабо- ту машины) можно будет производить с помощью частично вычислимых (даже примитивно рекурсивных) функций. Нам предстоит закодировать натуральными числами команды, программы, наборы входных данных, состояния счётчика команд, состояния регистров и, наконец, вычисления на машинах Шёнфилда.

Последовательности

Определение

Кодом последовательности $\langle x_0, x_1, \ldots, x_{k-1}\rangle$ натуральных чисел x_0, x_1, …, x_{k-1} назовём натуральное число $\mathrm{code}(\langle x_0, x_1, \ldots, x_{k-1}\rangle) = p_0^{x_0+1} \cdot p_1^{x_1+1} \cdot \ldots \cdot p_{k-1}^{x_{k-1}+1}$. Будем считать, что $\mathrm{code}(\langle\ \rangle) = 1$.

Последовательности

Определение

Кодом последовательности $\langle x_0, x_1, \ldots, x_{k-1}\rangle$ натуральных чисел x_0, x_1, …, x_{k-1} назовём натуральное число $\mathrm{code}(\langle x_0, x_1, \ldots, x_{k-1}\rangle) = p_0^{x_0+1} \cdot p_1^{x_1+1} \cdot \ldots \cdot p_{k-1}^{x_{k-1}+1}$. Будем считать, что $\mathrm{code}(\langle\ \rangle) = 1$.

Определение

Если x — код последовательности (скажем, $\langle x_0, x_1, \ldots, x_{k-1}\rangle$), то примитивно рекурсивные функции $\mathrm{lh}(x) = \mu i \leqslant x.(\mathrm{ex}(i, x) = 0)$ вычисляет длину данной последовательности, а $(x)_i = \mathrm{ex}(i, x) \dot{-} 1$ при $i < \mathrm{lh}(x)$ — i-ую координату x_i.

Последовательности

Определение

Кодом последовательности $\langle x_0, x_1, \ldots, x_{k-1}\rangle$ натуральных чисел x_0, x_1, …, x_{k-1} назовём натуральное число $\mathrm{code}(\langle x_0, x_1, \ldots, x_{k-1}\rangle) = p_0^{x_0+1} \cdot p_1^{x_1+1} \cdot \ldots \cdot p_{k-1}^{x_{k-1}+1}$. Будем считать, что $\mathrm{code}(\langle\ \rangle) = 1$.

Определение

Если x — код последовательности (скажем, $\langle x_0, x_1, \ldots, x_{k-1}\rangle$), то примитивно рекурсивные функции $\mathrm{lh}(x) = \mu i \leqslant x.(\mathrm{ex}(i, x) = 0)$ вычисляет длину данной последовательности, а $(x)_i = \mathrm{ex}(i, x) \dot{-} 1$ при $i < \mathrm{lh}(x)$ — i-ую координату x_i.

Лемма C6

Множество Seq всех кодов последовательностей является примитивно рекурсивным.

Команды

Доказательство.

$\mathrm{Seq}(x) \iff$
$[(x \neq 0) \wedge \forall i \leqslant x(\mathrm{ex}(i, x) \neq 0 \to \forall j \leqslant i(\mathrm{ex}(j, x) \neq 0))].$ $\square$

Команды

Доказательство.

$\mathrm{Seq}(x) \iff$
$[(x \neq 0) \wedge \forall i \leqslant x(\mathrm{ex}(i, x) \neq 0 \to \forall j \leqslant i(\mathrm{ex}(j, x) \neq 0))]$. □

Коды операторов (команд)

$\mathrm{cd}(\mathrm{INC}\,[i]) = \mathrm{code}(\langle 0, i \rangle)$,
$\mathrm{cd}(\mathrm{DEC}\,[i], j) = \mathrm{code}(\langle 1, i, j \rangle)$.

Команды

Доказательство.

$\mathrm{Seq}(x) \iff$
$[(x \neq 0) \wedge \forall i \leqslant x(\mathrm{ex}(i,x) \neq 0 \to \forall j \leqslant i(\mathrm{ex}(j,x) \neq 0))].$ $\square$

Коды операторов (команд)

$\mathrm{cd}(\mathrm{INC}\,[i]) = \mathrm{code}(\langle 0, i \rangle),$
$\mathrm{cd}(\mathrm{DEC}\,[i], j) = \mathrm{code}(\langle 1, i, j \rangle).$

Код программы

Пусть программа P имеет вид:
$0 : P_0$
$1 : P_1$
$\ldots$
$k-1 : P_{k-1}$
Тогда положим $\mathrm{code}(P) = \mathrm{code}(\langle \mathrm{cd}(P_0), \mathrm{cd}(P_1), \ldots, \mathrm{cd}(P_{k-1}) \rangle).$

Команды

Лемма C7

Множество $\mathrm{Com}(x)$ кодов команд примитивно рекурсивно.

Команды

Лемма С7

Множество $\mathrm{Com}(x)$ кодов команд примитивно рекурсивно.

Доказательство.

$\mathrm{Com}(x) \iff [\mathrm{Seq}(x) \wedge ((((x)_0 = 0) \wedge (\mathrm{lh}(x) = 2)) \vee (((x)_0 = 1) \wedge (\mathrm{lh}(x) = 3)))]$. $\square$

Команды

Лемма С7

Множество $\mathrm{Com}(x)$ кодов команд примитивно рекурсивно.

Доказательство.

$\mathrm{Com}(x) \iff [\mathrm{Seq}(x) \wedge ((((x)_0 = 0) \wedge (\mathrm{lh}(x) = 2)) \vee (((x)_0 = 1) \wedge (\mathrm{lh}(x) = 3)))]$. □

Лемма С8

Множество $\mathrm{Prog}(x)$ кодов программ является примитивно рекурсивным.

Команды

Лемма C7

Множество $\mathrm{Com}(x)$ кодов команд примитивно рекурсивно.

Доказательство.

$\mathrm{Com}(x) \iff [\mathrm{Seq}(x) \wedge ((((x)_0 = 0) \wedge (\mathrm{lh}(x) = 2)) \vee (((x)_0 = 1) \wedge (\mathrm{lh}(x) = 3)))]$. $\square$

Лемма C8

Множество $\mathrm{Prog}(x)$ кодов программ является примитивно рекурсивным.

Доказательство.

$\mathrm{Prog}(x) \iff \mathrm{Seq}(x) \wedge \forall i < \mathrm{lh}(x) \mathrm{Com}((x)_i)$. $\square$

Совместная рекурсия

Предложение С10

Пусть функции $f_0(x_1x_2,\ldots,x_n,y)$, $f_1(x_1x_2,\ldots,x_n,y)$ определены по схеме совместной рекурсии:

$$\left[\begin{array}{l} f_0(x_1x_2,\ldots,x_n,0)=g_0(x_1,x_2,\ldots,x_n),\\ f_1(x_1x_2,\ldots,x_n,0)=g_1(x_1,x_2,\ldots,x_n),\\ f_0(x_1x_2,\ldots,x_n,y+1)=h_0(x_1x_2,\ldots,x_n,y,f_0(x_1x_2,\ldots,x_n,y),\\ f_1(x_1x_2,\ldots,x_n,y)),\\ f_1(x_1x_2,\ldots,x_n,y+1)=h_1(x_1x_2,\ldots,x_n,y,f_0(x_1x_2,\ldots,x_n,y),\\ f_1(x_1x_2,\ldots,x_n,y)).\end{array}\right.$$

Совместная рекурсия

Предложение C10

Пусть функции $f_0(x_1 x_2, \ldots, x_n, y)$, $f_1(x_1 x_2, \ldots, x_n, y)$ определены по схеме совместной рекурсии:

$$\left[\begin{array}{l} f_0(x_1 x_2, \ldots, x_n, 0) = g_0(x_1, x_2, \ldots, x_n), \\ f_1(x_1 x_2, \ldots, x_n, 0) = g_1(x_1, x_2, \ldots, x_n), \\ f_0(x_1 x_2, \ldots, x_n, y+1) = h_0(x_1 x_2, \ldots, x_n, y, f_0(x_1 x_2, \ldots, x_n, y), \\ f_1(x_1 x_2, \ldots, x_n, y)), \\ f_1(x_1 x_2, \ldots, x_n, y+1) = h_1(x_1 x_2, \ldots, x_n, y, f_0(x_1 x_2, \ldots, x_n, y), \\ f_1(x_1 x_2, \ldots, x_n, y)). \end{array}\right.$$

Доказательство.

