УДК 510.6

ВЫРАЗИТЕЛЬНЫЕ ВОЗМОЖНОСТИ ОБОБЩЕННЫХ ФОРМУЛ

Попов С.В.

ООО «Научно-внедренческая фирма БП+», Москва, *e-mail*:s-v-popov@yandex.ru

Обобщенный пропозициональный язык получается путем расширения области определения обычных логических связок на счетные аргументы. Такой язык обладает интересными семантиками и оказывается полезным при решении различных прикладных задач, в том числе связанными с анализом вычислительных моделей. Однако, если не вводить ограничений на обобщенные логические связки, то получается язык с весьма богатыми выразительными возможностями. Но именно по этой причине его трудно исследовать для получения реальных результатов, касающихся вычислимых функций. Поэтому такой язык не представляет интереса в силу ограниченности его применения. С другой стороны, если ввести достаточно естественные ограничения на бесконечные связки, удается получить ограниченный язык т.н. примитивных формул с интересными свойствами, которые описаны в настоящей статье. С этой целью вводится понятие *представления* вычислимых функций обобщенными формулами. Двоичная функция $y = f(x)$ представима обобщенной формулой $F(x, y)$, если для каждой пары значений σ_x, σ_y соответственно аргумента и значения функции f равенство $\sigma_y = f(\sigma_x)$ выполняется тогда и только тогда, когда формула $F(\sigma_x, \sigma_y)$ истинная. В статье показывается, что все функции арифметики Пресбургера представимы примитивными формулами обобщенного языка.

Ключевые слова: бесконечные логические связки, обобщенный пропозициональный язык, примитивные формулы, представимость функций, арифметика Пресбургера.

EXPRESSIVE POSSIBILITIES OF GENERALIZED FORMULAS

Popov S. V.

LLC "Nauchno-vnedrencheskaya firma BP+", Moscow, e-mail: s-v-popov@yandex.ru

A generalized propositional language is obtained by extending the domain of definition of ordinary logical connectives to countable arguments. Such a language has interesting semantics and is useful in solving various applied problems, including those related to the analysis of computational models. However, if you do not impose restrictions on generalized logical connectives, then you get a language with very rich expressive capabilities. But it is precisely for this reason that it is difficult to investigate it in order to obtain real results concerning computable functions. Therefore, such a language is of no interest due to its limited application. On the other hand, if we introduce sufficiently natural restrictions on infinite bundles, it is possible to obtain a limited language of so-called primitive formulas with interesting properties, which are described in this article. For this purpose, the concept of representation of computable functions by generalized formulas is introduced. Binary function $y = f(x)$ is representable by generalized by the formula $F(x, y)$, if for each pair of values σ_x, σ_y respectively argument and function values f equality $\sigma_y = f(\sigma_x)$ is true if and only if the formula $F(\sigma_x, \sigma_y)$ is true. The article shows that all the functions of Presburger arithmetic are representable by primitive formulas of a generalized language.

Keywords: infinite logical connectives, generalized propositional language, primitive formulas, representability of functions, Presburger's arithmetic.

Введение. Исходя из практических задач анализа информационных моделей, был введен т.н. обобщенный пропозициональный язык, который получается расширение области определения обычных логических связок на счетные аргументы [1]. Такой язык обладает различными интересными семантиками, и оказывается полезным при решении различных прикладных задач, в том числе связанными с анализом вычислительных моделей [2, 3, 4]. Однако, если не вводить ограничений на обобщенные логические связки, то получается язык с весьма богатыми выразительными возможностями. Но именно поэтому его трудно применять для получения реальных результатов, касающихся вычислимых функций.

