Ученые записки

Теория алгоритмов и программ

I

Министерство высшего и среднего специального образования
Латвийской ССР
Латвийский ордена Трудового Красного Знамени
государственный университет имени Петра Стучки

Вычислительный центр

Ученые записки
Латвийского государственного университета
имени Петра Стучки
том 210

ТЕОРИЯ АЛГОРИТМОВ И ПРОГРАММ

Выпуск I

Под ред. Я.М.Барздиня

Редакционно-издательский отдел ЛГУ им. Петра Стучки
Рига 1974

УДК 51.01 : 518.5

Настоящий выпуск сборника "Теория алгоритмов и программ, I" посвящен теории индуктивного вывода. Часть статей посвящена изучению математического аппарата индуктивного вывода - предельно вычислимым функциям и функционалам. Рассматриваются также вопросы проверки правильности гипотез (в более практическом плане), связанные с отладкой программ.

По мере накопления материала по данной теме предполагается издание следующих сборников.

Сборник рассчитан на научных работников, занимающихся или интересующихся теорией алгоритмов и программ, аспирантов и студентов.

Т $\frac{0-2-2-4-088у}{М 812(11)-74}$ 299-74

ПРЕДИСЛОВИЕ

В настоящем сборнике делается попытка в рамках общей теории алгоритмов исследовать индуктивный вывод (inductive inference) автоматов, общерекурсивных функций и программ. Под этим понимается восстановление (вывод) самих объектов (автоматов, функций, программ) по результатам их работы на отдельных примерах. При этом, следуя Голду (Gold E.M. Limiting recursion. - "Journal of Symbolic Logic", 1965, 3, No.1), Фелдману (Feldman J.A. Some decidability results on grammatical inferemce and complexity. - "Information and Control", 1972, 20, No.3), Блюму (Blum L., Blum M. Inductive inference: a recursion theoretic approach. - Memorandum No. ERL-M386, March 1973, Univ.of California) индуктивный вывод рассматривается как предельный процесс. Это означает следующее. Сначала выдается одна гипотеза κ_1, потом она проверяется на дальнейших примерах, обнаруживается ошибка, выдается следующая гипотеза κ_2, затем она снова проверяется, возможно, обнаруживается ошибка, выдается гипотеза κ_3 и т.д. Мы говорим, что этот процесс в пределе дает искомый результат, если, начиная с некоторого i, гипотезы κ_i истинны (хотя мы сами этого можем и не знать и продолжать проверку). По такой схеме происходит, например, отладка программ. Число вариантов программ $\kappa_1, \kappa_2, \kappa_3, \ldots$, которые программист перебирает, иногда бывает довольно большим. По такой же схеме осуществляются и различные процессы обучения.

Легко видеть, что для любого эффективно перечислимого класса общерекурсивных функций существует алгоритм индуктивного вывода (стратегия), который для любой функции этого класса по таблице ее значений выдает в пределе геделевский номер этой функции. Таким, например, является алгоритм, который, перебирая все функции по порядку, ищет функцию, которая не противоречит заданной таблице значений. Наиболее важной характеристикой алгоритма индуктивного вывода является число гипотез, которое он перебирает, пока доходит до искомого результата. Упомянутый алгоритм на функции с номером n перебирает, вообще говоря, до n гипотез. Однако, оказывается (Barzdin J.M. Prognostication of automata and functions. - IFIP Congress 71, см.также Барздинь Я.М., Фрейвалд Р.В. Прогнозирование и предельный синтез эффективно перечислимых классов функций. - Наст.сборник), что существует алгоритм, который на функции с номером n перебирает по порядку не более, чем $\log_2 n$ гипотез. А это число очень небольшое по сравнению с n. Этот относительно простой, но несколько неожиданный результат послужил стимулом для проведения ряда дальнейших исследований, составляющих содержание настоящего сборника.

Формальным аппаратом для постановки ряда задач индуктивного вывода служат предельно вычислимые функции и функционалы. Исследованию этого аппарата посвящены статьи "Предельно вычислимые функции и функционалы", "Об одном свойстве предельно вычислимых функционалов", "О предельных вычислениях на недетерминированных машинах Тьюринга", "Предельные вычисления на вероятностных машинах", "О частотных вычислениях на бесконечных выборках".

Статьи:"Сравнение различных типов предельного синтеза и прогнозирования", "Две теоремы о предельном синтезе функций", "Равномерная и неравномерная прогнозируемость", "Предельный синтез τ -номеров", "Об ускорении синтеза и прогнозирования функций", "Прогнозирование и предельный синтез конечных автоматов", "Замечание о синтезе программ по историям их работы", "О предельном синтезе почти минимальных геделевских номеров"–непосредственно относятся к индуктивному выводу функций, автоматов и программ.