Положим

$F(x_1, x_2, \ldots, x_n, y) = \mathrm{code}(\langle f_0(x_1 x_2, \ldots, x_n, y), f_1(x_1 x_2, \ldots, x_n, y)\rangle)$.

Тогда

Совместная рекурсия

Доказательство (продолжение)

$$\left[\begin{array}{l} F(x_1,x_2,\ldots,x_n,0)=\mathrm{code}(\langle g_0(x_1,x_2,\ldots,x_n),g_1(x_1,x_2,\ldots,x_n)\rangle),\\ F(x_1,x_2,\ldots,x_n,y+1)=\\ =\mathrm{code}(\langle h_0(x_1x_2,\ldots,x_n,y,f_0(x_1x_2,\ldots,x_n,y),f_1(x_1x_2,\ldots,x_n,y)),\\ h_1(x_1x_2,\ldots,x_n,y,f_0(x_1x_2,\ldots,x_n,y),f_1(x_1x_2,\ldots,x_n,y)))\rangle)\end{array}\right.$$

Наконец, $f_0(x_1x_2,\ldots,x_n,y)=(F(x_1,x_2,\ldots,x_n,y))_0$,
$f_1(x_1x_2,\ldots,x_n,y)=(F(x_1,x_2,\ldots,x_n,y))_1$. $\square$

Кодировка

Теперь мы введём две важные функции, кодирующие полностью всю информацию о ходе вычисления на машине Шёнфилда. Чтобы целиком охватить поток данных, изменяющихся в ходе такого вычисления, необходимо знать на каждом шаге содержимое счётчика команд и содержимые всех регистров, которые влияют на ход вычислений.
Оценим, насколько большим может быть номер регистра, существенно влияющего на ход работы заданной машины Шёнфилда. Для этого заметим, что $(x)_i < x$ для всех $x > 0$. Действительно,
$(x)_i = \mathrm{ex}(x, i) \dot{-} 1 \leqslant \mathrm{ex}(x, i) < 2^{\mathrm{ex}(x,i)} \leqslant p_i^{\mathrm{ex}(x,i)} \leqslant x$.
Отсюда следует, что если e — код программы, m — номер упомянутого регистра в этой программе, то $e > m$. Кроме того, ход вычислений по программе с кодом e зависит от входных данных, которые могут быть записаны в регистрах с 1-го по k-ый, где k, вообще говоря, — произвольное натуральное число. Тем самым, вычисления программы с кодом e, вычисляющей k-местную функцию, не используют содержимые регистров с номерами $\geqslant e + k$.

Кодировка

1. $\mathrm{ct}(e, x, n)$ выдаёт содержимое счётчика команд после n шагов вычисления с программой с кодом e и содержимых x_1, x_2, ..., x_k регистров с 1-го по k-ый, если $x = \mathrm{code}(\langle x_1, x_2, \ldots, x_k \rangle)$.
2. $\mathrm{rg}(e, x, n)$ выдаёт код последовательности $\langle r_0, r_1, \ldots, r_{e+k-1} \rangle$ содержимых регистров после n шагов вычисления с программой с кодом e и содержимых x_1, x_2, ..., x_k регистров с 1-го по k-ый, если $x = \mathrm{code}(\langle x_1, x_2, \ldots, x_k \rangle)$.

Кодировка

1. $\mathrm{ct}(e, x, n)$ выдаёт содержимое счётчика команд после n шагов вычисления с программой с кодом e и содержимых x_1, x_2, ..., x_k регистров с 1-го по k-ый, если $x = \mathrm{code}(\langle x_1, x_2, \ldots, x_k \rangle)$.
2. $\mathrm{rg}(e, x, n)$ выдаёт код последовательности $\langle r_0, r_1, \ldots, r_{e+k-1} \rangle$ содержимых регистров после n шагов вычисления с программой с кодом e и содержимых x_1, x_2, ..., x_k регистров с 1-го по k-ый, если $x = \mathrm{code}(\langle x_1, x_2, \ldots, x_k \rangle)$.

Определение

$$\mathrm{ct}(e, x, n) = \begin{cases} y, & \text{если выполняется следующее:} \\ & (\imath)\ e \text{ — код программы } P, \\ & (\imath\imath)\ x = \mathrm{code}(\langle x_1, x_2, \ldots, x_k \rangle), \\ & (\imath\imath\imath)\ y \text{ — содержимое счётчика команд после} \\ & n \text{ шагов выполнения программы } P, \text{ начатой с} \\ & \text{содержимыми регистров } 0, x_1, x_2, \ldots, x_k, 0, \ldots, 0; \\ 0 & \text{в противном случае.} \end{cases}$$

Кодировка

Определение

$$\mathrm{rg}(e, x, n) = \begin{cases} \mathrm{code}(\langle r_0, \ldots, r_{e+k-1} \rangle), & \text{если выполняется следующее:} \\ & (\imath)\ e \text{ — код программы } P, \\ & (\imath\imath)\ x = \mathrm{code}(\langle x_1, x_2, \ldots, x_k \rangle), \\ & (\imath\imath\imath)\ r_i \text{ — содержимое } i\text{-го} \\ & \text{регистра после } n \text{ шагов} \\ & \text{выполнения} \\ & \text{программы } P, \text{ начатой с} \\ & \text{содержимыми регистров} \\ & 0, x_1, x_2, \ldots, x_k, 0, \ldots, 0; \\ 0 & \text{в противном случае.} \end{cases}$$

Кодировка

Определение

$$\mathrm{rg}(e,x,n)=\begin{cases}\mathrm{code}(\langle r_0,\ldots,r_{e+k-1}\rangle), & \text{если выполняется следующее:}\\ & (\imath)\ e \text{ — код программы } P,\\ & (\imath\imath)\ x=\mathrm{code}(\langle x_1,x_2,\ldots,x_k\rangle),\\ & (\imath\imath\imath)\ r_i \text{ — содержимое } i\text{-го}\\ & \text{регистра после } n \text{ шагов}\\ & \text{выполнения}\\ & \text{программы } P,\ \text{начатой с}\\ & \text{содержимыми регистров}\\ & 0,x_1,x_2,\ldots,x_k,0,\ldots,0;\\ 0 & \text{в противном случае.}\end{cases}$$

Лемма C9

Функции $\mathrm{ct}(e,x,n)$ и $\mathrm{rg}(e,x,n)$ примитивно рекурсивны.

Кодировка

Доказательство.

Воспользуемся схемой совместной рекурсии.

$\mathrm{ct}(e,x,0)=0$.

Для задания $\mathrm{rg}(e,x,0)$ определим вспомогательную прф:

$$\alpha(i,x)=\begin{cases}\mathrm{ex}(i\dot{-}1,x), & \text{если } 1\leqslant i\leqslant \mathrm{lh}(x);\\ 0 & \text{в противном случае.}\end{cases}$$

Тогда имеем

$$\mathrm{rg}(e,x,0)=\begin{cases}\prod\limits_{i=0}^{e+(\mathrm{lh}(x)\dot{-}1)} p_i^{\alpha(i,x)}, & \text{если } \mathrm{Prog}(e)\wedge \mathrm{Seq}(x);\\ 0, & \text{если } \neg\mathrm{Prog}(e)\vee\neg\mathrm{Seq}(x).\end{cases}$$

Кодировка

Доказательство.

Воспользуемся схемой совместной рекурсии.

