Записки научных
семинаров ПОМИ
Том 304, 2003 г.

А. П. Бельтюков

СЛАБАЯ КОНСТРУКТИВНАЯ АРИФМЕТИКА ВТОРОГО ПОРЯДКА С ИЗВЛЕЧЕНИЕМ АЛГОРИТМОВ ПОЛИНОМИАЛЬНОЙ ВРЕМЕННОЙ СЛОЖНОСТИ

1. Основные понятия

В настоящей работе рассматриваются теории натуральных чисел с функциональными переменными. Причем, функции, являющиеся значениями функциональных переменных, должны удовлетворять следующему условию *полиномиальной ограниченности*:

- каждая такая функция вычислима за время, ограниченное некоторым полиномом от ее аргументов,
- каждое значение такой функции ограничено некоторым полиномом от соответствующих значений аргументов.

Считается, что языки теорий содержат функциональные константы для сложения и умножения натуральных чисел и предикатную константу =. Допустимо и наличие других функциональных и предикатных констант. Требуется, чтобы функции, обозначаемые этими функциональными константами, удовлетворяли условию полиномиальной ограниченности (для упомянутых констант сложения и умножения выполнение этого условия очевидно).

Введем следующие сокращения:

- если t_1 и t_2 – предметные термы, то $t_1 \le t_2$ – сокращение для формулы
$$\exists x(t_1 + x = t_2).$$
- если $(\mathbf{x}) = (x_1, \dots, x_n)$, t – терм, то $\mathbf{x} \le t$ – сокращение для формулы $x_1 \le t \& \dots \& x_n \le t$, формула $\mathbf{x} \not\le t$ – отрицание формулы $\mathbf{x} \le t$.

Это исследование было поддержано грантом “Weak arithmetics” INTAS 2001–447.

- если ρ – знак бинарного отношения, то запись $\forall x \rho t A$ – сокращение для формулы $\forall x(x \rho t => A)$,
- если ρ – знак бинарного отношения, то запись $\exists x \rho t A$ – сокращение для формулы $\exists x(x \rho t \& A)$,
- если f – функция, а t – терм, то $f \leq t$ означает, что

$$\forall \mathbf{x} \nleq t(f(\mathbf{x}) = 0) \& \forall \mathbf{x} \leq t(f(\mathbf{x}) \leq t).$$

Известно, что если в теориях типа интуиционистской арифметики не накладывать никаких ограничений на схему математической индукции, то реализации строящихся формул (в смысле С. К. Клини) могут оказаться содержащими алгоритмы, очень сложные для вычисления. В частности, это означает, что, доказав формулу $\forall x \exists y A(x, y)$, не стоит надеяться на наличие алгоритма, вычисляющего требуемое y по данному x за время, ограниченное, например, некоторым многочленом от x. Проблема состоит в том, что если индукцией доказывается формула, имеющая достаточно большую кванторную сложность, то вычислительная сложность алгоритмов, участвующих в реализации доказываемой формулы, может существенно возрастать с каждым шагом индукции и в результате стать далеко не полиномиальной. Некоторые ответы на вопросы о том, как можно ограничить сложность формулы, доказываемой по индукции, даются в работах [1] и [2]. Однако в эти работы не относятся к случаю теорий второго порядка, который рассматривается здесь.

В связи с этим зададимся вопросом: при каких самых сильных схемах ограниченной математической индукции в теориях указанного семейства каждой доказанной формуле можно поставить в соответствие реализацию, вычислимую за полиномиальное (от числовых аргументов) время? Например, при каких схемах индукции доказательство формулы

$$\forall x \forall f \exists y A(x, f, y)$$

(где x, y – предметные переменные, f – функциональная переменная) означает, что y можно вычислить по x и f за время, ограниченное некоторым полиномом от x?

2. Описание схемы индукции

Для получения ответа на поставленный вопрос рассмотрим следующий общий вид схемы математической индукции:

$$B(0) \& \forall x (B(x) => B(x+1)) => \forall x B(x).$$

Далее этот вид схемы будем обозначать Ind.

Для того, чтобы описать ограничения, накладываемые на формулу B, определим некоторые вспомогательные языки. Далее через $\mathcal{L}$ обозначаем полный язык теории. Под $\mathcal{D}$ будем иметь в виду так называемый *дескриптивный* язык, определяемый следующими правилами.

- Атомарные формулы $\mathcal{L}$ принадлежат языку $\mathcal{D}$.
- Если L – формула языка $\mathcal{L}$, то формула $\neg L$, принадлежат языку $\mathcal{D}$.
- Если D – формула языка $\mathcal{D}$, а L – формула языка $\mathcal{L}$, то формула $L => D$ принадлежит языку $\mathcal{D}$.
- Если D_1 и D_2 – формулы языка $\mathcal{D}$, то формула $D_1 \& D_2$ принадлежит языку $\mathcal{D}$.
- Если D – формула языка $\mathcal{D}$, а x – предметная переменная, то формула $\forall x D$ принадлежит языку $\mathcal{D}$.
- Других формул в языке $\mathcal{D}$ нет.

Заметим, что реализации формул языка $\mathcal{D}$ тривиальны.

Под $\mathcal{F}$ будем иметь в виду язык формул, определяемый следующими правилами.

- Все формулы языка $\mathcal{D}$ принадлежат языку $\mathcal{F}$.
- Если формулы F_1 и F_2 принадлежат языку $\mathcal{F}$, то формулы $F_1 => F_2$, $F_1 \& F_2$ и $F_1 \vee F_2$ принадлежат языку $\mathcal{F}$.
- Других формул в языке $\mathcal{F}$ нет.