Поэтому такой язык не представляет интереса в силу ограниченности его применения. С другой стороны, если ввести достаточно естественные ограничения на бесконечные связки, удается получить ограниченный язык т.н. примитичных формул с интересными свойствами, которые описаны в настоящей статье. С этой целью вводится понятие *представления* вычислимых функций обобщенными формулами. Двоичная функция $\boldsymbol{y} = f(\boldsymbol{x})$ представима обобщенной формулой $F(\boldsymbol{x}, \boldsymbol{y})$, если для каждой пары значений $\boldsymbol{\sigma}_x$, $\boldsymbol{\sigma}_y$ соответственно аргумента и значения функции f равенство $\boldsymbol{\sigma}_y = f(\boldsymbol{\sigma}_x)$ выполняется тогда и только тогда, когда формула $F(\boldsymbol{\sigma}_x, \boldsymbol{\sigma}_y)$ истинная. В статье показывается, что все функции арифметики Пребургера представимы примитивными формулами обобщенного языка.

1. Представление вычислимых функций обобщенными формулами. Исходя из практических соображений, в множестве обобщенных формул, определенных в [1], выделено семейство примитивных формул, обладающих относительно простыми ограничениями на бесколнечные логические связки. В этом разделе исследуем выразительные возможности примитивного языка. С этой целью покажем, что с помощью примитивных формул можно описывать свойства некоторых функций двоичной арифметики.

При обозначении двоичных чисел младшие разряды будем располагать слева. Поэтому вектор $\boldsymbol{\sigma}$, представляющий двоичное число, содержит ненулевыми лишь конечное число левых разрядов. Все разряды, расположенные справа от самого правого единичного – нулевые. Таким образом, если бесконечный вектор обозначает двоичное число, то он имеет бесконечный нулевой суффикс.

Будем говорить, что логическая формула $F(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n, \boldsymbol{y})$ *представляет* двоичную функцию $\boldsymbol{y} = f(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n)$, если при означивании переменных $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n, \boldsymbol{y}$ соответственно бинарными векторами $\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2, \ldots, \boldsymbol{\sigma}_n, \boldsymbol{\sigma}_{n+1}$ формула $F(\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2, \ldots, \boldsymbol{\sigma}_n, \boldsymbol{\sigma}_{n+1})$ истинна тогда и только тогда, когда $f(\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2, \ldots, \boldsymbol{\sigma}_n) = \boldsymbol{\sigma}_{n+1}$. Отметим, что формула $F(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n, \boldsymbol{y})$ может содержать и другие переменные, кроме указанных. Назовем их *рабочими*.

Просто доказывается следующая лемма.

***Лемма* 1.1.** *Если формула $F(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n, \boldsymbol{y})$ представляет функцию $f(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n)$ двоичной арифметики, то д-эквивалентная ей формула $G(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n, \boldsymbol{y})$ также представляет эту функцию.*

***Пример* 1.1.** Обозначим $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$ наборы двоичных переменных соответственно: $x_0, x_1, \ldots, x_n, \ldots$; $y_0, y_1, \ldots, y_n, \ldots$ и $z_0, z_1, \ldots, z_n, \ldots$. Тогда формула $\text{Сл}_1(\boldsymbol{x}, \boldsymbol{y}) = (y_0 \equiv \overline{x}_0) \cap_{i=1,\,\omega} (y_i \equiv x_i \oplus \cap_{j=0,\,i-1} x_j)$ представляет двоичную функцию прибавления единицы. В этой формуле конъюнкция $\cap_{j=0,\,i-1} x_i$ представляет перенос в i-й разряд, он равен 1 тогда и только тогда, когда все разряды от нулевого до $(i - 1)$-го единичные.

Согласно преобразованиям из [1], формула

$$\text{Сл}_1(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{y}): (y_0 \equiv \bar{x}_0)\ (u_0 \equiv x_0) \cap_{i=1,\omega} (y_i \equiv x_i \oplus u_i) \cap_{i=0,\omega} (u_{i+1} \equiv u_i\, x_i)$$

д-эквивалентна первоначальной и также представляет функцию прибавления единицы. Легко увидеть, что новая переменная u_i определяет перенос в i-ый разряд, который вычисляется после прибавления 1 к вектору с разрядами от нулевого до (i – 1)-го включительно. Перенос в нулевой разряд совпадает с нулевым разрядом самого числа.