$\mathrm{ct}(e, x, 0) = 0$.

Для задания $\mathrm{rg}(e, x, 0)$ определим вспомогательную прф:

$$\alpha(i, x) = \begin{cases} \mathrm{ex}(i \dot{-} 1, x), & \text{если } 1 \leqslant i \leqslant \mathrm{lh}(x); \\ 0 & \text{в противном случае.} \end{cases}$$

Тогда имеем

$$\mathrm{rg}(e, x, 0) = \begin{cases} \prod\limits_{i=0}^{e+(\mathrm{lh}(x) \dot{-} 1)} p_i^{\alpha(i,x)}, & \text{если } \mathrm{Prog}(e) \wedge \mathrm{Seq}(x); \\ 0, & \text{если } \neg\mathrm{Prog}(e) \vee \neg\mathrm{Seq}(x). \end{cases}$$

Предположим, что значения $\mathrm{ct}(e, x, n)$ и $\mathrm{rg}(e, x, n)$ уже заданы.

Кодировка

Доказательство (продолжение).

Сначала дадим неформальное описание $\mathrm{ct}(e, x, n+1)$ (здесь $z = \mathrm{ct}(e, x, n)$):

$$
\mathrm{ct}(e, x, n+1) = \begin{cases}
\mathrm{s}(z), & \text{если } \mathrm{Prog}(e), \mathrm{Seq}(x), \text{ и команда с} \\
 & \text{номером } z \text{ имеет вид } \mathrm{INC}\,[i]; \\
j, & \text{если } \mathrm{Prog}(e), \mathrm{Seq}(x), \text{ команда с} \\
 & \text{номером } z \text{ имеет вид } \mathrm{DEC}\,[i], j, \\
 & \text{ и содержимое } [i]\text{-го регистра } > 0 \\
 & \text{ в момент } n; \\
\mathrm{s}(z), & \text{если } \mathrm{Prog}(e), \mathrm{Seq}(x), \text{ команда с} \\
 & \text{номером } z \text{ имеет вид } \mathrm{DEC}\,[i], j, \\
 & \text{ и содержимое } [i]\text{-го регистра } = 0 \\
 & \text{ в момент } n; \\
z, & \text{если } \mathrm{Prog}(e), \mathrm{Seq}(x), \text{ и команда с} \\
 & \text{номером } z \text{ отсутствует}; \\
0 & \text{в остальных случаях.}
\end{cases}
$$

Кодировка

Доказательство (продолжение).

Формально (снова $z = \mathrm{ct}(e,x,n)$; к тому же, $v = \mathrm{rg}(e,x,n)$):
$\mathrm{ct}(e,x,n+1) =$

$$\begin{cases} \mathrm{s}(z), & \text{если } \mathrm{Prog}(e) \wedge \mathrm{Seq}(x) \wedge (z < \mathrm{lh}(e)) \wedge \Big((((e)_z)_0 = 0) \vee \\ & \vee ((((e)_z)_0 = 1) \wedge ((v)_{((e)_z)_1} = 0)) \Big); \\ ((e)_z)_2, & \text{если } \mathrm{Prog}(e) \wedge \mathrm{Seq}(x) \wedge (z < \mathrm{lh}(e)) \wedge \\ & \wedge ((((e)_z)_0 = 1) \wedge ((v)_{((e)_z)_1} > 0)); \\ z, & \text{если } \mathrm{Prog}(e) \wedge \mathrm{Seq}(x) \wedge (z \geqslant \mathrm{lh}(e)); \\ 0 & \text{в остальных случаях.} \end{cases}$$

Кодировка

Доказательство (продолжение).

Формально (снова $z = \mathrm{ct}(e, x, n)$; к тому же, $v = \mathrm{rg}(e, x, n)$):
$\mathrm{ct}(e, x, n+1) =$

$$\begin{cases} \mathrm{s}(z), & \text{если } \mathrm{Prog}(e) \wedge \mathrm{Seq}(x) \wedge (z < \mathrm{lh}(e)) \wedge \Big((((e)_z)_0 = 0) \vee \\ & \vee ((((e)_z)_0 = 1) \wedge ((v)_{((e)_z)_1} = 0)) \Big); \\ ((e)_z)_2, & \text{если } \mathrm{Prog}(e) \wedge \mathrm{Seq}(x) \wedge (z < \mathrm{lh}(e)) \wedge \\ & \wedge ((((e)_z)_0 = 1) \wedge ((v)_{((e)_z)_1} > 0)); \\ z, & \text{если } \mathrm{Prog}(e) \wedge \mathrm{Seq}(x) \wedge (z \geqslant \mathrm{lh}(e)); \\ 0 & \text{в остальных случаях.} \end{cases}$$

$\mathrm{Prog}(e)$ означает, что e — код программы (скажем, P);
$\mathrm{Seq}(x)$ означает, что x — код последовательности (скажем, $\langle x_1, x_2, \ldots, x_k \rangle$);
$\mathrm{ct}(e, x, n) < \mathrm{lh}(e)$ означает, что после шага n вычисления будет исполняться существующая в программе P команда с номером $\mathrm{ct}(e, x, n)$;

Кодировка

Доказательство (продолжение).

$(((e)_z)_0 = 0)$ означает, что выполняется команда вида $\mathrm{INC}\,[i]$;
$(((e)_z)_0 = 1) \wedge ((v)_{((e)_z)_1} = 0)$ означает, что выполняется команда вида $\mathrm{DEC}\,[i], j$, причём содержимое $[i]$-го регистра равно нулю (в этом случае счётчик команд увеличивается на единицу);
$(((e)_z)_0 = 1) \wedge ((v)_{((e)_z)_1} > 0)$ означает, что выполняется команда вида $\mathrm{DEC}\,[i], j$, причём содержимое $[i]$-го регистра больше нуля (в этом случае счётчику команд присваивается значение j, т.е. $((e)_z)_2)$;
$\mathrm{ct}(e, x, n) \geqslant \mathrm{lh}(e)$ означает, что команда с номером $\mathrm{ct}(e, x, n)$ отсутствует в программе P.

Кодировка

Доказательство (продолжение).

Дадим теперь неформальное описание $\mathrm{rg}(e, x, n+1)$ (как и прежде, пусть $z = \mathrm{ct}(e, x, n)$ и $\mathrm{rg}(e, x, n) = \mathrm{code}(\langle r_0, r_1, \ldots, r_{e+k-1}\rangle)$):

$$\mathrm{rg}(e, x, n+1) = \begin{cases} \mathrm{code}(r_0, \ldots, r_i + 1, \ldots, r_{e+k-1}), & \text{если } \mathrm{Prog}(e), \mathrm{Seq}(x), \text{ и команда с номером } z \text{ имеет вид } \mathrm{INC}\,[i]; \\ \mathrm{code}(r_0, \ldots, r_i \dot{-} 1, \ldots, r_{e+k-1}), & \text{если } \mathrm{Prog}(e), \mathrm{Seq}(x), \text{ команда с номером } z \text{ имеет вид } \mathrm{DEC}\,[i], j, \text{ и } r_i > 0; \\ \mathrm{code}(r_0, \ldots, r_i, \ldots, r_{e+k-1}), & \text{если } \mathrm{Prog}(e), \mathrm{Seq}(x), \text{ команда с номером } z \text{ имеет вид } \mathrm{DEC}\,[i], j, \text{ и } r_i = 0; \\ \mathrm{code}(r_0, \ldots, , r_{e+k-1}), & \text{если } \mathrm{Prog}(e), \mathrm{Seq}(x), \text{ и команда с номером } z \text{ отсутствует}; \\ 0 & \text{в остальных случаях.} \end{cases}$$

Кодировка

Доказательство (продолжение).

Отметим, что второй и третий случаи можно рассмотреть одновременно, поскольку $0 = r_i = r_i \dot{-} 1$.

Кодировка

Доказательство (продолжение).

Отметим, что второй и третий случаи можно рассмотреть одновременно, поскольку $0 = r_i = r_i \dot{-} 1$.