Заметим, что все реализации формул языка $\mathcal{F}$ нетрудно задать конечными таблицами, размер которых оценивается через число связок в формуле.

Под $\mathcal{C}$ будем иметь в виду язык формул, определяемый следующими правилами.

- Все формулы языка $\mathcal{D}$ принадлежат языку $\mathcal{C}$.
- Если формула F принадлежит языку $\mathcal{F}$, а формула C принадлежит языку $\mathcal{C}$, то формула $F => C$ принадлежит языку $\mathcal{C}$.
- Если формулы C_1 и C_2 принадлежат языку $\mathcal{C}$, то формулы $C_1 \& C_2$ и $C_1 \vee C_2$ принадлежат языку $\mathcal{C}$.
- Если формула C принадлежит языку $\mathcal{C}$, а x – предметная переменная, а t – предметный терм, то формула $\exists x \leq t C$ принадлежит языку $\mathcal{C}$.

- Других формул в языке $\mathcal{C}$ нет.

Под $\mathcal{B}$ будем иметь в виду язык формул, определяемый следующими правилами.

- Все формулы языка $\mathcal{D}$ принадлежат языку $\mathcal{B}$.
- Если формула C принадлежит языку $\mathcal{C}$, а формула B принадлежит языку $\mathcal{B}$, то формула $C => B$ принадлежит языку $\mathcal{B}$.
- Если формулы B_1 и B_2 принадлежат языку $\mathcal{B}$, то формулы $B_1 \& B_2$ и $B_1 \vee B_2$ принадлежат языку $\mathcal{C}$.
- Если формула B принадлежит языку $\mathcal{B}$, x – предметная переменная, f – функциональная переменная, а t – предметный терм, то формулы $\exists x \leq tB$, $\forall x \leq tB$ и $\exists f \leq tB$ принадлежат языку $\mathcal{B}$.
- Других формул в языке $\mathcal{B}$ нет.

Далее ограничением для формулы B в схеме индукции Инд будем считать принадлежность этой формулы языку $\mathcal{B}$. За основу строящихся теорий возьмем конструктивное (интуиционистское) исчисление предикатов с равенством. Потребуем, чтобы каждая предметная аксиома этих теорий имела один из следующих видов:

- $\forall \mathbf{x} \exists 1 y A(\mathbf{x}, y) => \exists f \forall \mathbf{x} A(\mathbf{x}, f(\mathbf{x}))$, где $\exists 1 y B(y)$ – традиционное сокращение для

$$\exists y(B(y) \& \forall z(B(z) => y = z))$$

(«квантор существования и единственности»),
- $\forall x(x = 0 \vee \exists y(y + 1 = x))$,
- некоторое предложение дескриптивного языка $\mathcal{D}$.

Семейство получаемых таким образом теорий будем обозначать через $\mathcal{T}$.

3. Теорема

Сформулируем теорему о сложности вычисления объектов, существование которых доказывается в теориях построенного семейства.

Теорема. *По любому доказательству в любой теории семейства $\mathcal{T}$ формулы*

$$\forall \mathbf{z} \forall \mathbf{f} \exists y A(\mathbf{z}, \mathbf{f}, x)$$

можно построить такой алгоритм $g_{\mathbf{f}}$ *с оракулами* $\mathbf{f}$, *время работы которого ограничено полиномом от значений своих аргументов, что*

$$\forall \mathbf{z} \forall \mathbf{f} A(\mathbf{z}, \mathbf{f}, g_{\mathbf{f}}(\mathbf{z})).$$

Теорема доказывается методом, изложенным в статье [2]. В соответствии с этим методом нетрудно сформулировать понятие *полиномиально ограниченной реализации* для теорий описанного семейства. Основная задача при доказательстве этой теоремы состоит в том, чтобы показать, что любая аксиома, полученная по ограниченной схеме аксиом индукции, полиномиально ограниченно реализуема. В соответствии с методом доказательства аналогичной теоремы из статьи [2] для этого следует научиться преобразовывать реализацию доказываемой по индукции формулы B из алгоритмического в табличный вид и обратно с полиномиальной сложностью. Приведенные нами ограничения на формулу B позволяют это сделать без труда: размеры строящихся таблиц определяются объемами областей определения функций в реализациях, а эти объемы полиномиально ограничены вследствие ограничений, накладываемых на кванторы общности и посылки импликаций внутри формулы B.

Полученный результат дает весьма сильный вариант аксиомы ограниченной математической индукции, который, тем не менее, сохраняет существенное ограничение сложности вычисления реализаций доказываемых формул.

Литература

1. S. R. Buss, *Bounded arithmetic.* — Studies in Proof Theory, Bibliopolis, Naples (1986).
2. A. P. Beltiukov, *Intuitionistic formal theories with realizability in subrecursive classes. — Annals of Pure and Applied Logic*, **89** (1997), 3–15.

Beltiukov A. P. Weak constructive second order arithmetic with extracting polynomial time computable algorithms.

A family of weak constructive theories is built in this work. The theories contain arithmetic and a theory of natural valued functions with natural arguments. These functions are polynomially bounded and are computable in a time polynomially bounded in values of their arguments. Theory languages contain functional constants for addition, and multiplication and equality predicate. Other functional constants also may

be used if their functions satisfy the polynomial boundedness conditions above. Polynomial time computable (in numeric values of the arguments) witness functions for proved formulas can be algorithmically extracted from the proofs of these theories. If one of the arguments of witness is a function, then this function is used in the witness algorithm as an oracle.

Удмуртский университет
г. Ижевск
e-mail: belt@uni.udm.ru

Поступило 30 ноября 2002 г.