Пример **1.2.** Формула

$$\text{Сл}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = (z_0 \equiv x_0 \oplus y_0)(u_1 \equiv x_0 y_0) \cap_{i=1,\omega} \{(z_i \equiv x_i \oplus y_i \oplus u_i)(u_{i+1} \equiv u_i(x_i \vee y_i) \vee x_i y_i)\}$$

представляет сложение $\boldsymbol{x} + \boldsymbol{y}$. Здесь $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$ суть соответственно значения аргументов и суммы, а каждая переменная u_i, i = 1, 2, … обозначает перенос в i–й разряд.

Пример **1.3.** По аналогии с формулой Сл$_1$($\boldsymbol{x}$, $\boldsymbol{y}$) строится формула Выч$_1$($\boldsymbol{x}$, $\boldsymbol{y}$), представляющая разность $\boldsymbol{y} = \boldsymbol{x}$ - 1:

$$\text{Выч}_1(\boldsymbol{x}, \boldsymbol{y}) = \cup_{i=0,\,\omega} x_i \supset \{(y_0 \equiv \bar{x}_0) \cap_{i=0,\omega}(y_i \equiv x_i \oplus \cap_{j=0,\, i-1} x_j)\}\ \{\cap_{i=0,\omega} \bar{x}_i \supset \cap_{i=0,\,\omega} \bar{y}_i\}.$$

В этой формуле $\boldsymbol{x}, \boldsymbol{y}$ соответственно значения аргументов и разности. При этом первый член конъюнкции является импликацией, посылка которой есть проверка условии, что аргумент не нулевой. В этом случае разность на 1 меньше аргумента. Заключение этой импликации представляет собой конъюнкцию. Первый ее член показывает, что нулевой компонент результата есть отрицание нулевого компонента аргумента. Второй - что занимаем мы из следующего разряда только в случае, когда рассматриваемый компонент аргумента равен нулю. Второй член формулы – также импликация, посылка которой истинна, если аргумент равен 0. Но тогда значение разности нулевое.

Избавившись от вложенности обобщенных связок, получим формулу Выч$_1$($\boldsymbol{x}$, $\boldsymbol{u}$, $\boldsymbol{y}$) = $\cup_{i=0,\,\omega} x_i \supset \{(y_0 \equiv \bar{x}_0)(u_1 \equiv \bar{x}_0) \cap_{i=1,\,\omega} (y_i \equiv x_i \oplus u_i) \wedge \wedge (u_{i+1} \equiv u_i\, \bar{x}_i)\} \wedge \{\cap_{i=0,\,\omega} \bar{x}_i \supset \cap_{i=0,\,\omega} \bar{y}_i\}$. Ее конъюнкция $\cap_{i=1,\,\omega}(y_i \equiv x_i \oplus u_i)(u_{i+1} \equiv u_i\, \bar{x}_i)$ описывает вычисление компонентов разности и занимаемой в старшем разряде величины. Значение u_i обозначает занимаемую величину в i+1-м разряде, в зависимости от наличия занимания в i-м разряде и значения i-го компонента аргумента.

Пример **1.4.** Аналогично представляются предикаты равенства и неравенства: $\boldsymbol{x} = \boldsymbol{y}$: $\cap_{i=0,\omega} x_i \equiv y_i$; $\boldsymbol{x} > 0 : \cup_{i=0,\omega} x_i$.

Пример **1.5.** Конъюнкция $\cap_{i=0,\omega} x_i^{\sigma_i}$ представляет двоичную константу $\sigma_0, \sigma_1, \sigma_2, \ldots$.

Справедлива такая теорема.

Теорема **1.2.** *Пусть формулы* $F(\boldsymbol{x}, \boldsymbol{y})$ *и* $G(\boldsymbol{y}, \boldsymbol{z})$ *представляют двоичные функции соответственно* $\boldsymbol{y} = f(\boldsymbol{x})$ *и* $\boldsymbol{z} = g(\boldsymbol{y})$. *Тогда конъюнкция* $F(\boldsymbol{x}, \boldsymbol{y})G(\boldsymbol{y}, \boldsymbol{z})$ *представляет суперпозицию* $\boldsymbol{z} = f(g(\boldsymbol{y}))$.