Введём вспомогательную прф $\beta(i,x,y) = \left[\frac{x}{p_i^{\mathrm{ex}(i,x)}}\right] \cdot p_i^{y+1}$. Данная функция удовлетворяет следующему условию: если $x = \mathrm{code}(\langle x_0,\ldots,x_{i-1},x_i,x_{i+1},\ldots,x_k\rangle)$ и $i \leqslant k$, то $\beta(i,x,y) = \mathrm{code}(\langle x_0,\ldots,x_{i-1},y,x_{i+1},\ldots,x_k\rangle)$.

Кодировка

Доказательство (продолжение).

Отметим, что второй и третий случаи можно рассмотреть одновременно, поскольку $0 = r_i = r_i \dot{-} 1$.

Введём вспомогательную прф $\beta(i,x,y) = \left[\dfrac{x}{p_i^{\mathrm{ex}(i,x)}}\right] \cdot p_i^{y+1}$. Данная функция удовлетворяет следующему условию: если $x = \mathrm{code}(\langle x_0, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_k\rangle)$ и $i \leqslant k$, то $\beta(i,x,y) = \mathrm{code}(\langle x_0, \ldots, x_{i-1}, y, x_{i+1}, \ldots, x_k\rangle)$.

Формально (снова $z = \mathrm{ct}(e,x,n)$, $v = \mathrm{rg}(e,x,n)$):

$$\mathrm{rg}(e,x,n+1) = \begin{cases} \beta(((e)_z)_1, v, \mathrm{s}((v)_{((e)_z)_1})), & \text{если } \mathrm{Prog}(e) \wedge \mathrm{Seq}(x) \wedge (z < \mathrm{lh}(e)) \wedge \\ & \wedge(((e)_z)_0 = 0); \\ \beta(((e)_z)_1, v, (v)_{((e)_z)_1} \dot{-} 1), & \text{если } \mathrm{Prog}(e) \wedge \mathrm{Seq}(x) \wedge (z < \mathrm{lh}(e)) \wedge \\ & \wedge(((e)_z)_0 = 1); \\ v, & \text{если } \mathrm{Prog}(e) \wedge \mathrm{Seq}(x) \wedge (z \geqslant \mathrm{lh}(e)); \\ 0 & \text{в остальных случаях.} \end{cases}$$

□

Предикат остановки

Определение

Определим предикат $\mathrm{Stop}(e, x, n)$ как отношение, удовлетворяющее следующим условиям в точности:

$(\imath)$ e — код некоторой программы (скажем, P);

$(\imath\imath)$ $x = \mathrm{code}(\langle x_1, x_2, \ldots, x_k \rangle)$

$(\imath\imath\imath)$ программа P, начав работу с содержимым регистров 0, x_1, x_2, ..., x_k, 0, 0, ..., 0, останавливается к шагу n.

Предикат остановки

Определение

Определим предикат $\mathrm{Stop}(e, x, n)$ как отношение, удовлетворяющее следующим условиям в точности:

(i) e — код некоторой программы (скажем, P);

(ii) $x = \mathrm{code}(\langle x_1, x_2, \ldots, x_k \rangle)$

(iii) программа P, начав работу с содержимым регистров 0, x_1, x_2, $\ldots$, x_k, 0, 0, $\ldots$, 0, останавливается к шагу n.

Лемма C10

Отношение $\mathrm{Stop}(e, x, n)$ примитивно рекурсивно.

Предикат остановки

Определение

Определим предикат $\mathrm{Stop}(e, x, n)$ как отношение, удовлетворяющее следующим условиям в точности:

(i) e — код некоторой программы (скажем, P);

(ii) $x = \mathrm{code}(\langle x_1, x_2, \ldots, x_k \rangle)$

(iii) программа P, начав работу с содержимым регистров 0, x_1, x_2, ..., x_k, 0, 0, ..., 0, останавливается к шагу n.

Лемма C10

Отношение $\mathrm{Stop}(e, x, n)$ примитивно рекурсивно.

Доказательство.

Действительно,

$\mathrm{Stop}(e, x, n) \iff \mathrm{Prog}(e) \wedge \mathrm{Seq}(x) \wedge (\mathrm{ct}(e, x, n) \geqslant \mathrm{lh}(e))$. □

Коды вычислений

Пусть натуральные числа e, x и n таковы, что $\mathrm{Stop}(e, x, n)$.

Определение

Кодом вычисления на машине Шёнфилда с программой P, имеющей код e, и начальной конфигурацией содержимого регистров 0, $(x)_0$, $(x)_1$, …, $(x)_{\mathrm{lh}(x)-1}$, 0, …, 0, будем называть $\mathrm{code}(\langle \mathrm{rg}(e, x, 0), \mathrm{rg}(e, x, 1), \ldots, \mathrm{rg}(e, x, n)\rangle)$.

Коды вычислений

Пусть натуральные числа e, x и n таковы, что $\mathrm{Stop}(e, x, n)$.

Определение

Кодом вычисления на машине Шёнфилда с программой P, имеющей код e, и начальной конфигурацией содержимого регистров 0, $(x)_0$, $(x)_1$, …, $(x)_{\mathrm{lh}(x)-1}$, 0, …, 0, будем называть $\mathrm{code}(\langle \mathrm{rg}(e, x, 0), \mathrm{rg}(e, x, 1), \ldots, \mathrm{rg}(e, x, n)\rangle)$.

Определение

Если y — код вычисления, то результат вычисления содержится в 0-м регистре и, следовательно, вычисляется с помощью прф $U(y) = ((y)_{\mathrm{lh}(y) \dot{-} 1})_0$.

Коды вычислений

Пусть натуральные числа e, x и n таковы, что $\mathrm{Stop}(e, x, n)$.

Определение

Кодом вычисления на машине Шёнфилда с программой P, имеющей код e, и начальной конфигурацией содержимого регистров 0, $(x)_0$, $(x)_1$, …, $(x)_{\mathrm{lh}(x)-1}$, 0, …, 0, будем называть $\mathrm{code}(\langle \mathrm{rg}(e, x, 0), \mathrm{rg}(e, x, 1), \ldots, \mathrm{rg}(e, x, n)\rangle)$.

Определение

Если y — код вычисления, то результат вычисления содержится в 0-м регистре и, следовательно, вычисляется с помощью прф $U(y) = ((y)_{\mathrm{lh}(y) \dot{-} 1})_0$.

Если $e, x \in \omega$ не удовлетворяют $\mathrm{Stop}(e, x, n)$ ни для какого $n \in \omega$, то считаем код вычисления не определённым.

Предикат Клини

Определение

Пусть $k \geqslant 1$; определим $k + 2$-арный **предикат Клини** $T_k(e, x_1, x_2, \ldots, x_k, y)$ как отношение, удовлетворяющее в точности следующим условиям:

1. e — код некоторой программы (скажем, P);
2. y — код вычисления программы P с начальной конфигурацией содержимого регистров 0, x_1, x_2, ..., x_k, 0, ..., 0.

Предикат Клини

Определение

Пусть $k \geqslant 1$; определим $k+2$-арный **предикат Клини** $T_k(e, x_1, x_2, \ldots, x_k, y)$ как отношение, удовлетворяющее в точности следующим условиям:

1. e — код некоторой программы (скажем, P);
2. y — код вычисления программы P с начальной конфигурацией содержимого регистров 0, x_1, x_2, ..., x_k, 0, ..., 0.

Лемма C11

Для любого $k \geqslant 1$ предикат $T_k(e, x_1, \ldots, x_k, y)$ примитивно рекурсивен.

Предикат Клини

Определение

Пусть $k \geqslant 1$; определим $k+2$-арный **предикат Клини** $T_k(e, x_1, x_2, \ldots, x_k, y)$ как отношение, удовлетворяющее в точности следующим условиям:

1. e — код некоторой программы (скажем, P);
2. y — код вычисления программы P с начальной конфигурацией содержимого регистров 0, x_1, x_2, …, x_k, 0, …, 0.