Особое место занимают статьи "Построение полной системы примеров для проверки корректности программ" и "Разрешимые и неразрешимые случаи проблемы построения полной системы примеров". Они посвящены проверке правильности гипотез, но в более практическом плане, связанном с отладкой программ.

ПРЕДЕЛЬНО ВЫЧИСЛИМЫЕ ФУНКЦИИ И ФУНКЦИОНАЛЫ

Р.В.Фрейвалд

Математическим аппаратом для формализации ряда задач индуктивного вывода служат понятия предельно вычислимых функций и функционалов. Эти понятия впервые систематически исследовались Э.М.Голдом [1] и Х.Патнамом [2], однако первые результаты, касающиеся предельно вычислимых функционалов, восходят к Мазуру, Крейселу и Пур-Эль (см. [3]).

Настоящая статья содержит изложение основных элементарных фактов о предельно вычислимых функциях и функционалах. Целью написания статьи явилось не столько изложение новых математических результатов, сколько стремление иметь в сборнике обоснование выбора основной концепции предельных вычислений, рассмотрение различных естественных обобщений этой концепции, выяснение эквивалентности или неэквивалентности их.

Интуитивное понятие предельных вычислений является достаточно ясным. Типичным случаем, когда требуемый результат получается не одним приемом, а лишь предельно, является процесс программирования. Как правило, программисту не удается составить длинную программу без ошибок. Поэтому необходим процесс отладки. Ошибки, содержащиеся в программе, постепенно устраняются, и некоторое время спустя программа работает правильно. Однако абсолютной уверенности в том, что все ошибки уже обнаружены, у программиста обычно не бывает.

Эта особенность процесса программирования положена в основу интуитивного понятия предельно вычислимой функции. Алгоритм предельного вычисления функции вместо окончательной выдачи результата показывает предполагаемый результат на специальном "табло" или "регистре" (на котором может быть записано любое натуральное число). Это позволяет алгоритму предложить в качестве результата одно значение k_1, через некоторое время убрать это значение, еще через некоторое время предложить другое значение k_2 и т.д. Если при работе алгоритма на некотором значении аргумента x после появления на табло некоторого значения k_n оно уже с табло не убирается и не заменяется, говорят, что алгоритм предельно вычисляет значение функции $f(x)=k_n$; в противном случае $f(x)$ не определена.

Аналогично определяются и функционалы, предельно вычислимые в интуитивном смысле. В этом случае вместо записи числа x в распоряжении алгоритма предельного вычисления имеется написанный на бесконечной ленте в произвольном порядке, возможно с пробелами, график функции, являющейся аргументом функционала.

Эти интуитивные понятия предельных вычислений могут быть уточнены. Для этого в качестве алгоритма предельного вычисления надо брать машины типа Тьюринга со специальной выходной лентой, на которой она печатает гипотезы $k_1, k_2, \ldots$

Формально это - машины Тьюринга с входной, рабочей и выходной лентой. На выходной ленте записано значение аргумента x (в случае вычисления функций) или график функции φ в произвольном порядке и с произвольными пробелами (в случае вычисления функционалов). В начале работы рабочая и выходная ленты пусты. Головка на выходной ленте является только-пишущей. Она может стоять на месте или двигаться вправо, но не может двигаться влево. Во время работы эта головка печатает последовательность натуральных чисел $k_1, k_2, \ldots$ с разделительными знаками между их записями. Будем считать запись очередного числа законченной, если за этой записью напечатан хотя бы один разделительный знак и головка уже ушла вправо от этого разделительного знака. Между двумя соседними числами допускается произвольное число разделительных знаков. Тогда любое заполнение машиной выходной ленты или ее части определяет пустую, конечную или бесконечную последовательность тех натуральных чисел, печать которых уже закончена. Таким образом, числа, за которыми не следует разделительный знак, или, например, бесконечная последовательность единиц (если такие имеются на ленте) во внимание не принимаются. Если последовательность $k_1, k_2, \ldots$ конечная, то ее пределом назовем последнее число последовательности. Если последовательность бесконечная, но все члены, начиная с некоторого, равны некоторому числу b, то пределом последовательности называется число b. Во всех остальных случаях, в том числе, когда последовательность пустая, будем говорить, что последовательность предела не имеет.

Под функцией, которую предельно вычисляет данная машина $\mathfrak{M}$, будем понимать функцию, которая каждому x сопоставляет предел последовательности $k_1, k_2, \ldots$, напечатанной машиной $\mathfrak{M}$ при работе на аргументе x.