***Пример* 1.6.** Используя последнюю теорему о суперпозиции функций, приведем ряд формул, представляющих арифметические функции и отношения. Для начала построим формулу, представляющую функцию $x + 2$, суперпозицией двух функций прибавления единицы. Тогда формула, представляющая функцию прибавления 2, представляет собой конъюнкцию $\text{Сл}_1(\boldsymbol{x}, \boldsymbol{y}) \wedge \text{Сл}_1(\boldsymbol{y}, \boldsymbol{z}) = (y_0 \equiv \overline{x}_0) \cap_{i=1,\,\omega}(y_i \equiv x_i \oplus \cap_{j=0,\,i\text{-}1}\, x_j)(z_0 \equiv \overline{y}_0) \cap_{k=1,\,\omega}(z_k \equiv y_k \oplus \cap_{h=0,\,k\text{-}1}\, y_h) \equiv (y_0 \equiv \overline{x}_0)(z_0 \equiv \overline{y}_0)\cap_{i=1,\,\omega}(y_i \equiv x_i \oplus \cap_{j=0,\,i\text{-}1}\, x_j)\cap_{k=1,\,\omega}(z_k \equiv y_k \oplus \cap_{h=0,\,k\text{-}1}\, y_h)$.

Преобразуя, получаем $z_0 \equiv x_0$. Используя две эквивалентности $z_1 \equiv y_1 \oplus y_0$, $y_1 \equiv x_1 \oplus x_0$, получаем $z_1 \equiv x_1 \oplus x_0 \oplus \overline{x}_0 \equiv x_1$. Вспользуя две эквивалентности $z_2 \equiv y_2 \oplus y_0y_1$, $y_2 \equiv x_2 \oplus x_0x_1$, получаем $z_2 \equiv x_2 \oplus x_0x_1 \oplus \overline{x}_0\ (x_1 \oplus x_0) \equiv x_2 \oplus x_0x_1 \oplus \overline{x}_0x_1 \equiv x_2 \oplus x_1$. Используя две эквивалентности $z_3 \equiv y_3 \oplus y_0y_1y_2$, $y_3 \equiv x_3 \oplus x_0x_1x_2$, получаем $z_3 \equiv x_3 \oplus x_0x_1x_2 \oplus \overline{x}_0(x_1 \oplus x_0)(x_2 \oplus x_0x_1) \equiv x_3 \oplus x_1x_2$. Индукцией по k нетрудно доказать, что $z_k \equiv x_k \oplus \cap_{h=1,\,k\text{-}1}\, x_h$, при $k = 2, 3, \ldots$. Окончательная формула выглядит так: $(y_0 \equiv x_0)(y_1 \equiv \overline{x}_1) \cap_{i=2,\omega}\, y_i \equiv x_i \oplus \cap_{j=1,\,i\text{-}1}\, x_j$. Формулы, представляющие функции: $\boldsymbol{x} + 3$, $\boldsymbol{x} + 4$, $\boldsymbol{x} + 5$, $\boldsymbol{x} + 6$ выглядят как соответственно: $(y_0 \equiv \overline{x}_0)(y_1 \equiv x_0 \oplus \overline{x}_1) \cap_{i=2,\omega}(y_i \equiv x_i \oplus x_1 \oplus x_0\ \overline{x}_1 \cap_{j=2,\,i\text{-}1}\, x_j$, $(y_0 \equiv x_0)(y_1 \equiv x_1)(y_2 \equiv \overline{x}_2) \cap_{i=3,\omega}\, y_i \equiv x_i \oplus \cap_{j=2,\,i\text{-}1}\, x_j$, $(y_0 \equiv \overline{x}_0\ )(y_1 \equiv x_0 \oplus x_1)(y_2 \equiv \overline{x}_2 \oplus x_0x_1\ ) \cap_{i=3,\omega}\ y_i \equiv x_i \oplus(x_2 \oplus x_0\ x_1\ \overline{x}_2) \cap_{j=3,\ i\text{-}1}\ x_j$, $(y_0 \equiv \overline{x}_0)(y_1 \equiv \overline{x}_1)(y_2 \equiv x_1 \oplus \overline{x}_2)(\cap_{i=3,\omega} y_i \equiv x_i \oplus(x_2 \oplus x_1\ \overline{x}_2)\cap_{j=3,\,i\text{-}1}\, x_j$.