Лемма C11

Для любого $k \geqslant 1$ предикат $T_k(e, x_1, \ldots, x_k, y)$ примитивно рекурсивен.

Доказательство.

Действительно,

$T_k(e, x_1, \ldots, x_k, y) \Longleftrightarrow \mathrm{Seq}(y) \wedge \mathrm{Stop}(e, \mathrm{code}(\langle x_1, x_2, \ldots, x_k \rangle), \mathrm{lh}(y) \dot{-} 1) \wedge \forall i < \mathrm{lh}(y)[(y)_i = \mathrm{rg}(e, \mathrm{code}(\langle x_1, x_2, \ldots, x_k \rangle), i)]$. □

МШ $\mapsto$ ЧВФ

Теорема C3

Любая частичная функция, вычислимая на машине Шёнфилда, частично вычислима.

МШ $\mapsto$ ЧВФ

Теорема C3

Любая частичная функция, вычислимая на машине Шёнфилда, частично вычислима.

Доказательство.

Пусть $f(x_1, x_2, \ldots, x_k)$ — частичная функция, вычислимая на машине Шёнфилда и пусть P — программа этой машины. Пусть также e — код программы P. Возьмём произвольный набор n_1, n_2, ..., n_k чисел. Разберём два случая.

$f(n_1, n_2, \ldots, n_k)$ **определено.** Тогда программа P с начальной конфигурацией содержимого регистров 0, n_1, n_2, ..., n_k, 0, ..., 0 останавливается. Следовательно, определён код вычисления y, причём $U(y) = f(n_1, n_2, \ldots, n_k)$. Выберем наименьший код вычисления y_0; тогда
$f(n_1, n_2, \ldots, n_k) = U(y_0) = U(\mu y.T_k(e, n_1, n_2, \ldots, n_k, y))$.

МШ $\mapsto$ ЧВФ

Доказательство (продолжение).

$f(n_1, n_2, \ldots, n_k)$ **не определено.** Тогда программа P с начальной конфигурацией содержимого регистров 0, n_1, n_2, ..., n_k, 0, ..., 0 никогда не останавливается. Следовательно, не определён код вычисления y, а вместе с этим $T_k(e, n_1, n_2, \ldots, n_k, y)$ не выполняется ни для какого y. Тем самым, $\mu y.T_k(e, n_1, n_2, \ldots, n_k, y) \uparrow$ и $U(\mu y.T_k(e, n_1, n_2, \ldots, n_k, y)) \uparrow$.

Таким образом, $f(x_1, x_2, \ldots, x_k) = U(\mu y.T_k(e, x_1, x_2, \ldots, x_k, y))$ и, в частности, $f(x_1, x_2, \ldots, x_k)$ — чвф. $\square$

МШ $\mapsto$ ЧВФ

Доказательство (продолжение).

$f(n_1, n_2, \ldots, n_k)$ **не определено.** Тогда программа P с начальной конфигурацией содержимого регистров 0, n_1, n_2, …, n_k, 0, …, 0 никогда не останавливается. Следовательно, не определён код вычисления y, а вместе с этим $T_k(e, n_1, n_2, \ldots, n_k, y)$ не выполняется ни для какого y. Тем самым, $\mu y.T_k(e, n_1, n_2, \ldots, n_k, y)\uparrow$ и $U(\mu y.T_k(e, n_1, n_2, \ldots, n_k, y))\uparrow$.

Таким образом, $f(x_1, x_2, \ldots, x_k) = U(\mu y.T_k(e, x_1, x_2, \ldots, x_k, y))$ и, в частности, $f(x_1, x_2, \ldots, x_k)$ — чвф. $\square$

Теорема С4(Клини о нормальной форме)

Существует примитивно рекурсивная функция U такая, что для любого $k \geqslant 1$ найдётся примитивно рекурсивное отношение $T_k(e, x_1, x_2, \ldots, x_k, y)$, для которого выполняется следующее: для любой k-местной частично вычислимой функции $\varphi(x_1, x_2, \ldots, x_k)$ найдётся e_0, для которого имеет место $\varphi(x_1, x_2, \ldots, x_k) = U(\mu y.T_k(e_0, x_1, x_2, \ldots, x_k, y))$.

Универсальные функции

Следствие C1

Любая частично вычислимая функция может быть получена с помощью применения операторов S, R и M, причём оператор M используется не более одного раза.

Универсальные функции

Следствие C1

Любая частично вычислимая функция может быть получена с помощью применения операторов S, R и M, причём оператор M используется не более одного раза.

Определение

Пусть $k \geqslant 1$ и пусть $\mathcal{S}$ — семейство k-местных частичных функций. Функция $F(x_0, x_1, \ldots, x_k)$ называется **универсальной** для семейства $\mathcal{S}$, если $\mathcal{S} = \{\lambda x_1 x_2 \ldots x_k . F(e, x_1, x_2, \ldots, x_k) | e \in \omega\}$. В случае, когда $\mathcal{S}$ — семейство всех k-местных частично вычислимых функций, то $F(x_0, x_1, \ldots, x_k)$ называется просто **универсальной**.

Универсальные функции

Следствие C1

Любая частично вычислимая функция может быть получена с помощью применения операторов S, R и M, причём оператор M используется не более одного раза.

Определение

Пусть $k \geqslant 1$ и пусть $\mathcal{S}$ — семейство k-местных частичных функций. Функция $F(x_0, x_1, \dots, x_k)$ называется **универсальной** для семейства $\mathcal{S}$, если $\mathcal{S} = \{\lambda x_1 x_2 \dots x_k . F(e, x_1, x_2, \dots, x_k) | e \in \omega\}$. В случае, когда $\mathcal{S}$ — семейство всех k-местных частично вычислимых функций, то $F(x_0, x_1, \dots, x_k)$ называется просто **универсальной**.

Предложение C11

Каково бы ни было $k \geqslant 1$, не существует универсальной частично вычислимой (примитивно рекурсивной) функции семейства всех k-местных вычислимых (примитивно рекурсивных) функций.

Универсальные функции

Доказательство.

Допустим, что $F(x_0, x_1, x_2, \ldots, x_k)$ — соответствующая универсальная функция. Тогда $F(x_1, x_1, x_2, \ldots, x_k)$ является k-местной вычислимой (примитивно рекурсивной) функцией, а вместе с ней и $\mathrm{s}(F(x_1, x_1, x_2, \ldots, x_k))$ (поскольку функция $F(x_1, x_1, x_2, \ldots, x_k)$ всюду определена). Следовательно, существует e_0 такое, что
$F(e_0, x_1, x_2, \ldots, x_k) = \mathrm{s}(F(x_1, x_1, x_2, \ldots, x_k))$ для всех натуральных чисел x_1, x_2, …, x_k. В частности,
$F(e_0, e_0, \ldots, e_0) = \mathrm{s}(F(e_0, e_0, \ldots, e_0))$, противоречие. $\square$

Универсальные функции

Доказательство.

Допустим, что $F(x_0, x_1, x_2, \ldots, x_k)$ — соответствующая универсальная функция. Тогда $F(x_1, x_1, x_2, \ldots, x_k)$ является k-местной вычислимой (примитивно рекурсивной) функцией, а вместе с ней и $\mathrm{s}(F(x_1, x_1, x_2, \ldots, x_k))$ (поскольку функция $F(x_1, x_1, x_2, \ldots, x_k)$ всюду определена). Следовательно, существует e_0 такое, что $F(e_0, x_1, x_2, \ldots, x_k) = \mathrm{s}(F(x_1, x_1, x_2, \ldots, x_k))$ для всех натуральных чисел x_1, x_2, ..., x_k. В частности, $F(e_0, e_0, \ldots, e_0) = \mathrm{s}(F(e_0, e_0, \ldots, e_0))$, противоречие. $\square$

Предложение C12

Каково бы ни было $k \geqslant 1$, не существует универсальной частично вычислимой (примитивно рекурсивной) функции семейства всех k-местных вычислимых (примитивно рекурсивных) функций, принимающих значения $\subseteq \{0; 1\}$.