Под функционалом, который предельно вычисляет данная машина $\mathfrak{M}$, будем понимать функционал, который каждой φ сопоставляет предел последовательности $k_1, k_2, \ldots$, напечатанной машиной $\mathfrak{M}$ при работе на аргументе φ. (Как обычно при определениях функционала, подразумевается, что результат, т.е. предел последовательности $k_1, k_2, \ldots$ не зависит от расположения графика φ на входной ленте).

Функцию (функционал) будем называть предельно вычислимой (-ым) в формальном смысле, если существует машина Тьюринга, которая ее (его) предельно вычисляет.

В дальнейшем мы будем исходить из тезиса (аналогичного тезису Черча), что функции и функционалы предельно вычислимы в интуитивном смысле тогда и только тогда, когда они предельно вычислимы в формальном смысле. Поэтому ниже, говоря о предельно вычислимых функциях и функционалах, оговорку "в формальном смысле" мы будем опускать.

ПРЕДЛОЖЕНИЕ I. Функция $f(x)$ является предельно вычислимой тогда и только тогда, когда она представима в виде предела $f(x)=\lim\limits_n \tilde{f}(x,n)$ общерекурсивной функции $\tilde{f}(x,n)$

ДОКАЗАТЕЛЬСТВО. ДОСТАТОЧНОСТЬ очевидна.

НЕОБХОДИМОСТЬ. Пусть $f(x)$ предельно вычислима. Построим требуемую о.р. функцию $\tilde{f}(x,n)$. Как правило, значение $\tilde{f}(x,n)$ равно тому κ_i, печатание которого машиной Тьюринга было закончено последним за первые n тактов работы машины на x. Однако не исключено, что за всю бесконечно долгую работу на каком-то x машина не печатает полностью ни одного значения. В этом случае $\lim\limits_n \tilde{f}(x,n)$ не должен существовать. Поэтому до завершения печатания машиной Тьюринга значения κ_1 полагаем $\tilde{f}(x,0)=0$, $\tilde{f}(x,1)=1$, $\tilde{f}(x,2)=0$, $\tilde{f}(x,3)=1$ и т.д.

(Этот прием "непрерывных колебаний" оказывается полезным и при переходе от предельного вычисления в интуитивном смысле к предельному вычислению в формальном смысле. Ведь с "табло" результаты можно было и убирать, а в определении предельного вычисления на машинах Тьюринга такая возможность не предусмотрена).

ПРЕДЛОЖЕНИЕ 2. Функционал $F(\varphi)$ над произвольным классом $\mathfrak{F}$ всюду определенных функций является предельно вычислимым тогда и только тогда, когда он представим в виде $F(\varphi)=\lim\limits_n \tilde{F}(\varphi,n)$, где $\tilde{F}(\varphi,n)$ - общерекурсивный функционал над $\mathfrak{F}$.

ДОКАЗАТЕЛЬСТВО. ДОСТАТОЧНОСТЬ очевидна.

НЕОБХОДИМОСТЬ. Пусть функционал $F(\varphi)$ предельно вычислим машиной $\mathfrak{M}$. В доказательстве предложения I показано, как перестроить $\mathfrak{M}$ в машину $\mathfrak{M}'$, печатающую бесконечную выходную последовательность $\kappa_1, \kappa_2, \ldots$ при работе на каждой φ. Легко далее перестроить $\mathfrak{M}'$ в машину $\mathfrak{M}''$, выходная последовательность которой не зависит от порядка, в каком записан на входной ленте график φ. Машина $\mathfrak{M}''$ фактически моделирует работу $\mathfrak{M}'$ при условии, что график φ расположен в естественном порядке $<0,\varphi(0)>, <1,\varphi(1)>, <2,\varphi(2)>, \ldots$. Тогда $\tilde{F}(\varphi,n)$ определяется как n - е выходное значение при работе $\mathfrak{M}''$ на φ.

Оказывается, что аналог предложения 2 для функционалов над классами частично определенных функций не имеет места. Более того, справедливо

ПРЕДЛОЖЕНИЕ 3. Пусть класс $\mathcal{T}$ частичных функций содержит такую пару различных функций φ_1, φ_2, что график функции φ_1 является подмножеством графика функции φ_2. Тогда существует такой предельно вычислимый всюду определенный функционал $F(\varphi)$ над классом $\mathcal{T}$, что для любого k и любого общерекурсивного функционала $\tilde{F}(\varphi,i_1,i_2,\ldots,i_k)$ над $\mathcal{T}\times N^k$ существует $\varphi \in \mathcal{T}$, что $F(\varphi) \neq \lim\limits_{i_1} \lim\limits_{i_2} \ldots \lim\limits_{i_k} \tilde{F}(\varphi,i_1,i_2,\ldots,i_k)$.