Предикат $\boldsymbol{y} \le \boldsymbol{x}$ представим формулой $\cap_{i=0,\omega}(x_i\ \overline{y}_i \supset \cup_{j=i+1,\omega}\ \overline{x}_j y_j)$. Напомним, что в нашем представлении двоичных чисел старшие разряды располагаются правее. Функция $\boldsymbol{z} = \boldsymbol{x} \div \boldsymbol{y}$ представима формулой $(\boldsymbol{x} \le \boldsymbol{y} \supset \cap_{i=0,\omega}\ \overline{z}_i) \wedge (\boldsymbol{x} > \boldsymbol{y} \supset (z_0 \equiv x_0 \oplus y_0) \wedge u_1 \equiv \overline{x}_0 y_0\ \wedge \cap_{i=1,\omega}(z_i \equiv x_i \oplus y_i \oplus u_i) \wedge (u_{i+1} \equiv \overline{x}_i y_i \vee u_i(\ \overline{x}_i \vee y_i)) \equiv (\boldsymbol{x} \le \boldsymbol{y} \supset \cap_{i=0,\omega}\ \overline{z}_i) \wedge (\boldsymbol{x} > \boldsymbol{y} \supset (z_0 \equiv x_0 \oplus y_0) \wedge u_1 \equiv \overline{x}_0 y_0\ \wedge \cap_{i=1,\omega}(z_i \equiv x_i \oplus y_i \oplus u_i) \wedge (u_{i+1} \equiv \overline{x}_i y_i \vee u_i(x_i \equiv y_i))$. Используя введенные соотношения, легко показать, что функция $\boldsymbol{z} = |\boldsymbol{x} - \boldsymbol{y}|$ вычисления абсолютного значения разности представима формулой: $(\boldsymbol{x} \le \boldsymbol{y} \supset \text{Выч}(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{z}))(\boldsymbol{y} < \boldsymbol{x} \supset \text{Выч}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}))$.

***Пример* 1.7**. Рассмотрим функцию $\boldsymbol{y} = SL(\boldsymbol{x})$ сдвига влево на один разряд бесконечного бинарного вектора. Эта функция представима формулой $SLF(\boldsymbol{x}, \boldsymbol{y}) = \bigcap\limits_{i=0,\omega} y_i \equiv x_{i+1}$. В случае, когда мы рассматриваем конечные векторы, пусть длины h, то функция $\boldsymbol{y} = SLh(\boldsymbol{x})$ представима формулой $SLFh(\boldsymbol{x}, \boldsymbol{y}) = \bigcap\limits_{i=0,h-2} (y_i \equiv x_{i+1}) \wedge \overline{y}_{h-1}$.

Справедлива следующая

***Теорема* 1.3.** *Функция* $\boldsymbol{y} = \boldsymbol{c}\boldsymbol{x}$ *двоичного умножения числа* $\boldsymbol{x}$ *на константу* $\boldsymbol{c}$ *представима примитивной формулой.*

Просто доказывается такая

***Теорема* 1.4**. *Пусть арифметическая функция* $\boldsymbol{y} = f(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n)$ *представима примитивной формулой* $F(\boldsymbol{y}, \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n)$ *и существует функция* $\boldsymbol{x}_i = f_i^{-1}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{i-1}, \boldsymbol{x}_{i+1}, \ldots, \boldsymbol{x}_n, \boldsymbol{y})$. *Тогда она также представима формулой* $F(\boldsymbol{y}, \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n)$.