Универсальные функции

Доказательство.

Допустим, что $F(x_0, x_1, x_2, \ldots, x_k)$ — соответствующая универсальная функция. Тогда $F(x_1, x_1, x_2, \ldots, x_k)$ является k-местной вычислимой (примитивно рекурсивной) функцией, а вместе с ней и $\overline{\mathrm{sg}}(F(x_1, x_1, x_2, \ldots, x_k))$ (поскольку функция $F(x_1, x_1, x_2, \ldots, x_k)$ всюду определена). Следовательно, существует e_0 такое, что
$F(e_0, x_1, x_2, \ldots, x_k) = \overline{\mathrm{sg}}(F(x_1, x_1, x_2, \ldots, x_k))$ для всех натуральных чисел x_1, x_2, ..., x_k. В частности,
$F(e_0, e_0, \ldots, e_0) = \overline{\mathrm{sg}}(F(e_0, e_0, \ldots, e_0))$, противоречие. $\square$

Универсальные функции

Доказательство.

Допустим, что $F(x_0, x_1, x_2, \ldots, x_k)$ — соответствующая универсальная функция. Тогда $F(x_1, x_1, x_2, \ldots, x_k)$ является k-местной вычислимой (примитивно рекурсивной) функцией, а вместе с ней и $\overline{\mathrm{sg}}(F(x_1, x_1, x_2, \ldots, x_k))$ (поскольку функция $F(x_1, x_1, x_2, \ldots, x_k)$ всюду определена). Следовательно, существует e_0 такое, что $F(e_0, x_1, x_2, \ldots, x_k) = \overline{\mathrm{sg}}(F(x_1, x_1, x_2, \ldots, x_k))$ для всех натуральных чисел x_1, x_2, …, x_k. В частности, $F(e_0, e_0, \ldots, e_0) = \overline{\mathrm{sg}}(F(e_0, e_0, \ldots, e_0))$, противоречие. □

Теорема C5

Каково бы ни было $k \geqslant 1$, существует $k+1$-местная универсальная частично вычислимая функция.

Универсальные функции

Доказательство.

Действительно, таковой функцией является $U(\mu y.T_k(x_0, x_1, \ldots, x_k, y))$. $\square$

Универсальные функции

Доказательство.

Действительно, таковой функцией является $U(\mu y.T_k(x_0, x_1, \ldots, x_k, y))$. □

Теорема С6

Каково бы ни было $k \geqslant 1$, существует $k+1$-местная частично вычислимая функция, универсальная для семейства всех k-местных чвф, принимающих значения $\subseteq \{0; 1\}$.

Универсальные функции

Доказательство.

Действительно, таковой функцией является $U(\mu y.T_k(x_0, x_1, \ldots, x_k, y))$. $\square$

Теорема С6

Каково бы ни было $k \geqslant 1$, существует $k+1$-местная частично вычислимая функция, универсальная для семейства всех k-местных чвф, принимающих значения $\subseteq \{0; 1\}$.

Доказательство.

Пусть $F(x_0, x_1, \ldots, x_k)$ — универсальная чвф; покажем, что функция $F'(x_0, x_1, \ldots, x_k) = \mathrm{sg}(F(x_0, x_1, \ldots, x_k))$ удовлетворяет условию теоремы. Действительно, чвф $F'(x_0, x_1, \ldots, x_k)$ принимает значения $\subseteq \{0; 1\}$ и, в частности, для любого $e_0 \in \omega$ функция $\lambda x_1 x_2 \ldots x_k.F'(e_0, x_1, x_2, \ldots, x_k)$ принимает значения $\subseteq \{0; 1\}$.

Универсальные функции

Доказательство (продолжение)

Далее, пусть $\varphi(x_1, x_2, \ldots, x_k)$ — чвф, принимающая значения $\subseteq \{0; 1\}$. Тогда существует $e_1 \in \omega$ такое, что $\varphi(x_1, x_2, \ldots, x_k) = F(e_1, x_1, \ldots, x_k) = \mathrm{sg}(F(e_1, x_1, \ldots, x_k)) = F'(e_1, x_1, \ldots, x_k)$. $\square$

Нумерация Кантора ω^2

Определение

Определим следующую функцию $c^2 : \omega^2 \to \omega$

$$c^2(x,y) \leftrightharpoons \frac{(x+y)^2 + 3x + y}{2}.$$

Эта функция называется **нумерацией Кантора** и осуществляет биективное отображение $c^2 : \omega^2 \overset{1:1}{\twoheadrightarrow} \omega$ (**упражнение!!!**)

Нумерация Кантора ω^2

Определение

Определим следующую функцию $c^2 : \omega^2 \to \omega$

$$c^2(x, y) \leftrightharpoons \frac{(x+y)^2 + 3x + y}{2}.$$

Эта функция называется **нумерацией Кантора** и осуществляет биективное отображение $c^2 : \omega^2 \overset{1:1}{\twoheadrightarrow} \omega$ (**упражнение!!!**)

Лемма C12

Существуют такие примитивно рекурсивные функции $l : \omega \to \omega$ и $r : \omega \to \omega$, что для всех $x, y \in \omega$

- $c^2(l(x), r(x)) = x$;
- $l(c^2(x, y)) = x$;
- $r(c^2(x, y)) = y$.

Нумерация Кантора ω^n

Определение

Для каждого $1 \leqslant n \in \omega$ определим нумерацию Кантора $c^n : \omega^n \xrightarrow{1:1} \omega$ индукцией по n: $c^1(x) \leftrightharpoons x$,

$$c^n(x_1, \ldots, x_n) \leftrightharpoons c(c^{n-1}(x_1, x_2, \ldots, x_{n-1}), x_n).$$

Нумерация Кантора ω^n

Определение

Для каждого $1 \leqslant n \in \omega$ определим нумерацию Кантора $c^n : \omega^n \overset{1:1}{\twoheadrightarrow} \omega$ индукцией по n: $c^1(x) \leftrightharpoons x$,

$$c^n(x_1, \ldots, x_n) \leftrightharpoons c(c^{n-1}(x_1, x_2, \ldots, x_{n-1}), x_n).$$

Лемма C13

Для каждого $1 < n \in \omega$ существуют такие примитивно рекурсивные функции $c_{n,i} : \omega \to \omega$, где $1 \leq i \leq n$, что для всех $x_1, \ldots, x_n \in \omega$

- $c^n(c_{n,1}(x), \ldots, c_{n,n}(x)) = x$;
- $c_{n,i}(c^n(x_1, \ldots, x_n)) = x_i$ для всех $1 \leq i \leq n$.

Нумерация Кантора ω^n

Определение

Для каждого $1 \leqslant n \in \omega$ определим нумерацию Кантора $c^n : \omega^n \overset{1:1}{\twoheadrightarrow} \omega$ индукцией по n: $c^1(x) \leftrightharpoons x$,

$$c^n(x_1, \ldots, x_n) \leftrightharpoons c(c^{n-1}(x_1, x_2, \ldots, x_{n-1}), x_n).$$

Лемма С13

Для каждого $1 < n \in \omega$ существуют такие примитивно рекурсивные функции $c_{n,i} : \omega \to \omega$, где $1 \leq i \leq n$, что для всех $x_1, \ldots, x_n \in \omega$

- $c^n(c_{n,1}(x), \ldots, c_{n,n}(x)) = x$;
- $c_{n,i}(c^n(x_1, \ldots, x_n)) = x_i$ для всех $1 \leq i \leq n$.

Упражнение

Докажите леммы С12 и С13.