***Пример* 1.8**. Формула $Сл_1(\boldsymbol{x}, \boldsymbol{y}) = (y_0 \equiv \overline{x}_0) \wedge \cap_{i=1,\ \omega}(y_i \equiv x_i \oplus \cap_{j=0,\ i-1}\ x_j)$ представляет функцию прибавления единицы. Функция вычитания единицы (при условии, что вычитание происходит не из нулевого аргумента) представима формулой $Выч_1(\boldsymbol{x}, \boldsymbol{y}) = (y_0 \equiv \overline{x}_0) \wedge \cap_{i=0,\omega}(y_i \equiv x_i \oplus \cap_{j=0,\ i-1}\ \bar{x}_j)$. После переименования переменных $Выч_1(\boldsymbol{x}, \boldsymbol{y})$ превращается в формулу $(y_0 \equiv \overline{x}_0) \wedge \cap_{i=0,\omega}(x_i \equiv y_i \oplus \cap_{j=0,\ i-1}\ \bar{y}_j)$.

Из последней теоремы вытекает, что формула $F(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{c})$ представляет функцию $\boldsymbol{x} = \boldsymbol{y}/\boldsymbol{c}$ целочисленного деления на константу $\boldsymbol{c}$. Из этого следует такое утверждение.

***Теорема* 1.5.** *Функция нахождения целочисленного решения линейного уравнения с целыми коэффициентами представима примитивной формулой.*

2. Представимость функций арифметики Пресбургера. Покажем, что все функции арифметики Пресбургера представимы примитивными формулами. Нам потребуется следующее определение программы.

Пусть $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots \}$ – *входной*, а $\boldsymbol{Y} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots\}$ – *рабочий* алфавиты. Буквы этих алфавитов назовем соответственно *входными* и *рабочими* переменными. Программа с такими входными и рабочими переменными - $\pi(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n, \boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_m)$ представляет собой орграф с вершинами трех типов: арифметические операторы $0(\boldsymbol{y})$ – присваивания переменной $\boldsymbol{y}$ нулевого значения, $s(\boldsymbol{y})$ – прибавление к переменной $\boldsymbol{y}$ единицы, логический оператор $u = v$, где u, v – рабочие или входные переменные и единственная заключительная вершина. Каждая арифметическая вершина имеет ровно одного последователя, логическая – двух (один из которых называется 0-последователем, другой 1-последователем), а заключительная вершина последователей не имеет. Заключительная вершина помечена единственной рабочей переменной, которая называется *выходной*. Кроме того, одна из вершин программы является *начальной*.

Функцией, *вычисляемой программой* π, называется частичная функция f_π такая, что $f_\pi(a_1, a_2, \ldots, a_n) = b \Leftrightarrow$ программа останавливается при значениях ее входных переменных $\boldsymbol{x}_1 = a_1, \boldsymbol{x}_2 = a_2, \ldots, \boldsymbol{x}_n = a_n$ и выходная переменная равна b.

Две программы *эквивалентны*, если они вычисляют одну функцию.

Существуют эквивалентные преобразования программ, которые позволяют получать программы с перечисленными ниже свойствами.

***Свойство* 1**. *Всякая программа эквивалентна такой, в которой в каждую арифметическую вершину ведет в точности одна дуга*.

Поэтому ограничимся рассмотрением только таких программ, в которых *несколько дуг могут вести только в логические вершины.*

***Свойство* 2.** *Всякая программа в обычном базисе эквивалентна программе в расширенном логическом базисе*: $c = u$, $u + c_1 = v + c_2$, где u, v – рабочие переменные, а c, c_1, c_2 – *константы, в которой в каждую логическую вершину ведут несколько дуг или ей непосредственно предшествует логическая вершина.*

Очевидно следующее утверждение.

***Лемма* 2.1.** *Пусть* π *есть программа, состоящая только из арифметических операторов. Тогда вычисляемая ею функция представляет собой прибавление константы, определяемой видом программы* π.

Рассмотрим случай, когда программа содержит логические вершины, но не имеет циклов. В данном случае справедливо такое утверждение.