Унарные vs k-местные

Лемма C14

Пусть ψ — k-местная функция и пусть $A \subseteq \omega^k$ — множество. Тогда

1. ψ частично вычислима, если и только если $\psi(c_{k,1}(x), c_{k,2}(x), \ldots, c_{k,k}(x))$ частично вычислима;
2. ψ вычислима, если и только если $\psi(c_{k,1}(x), c_{k,2}(x), \ldots, c_{k,k}(x))$ вычислима;
3. ψ примитивно рекурсивна, если и только если $\psi(c_{k,1}(x), c_{k,2}(x), \ldots, c_{k,k}(x))$ примитивно рекурсивна;
4. A вычислимо, если и только если $c^k(A)$ вычислимо.

Унарные vs k-местные

Лемма С14

Пусть ψ — k-местная функция и пусть $A \subseteq \omega^k$ — множество. Тогда

1. ψ частично вычислима, если и только если $\psi(c_{k,1}(x), c_{k,2}(x), \ldots, c_{k,k}(x))$ частично вычислима;
2. ψ вычислима, если и только если $\psi(c_{k,1}(x), c_{k,2}(x), \ldots, c_{k,k}(x))$ вычислима;
3. ψ примитивно рекурсивна, если и только если $\psi(c_{k,1}(x), c_{k,2}(x), \ldots, c_{k,k}(x))$ примитивно рекурсивна;
4. A вычислимо, если и только если $c^k(A)$ вычислимо.

Доказательство.

1) ($\Rightarrow$) Действительно, если $\psi(x_1, x_2, \ldots, x_k)$ — частично вычислимая функция, то и $\psi(c_{k,1}(x), c_{k,2}(x), \ldots, c_{k,k}(x))$ также является чвф, как функция, полученная из частично вычислимых с помощью оператора суперпозиции.

Унарные vs k-местные

Доказательство (продолжение)

($\Leftarrow$) Пусть теперь $\psi(c_{k,1}(x), c_{k,2}(x), \ldots, c_{k,k}(x))$ — чвф; тогда $\psi(x_1, x_2, \ldots, x_k) = \psi(c_{k,1}(c^k(x_1, x_2, \ldots, x_k)), c_{k,2}(c^k(x_1, x_2, \ldots, x_k)), \ldots, c_{k,k}(c^k(x_1, x_2, \ldots, x_k)))$ также является чвф.

2,3) Рассматриваются аналогично рассмотренному случаю.

4) ($\Rightarrow$) Пусть A —вычислимое множество. Следовательно, $\chi_A(x_1, x_2, \ldots, x_k)$ — вычислимая функция, а вместе с ней и $\chi_{c^k(A)}(x) = \chi_A(c_{k,1}(x), c_{k,2}(x), \ldots, c_{k,k}(x))$ также является вф. Таким образом, $c^k(A)$ — вычислимое множество.

($\Leftarrow$) Пусть $\chi_{c^k(A)}$ — характеристическая функция множества $c^k(A)$. Определим характеристическую функцию для множества A:

$$\chi_A(x_1, x_2, \ldots, x_k) = \chi_{c^k(A)}(c^k(x_1, x_2 \ldots, x_k)).$$

Таким образом, A вычислимо. $\square$

Унарные vs k-местные

Лемма C15

Пусть ψ — унарная функция и пусть $A \subseteq \omega$ — множество. Тогда

1. ψ частично вычислима, если и только если $\psi(c^k(x_1, x_2, \dots, x_k))$ частично вычислима;
2. ψ вычислима, если и только если $\psi(c^k(x_1, x_2, \dots, x_k))$ вычислима;
3. ψ примитивно рекурсивна, если и только если $\psi(c^k(x_1, x_2, \dots, x_k))$ примитивно рекурсивна;
4. A вычислимо, если и только если $B = \{\langle c_{k,1}(x), c_{k,2}(x), \dots, c_{k,k}(x)\rangle | x \in A\}$ вычислимо.

Унарные vs k-местные

Лемма С15

Пусть ψ — унарная функция и пусть $A \subseteq \omega$ — множество. Тогда

1. ψ частично вычислима, если и только если $\psi(c^k(x_1, x_2, \ldots, x_k))$ частично вычислима;
2. ψ вычислима, если и только если $\psi(c^k(x_1, x_2, \ldots, x_k))$ вычислима;
3. ψ примитивно рекурсивна, если и только если $\psi(c^k(x_1, x_2, \ldots, x_k))$ примитивно рекурсивна;
4. A вычислимо, если и только если $B = \{\langle c_{k,1}(x), c_{k,2}(x), \ldots, c_{k,k}(x)\rangle | x \in A\}$ вычислимо.

Доказательство.

1) ($\Rightarrow$) Действительно, если $\psi(x)$ — частично вычислимая функция, то и $\psi(c^k(x_1, x_2, \ldots, x_k))$ также является чвф, как функция, полученная из частично вычислимых с помощью оператора суперпозиции.

Унарные vs k-местные

Доказательство (продолжение)

($\Leftarrow$) Пусть теперь $\psi(c^k(x_1, x_2, \ldots, x_k))$ — чвф; тогда $\psi(x) = \psi(c^k(c_{k,1}(x), c_{k,2}(x), \ldots, c_{k,k}(x)))$ также является чвф.

2,3) Рассматриваются аналогично рассмотренному случаю.

4) ($\Rightarrow$) Пусть A —вычислимое множество. Следовательно, $\chi_A(x)$ — вычислимая функция, а вместе с ней и $\chi_B(x_1, x_2, \ldots, x_k) = \chi_A(c^k(x_1, x_2, \ldots, x_k))$ также является вф. Таким образом, B — вычислимое отношение.

($\Leftarrow$) Пусть χ_B — характеристическая функция множества B. Тогда $\chi_A(x) = \chi_B(c_{k,1}(x), c_{k,2}(x), \ldots, c_{k,k}(x)) = \chi_A(c^k(c_{k,1}(x), c_{k,2}(x), \ldots, c_{k,k}(x)))$. Таким образом, A вычислимо. $\square$

Унарные vs k-местные

Доказательство (продолжение)

$(\Leftarrow)$ Пусть теперь $\psi(c^k(x_1, x_2, \dots, x_k))$ — чвф; тогда $\psi(x) = \psi(c^k(c_{k,1}(x), c_{k,2}(x), \dots, c_{k,k}(x)))$ также является чвф.
2,3) Рассматриваются аналогично рассмотренному случаю.
4) $(\Rightarrow)$ Пусть A —вычислимое множество. Следовательно, $\chi_A(x)$ — вычислимая функция, а вместе с ней и $\chi_B(x_1, x_2, \dots, x_k) = \chi_A(c^k(x_1, x_2, \dots, x_k))$ также является вф. Таким образом, B — вычислимое отношение.
$(\Leftarrow)$ Пусть χ_B — характеристическая функция множества B. Тогда $\chi_A(x) = \chi_B(c_{k,1}(x), c_{k,2}(x), \dots, c_{k,k}(x)) = \chi_A(c^k(c_{k,1}(x), c_{k,2}(x), \dots, c_{k,k}(x)))$. Таким образом, A вычислимо. $\square$

Замечание

Трансформации, описанные в леммах С14 и С15, взаимно обратны. К примеру, $\psi_0 \in \mathrm{PCF}_k \overset{\mathrm{C14(1)}}{\mapsto} \psi_1 \in \mathrm{PCF}_1 \overset{\mathrm{C15(1)}}{\mapsto} \psi_2 \in \mathrm{PCF}_k$ и $\psi_0 \in \mathrm{PCF}_1 \overset{\mathrm{C15(1)}}{\mapsto} \psi_1 \in \mathrm{PCF}_k \overset{\mathrm{C14(1)}}{\mapsto} \psi_2 \in \mathrm{PCF}_1$ тождественны.

Унарные vs k-местные

Лемма С16

Пусть $\varphi(x_0, x_1)$ — чвф и пусть $k \geqslant 1$. Тогда $\varphi(x_0, x_1)$ — универсальная, если и только если $\varphi(x_0, c^k(x_1, x_2, \ldots, x_k))$ также универсальная.

Унарные vs k-местные

Лемма C16

Пусть $\varphi(x_0, x_1)$ — чвф и пусть $k \geqslant 1$. Тогда $\varphi(x_0, x_1)$ — универсальная, если и только если $\varphi(x_0, c^k(x_1, x_2, \ldots, x_k))$ также универсальная.

Доказательство.

($\Rightarrow$) Пусть $\varphi(x_0, x_1)$ — универсальная чвф; покажем, что $\varphi(x_0, c^k(x_1, x_2, \ldots, x_k))$ является также универсальной чвф. Действительно, пусть $\psi(x_1, x_2, \ldots, x_k)$ — чвф; по лемме C14(1), $\psi'(x) = \psi(c_{k,1}(x), c_{k,2}(x), \ldots, c_{k,k}(x))$ — чвф и, следовательно, $\varphi(e_0, x) = \psi'(x)$ для некоторого e_0. Значит, $\varphi(e_0, c^k(x_1, x_2, \ldots, x_k)) = \psi'(c^k(x_1, x_2, \ldots, x_k)) = \psi(c_{k,1}(c^k(x_1, x_2, \ldots, x_k)), c_{k,2}(c^k(x_1, x_2, \ldots, x_k)), \ldots, c_{k,k}(c^k(x_1, x_2, \ldots, x_k))) = \psi(x_1, x_2, \ldots, x_k)$.

Унарные vs k-местные

Доказательство (продолжение)

($\Leftarrow$) Пусть $\varphi(x_0, c^k(x_1, x_2, \ldots, x_k))$ — универсальная чвф; покажем, что $\varphi(x_0, x_1)$ также универсальная чвф. Пусть $\theta(x)$ — чвф; по лемме C15(1), $\theta'(x_1, x_2, \ldots, x_k) = \theta(c^k(x_1, x_2, \ldots, x_k))$ также является чвф. Следовательно, $\varphi(e_1, c^k(x_1, x_2, \ldots, x_k)) = \theta'(x_1, x_2, \ldots, x_k)$ для некоторого e_1. Значит, $\varphi(e_1, x) = \varphi(e_1, c^k(c_{k,1}(x), c_{k,2}(x), \ldots, c_{k,k}(x))) = \theta'(c_{k,1}(x), c_{k,2}(x), \ldots, c_{k,k}(x)) = \theta(c^k(c_{k,1}(x), c_{k,2}(x), \ldots, c_{k,k}(x))) = \theta(x)$. $\square$

Унарные vs k-местные

Доказательство (продолжение)

($\Leftarrow$) Пусть $\varphi(x_0, c^k(x_1, x_2, \dots, x_k))$ — универсальная чвф; покажем, что $\varphi(x_0, x_1)$ также универсальная чвф. Пусть $\theta(x)$ — чвф; по лемме C15(1), $\theta'(x_1, x_2, \dots, x_k) = \theta(c^k(x_1, x_2, \dots, x_k))$ также является чвф. Следовательно, $\varphi(e_1, c^k(x_1, x_2, \dots, x_k)) = \theta'(x_1, x_2, \dots, x_k)$ для некоторого e_1. Значит, $\varphi(e_1, x) = \varphi(e_1, c^k(c_{k,1}(x), c_{k,2}(x), \dots, c_{k,k}(x))) = \theta'(c_{k,1}(x), c_{k,2}(x), \dots, c_{k,k}(x)) = \theta(c^k(c_{k,1}(x), c_{k,2}(x), \dots, c_{k,k}(x))) = \theta(x)$. $\square$

Лемма C17

Пусть $k \geqslant 1$ и пусть $\varphi(x_0, x_1, x_2, \dots, x_k)$ — чвф. Тогда $\varphi(x_0, x_1, \dots, x_k)$ — универсальная, если и только если $\varphi(x_0, c_{k,1}(x_1), c_{k,2}(x_1), \dots, c_{k,k}(x_1))$ также универсальная.

Унарные vs k-местные

Доказательство.

($\Rightarrow$) Пусть $\varphi(x_0, x_1, \ldots, x_k)$ — универсальная чвф; покажем, что $\varphi(x_0, c_{k,1}(x_1), c_{k,2}(x_1), \ldots, c_{k,k}(x_1))$ также универсальная чвф. Пусть $\psi(x)$ — чвф; по лемме С15(1), $\psi'(x_1, x_2, \ldots, x_k) = \psi(c^k(x_1, x_2, \ldots, x_k))$ также является чвф. Тогда $\varphi(e_0, x_1, x_2, \ldots, x_k) = \psi'(x_1, x_2, \ldots, x_k)$ для некоторого e_0 и, следовательно, $\varphi(e_0, c_{k,1}(x), c_{k,2}(x), \ldots, c_{k,k}(x)) = \psi'(c_{k,1}(x), c_{k,2}(x), \ldots, c_{k,k}(x)) = \psi(c^k(c_{k,1}(x), c_{k,2}(x), \ldots, c_{k,k}(x))) = \psi(x)$.

($\Leftarrow$) Пусть $\varphi(x_0, c_{k,1}(x_1), c_{k,2}(x_1), \ldots, c_{k,k}(x_1))$ — универсальная чвф; покажем, что $\varphi(x_0, x_1, \ldots, x_k)$ также универсальная чвф. Пусть $\theta(x_1, x_2, \ldots, x_k)$ — чвф; по лемме С14(1), $\theta'(x) = \theta(c_{k,1}(x), c_{k,2}(x), \ldots, c_{k,k}(x))$ также является чвф и, следовательно, $\varphi(e_1, c_{k,1}(x), c_{k,2}(x), \ldots, c_{k,k}(x)) = \theta'(x)$ для некоторого e_1. Значит, $\varphi(e_1, c_{k,1}(c^k(x_1, x_2, \ldots, x_k)), c_{k,2}(c^k(x_1, x_2, \ldots, x_k)), \ldots, c_{k,k}(c^k(x_1, x_2, \ldots, x_k))) = \theta'(c^k(x_1, x_2, \ldots, x_k)) = \theta(c_{k,1}(c^k(x_1, x_2, \ldots, x_k)), c_{k,2}(c^k(x_1, x_2, \ldots, x_k)), \ldots, c_{k,k}(c^k(x_1, x_2, \ldots, x_k))) = \theta(x_1, x_2, \ldots, x_k)$. $\square$

Не-вычислимые функции

Следствие C2

Каково бы ни было $k \geqslant 1$, существует всюду определённая k-местная функция, принимающая значения $\subseteq \{0; 1\}$, не являющаяся вычислимой.

Не-вычислимые функции

Следствие C2

Каково бы ни было $k \geqslant 1$, существует всюду определённая k-местная функция, принимающая значения $\subseteq \{0;1\}$, не являющаяся вычислимой.

Доказательство.

Из леммы C15(1) вытекает, что достаточно привести пример одноместной такой функции. Пусть $F(x_0, x_1)$ — чвф, универсальная для семейства всех унарных чвф, принимающих значения $\subseteq \{0;1\}$, и пусть $X_0 = \{e | \lambda x.F(e,x) \text{ вычислима}\}$. Множество X_0 счётно как бесконечное подмножество счётного множества. Возьмём $f_0 : \omega \overset{1:1}{\twoheadrightarrow} X_0$ и положим $f(x) = F(f_0(l(x)), r(x))$. Отметим, что функция $f(x)$ всюду определена.

Не-вычислимые функции

Доказательство (продолжение).

Если бы $f(x)$ была вычислимой, то и $f(c(x, y))$ также была бы вычислимой. Однако в этом случае $f(c(x, y)) = F(f_0(l(c(x, y))), r(c(x, y))) = F(f_0(x), y)$ является вычислимой функцией, универсальной для семейства всех одноместных вычислимых функций, принимающих значения $\subseteq \{0; 1\}$, противоречие предложению С12. $\square$