***Теорема* 2.2**. *Функция, вычисляемая программой без циклов в расширенном логическом базисе, представима примитивной формулой.*

Справедлива такая

***Лемма* 2.3.** *Пусть* π *есть программа, представляющая собой единственный цикл, которому предшествует последовательность арифметических вершин, единственная логическая вершина имеет следующий вид: ее* σ-*дуга принадлежит циклу, а* (1-σ)-*дуга ведет в выходную вершину. Тогда программа* π *вычисляет функцию, представимую примитивной формулой.*

На самом деле, программа с единственным циклом может содержать и несколько логических вершин, через которые этот цикл проходит. В этом случае аналогичное утверждение также верно. Чтобы не приводить громоздкого доказательства, покажем это для случая, когда цикл содержит только две логические вершины $E(u, v)$: $u + c_1 = v + c_2$ и $E(s, t)$: $s + c_3 = t + c_4$. По Свойству 2 они соседние, пусть σ-дуга из $E(u, v)$ ведет в $E(s, t)$, λ-дуга которой принадлежит циклу, а (1 - λ)-дуга – не принадлежит.

Пусть в процессе однократного прохождения цикла переменные u, v, s, t получают приращение соответственно c_u, c_v, c_s, c_t. Для доказательства надо рассмотреть все значения пары (σ, λ): (0, 0), (0, 1), (1, 0), (1, 1).

1. Значение пары (σ, λ) есть (0, 0). В этом случае число прохождений цикла определяется как $n = min(n_1, n_2)$, где $n_1 = \dfrac{|u + c_1 - v - c_2|}{c_v - c_u}$, $n_2 = \dfrac{|s + c_3 - t - c_4|}{c_t - c_s}$. Но все эти функции представимы примитивными формулами. В итоге всякая рабочая переменная цикла

в результате вычисления увеличится на величину *cn*, где константа *c* определяется видом и расположением арифметических операторов цикла.

2. Значение пары (σ, λ) есть (0, 1). В этом случае число прохождений цикла определяется как $n = min(n_1, n_2)$, где $n_1 = \frac{|u + c_1 - v - c_2|}{c_v - c_u}$, а n_2 определяет выполняется или не выполняется равенство $s + c_3 = t + c_4$ для начальных значений переменных *s*, *t*. В любом случае цикл проходится не более одного раза.

3. Аналогично разбираются оставшиеся два случая. В этом случае цикл также проходит не более одного раза, а вычисляемая программой функция представима примитивной формулой.

Рассмотрим арифметические программы над полным базисом, арифметические операторы которого суть $\mathbf{0}(\boldsymbol{x})$ – обнуление переменной $\boldsymbol{x}$, $\boldsymbol{s}(\boldsymbol{x})$ – прибавление единицы, и единственный логический $\boldsymbol{E}(\boldsymbol{x}, \boldsymbol{y})$ – равенство переменных. В [5] показано, что всякая функция арифметики Пресбургера вычисляется такой арифметической программой над полным базисом без вложенных циклов. Отсюда следует, что если заменить в программе без вложенных циклов каждый цикл на функцию, представляющую собой прибавление константы, то в таком расширенном базисе программа представляет собой ор-дерево, в котором уже три типа арифметических вершин, а каждая бинарная вершина - логическая. Назовем такие программы *расширенными*.

Справедливо такое утверждение.

***Теорема* 2.4**. *Функция, вычисляемая расширенной программой, представима примитивной формулой.*

В результате получаем

Следствие. *Всякая функция арифметики Пресбургера представима примитивной формулой.*

Литература

1. Попов С.В., Брошкова Н.Л. Длина вычисления арифметических программ. LAP Lambert Academic Publishing, Saarbrucken, Deutschland. 2014. - 250 с.

2. Верещагин Н. К., Шень А. В. Лекции по математической логике и теории алгоритмов. М.: МЦНМО, 2018. - 128 с.

3. Джон Хопкрофт, Раджив Мотвани, Джеффри Ульман. Введение в теорию автоматов, языков и вычислений. М.: Вильямс, 2015. — 528 с.

4. Яблонский С.В. Введение в дискретную математику. М.: Высшая школа, 2016. -385 с.

5. Янов Ю.И. О вычислениях в одном классе программ // Проблемы кибернетики. – 1977. №32.– С. 200-230.