ДОКАЗАТЕЛЬСТВО. Пусть график функции φ_2 содержит пару $<x_0,y_0>$, которая не принадлежит графику функции φ_1. Рассмотрим функционал $F(\varphi)$, равный 0, если график φ содержит пару $<x_0,y_0>$ и равный 1 в противном случае. Этот функционал предельно вычислим: машина просматривает всю входную ленту, до обнаружения $<x_0,y_0>$ выдавая значение 1 и после обнаружения такой пары - значение 0. Итак, F определен на φ_1, и $F(\varphi_2) \neq F(\varphi_1)$. Заметим, что, каковы бы ни были натуральные $k,\alpha_1,\ldots,\alpha_k$ и каков бы ни был общерекурсивный функционал $\tilde{F}(\varphi,i_1,i_2,\ldots,i_k)$ над $\mathcal{T}\times N^k$, имеет место $\tilde{F}(\varphi_1,\alpha_1,\ldots,\alpha_k)=\tilde{F}(\varphi_2,\alpha_1,\ldots,\alpha_k)$ (так как: I) значение функционала не должно зависеть от порядка записи графика функции-аргумента и 2) значение $\tilde{F}(\varphi_1,\alpha_1,\ldots,\alpha_k)$ должно быть определено). Но тогда $\lim\limits_{i_1} \ldots \lim\limits_{i_k} \tilde{F}(\varphi_1,i_1,\ldots,i_k)=\lim\limits_{i_1} \ldots \lim\limits_{i_k} \tilde{F}(\varphi_2,i_1,\ldots,i_k)$.

Э.М.Голд [I] доказал такую характеристику предельно вычислимых функций.

ПРЕДЛОЖЕНИЕ 4. Функция является предельно вычислимой тогда и только тогда, когда ее график принадлежит классу Σ_2 иерархии Клини-Мостовского.

СЛЕДСТВИЕ. Функция является предельно вычислимой тогда и только тогда, когда она вычислима на машине Тьюринга с креативным оракулом.

(Определение классов иерархии Клини-Мостовского и определение креативности см.[4]).

Аналоги предложения 4 (но не следствия!) можно доказать и для функционалов над произвольными классами всюду определенных функций. Нижеследующее предложение 5, которое доказывается по образцу предложения 3, однако показывает, что для функционалов над многими классами частичных функций даже аналоги предложения 4 не выполняются.

ПРЕДЛОЖЕНИЕ 5. Пусть класс $\mathcal{T}$ частичных функций содержит такую пару различных функций, что график функции φ_1 является подмножеством графика функции φ_2. Тогда существует такой предельно вычислимый всюду определенный функционал $F(\varphi)$ над классом $\mathcal{T}$, что для любого k и любого общерекурсивного функционала $\bar{F}(\varphi, y, f_1, f_2, \ldots, f_k)$ над $\mathcal{T} \times N \times \mathcal{F}^k$ (где $\mathcal{F}$ - класс всех всюду определенных функций) существуют $\varphi \in \mathcal{T}$ и $y \in N$, что $(F(\varphi) = y) \not\Leftrightarrow \exists f_1 \forall f_2 \exists f_3 \ldots \bar{F}(\varphi, y, f_1, f_2, \ldots, f_k)$.

Понятие предельно вычислимой функции (в формальном смысле) оказывается понятием весьма инвариантным относительно привлечения новых вычислительных возможностей. Так, например, в статьях [5, 6] настоящего сборника показано, что если в качестве алгоритмов предельного вычисления привлекаются недетерминированные или вероятностные машины, но при такой формализации мы не отходим от упомянутого выше интуитивного смысла предельного вычисления, то класс вычислимых функций не расширяется.

С другой стороны, уже сами формализации понятий предельно вычислимых функций и функционалов через машины Тьюринга и через о.р. функции подсказывают некоторые естественные видоизменения этих понятий, не укладывающиеся в первоначальную модель. Остановимся на некоторых из них.

Ряд видоизменений понятия предельно вычислимой функции связан с формализацией понятия предельной вычислимости в терминах машин Тьюринга. Дело в том, что при работе машины на очередном значении аргумента x могут случиться четыре ситуации: I) не будет напечатано ни одно выходное значение; 2) будет напечатана конечная последовательность выходных значений $k_1, k_2, \dots, k_n$; 3) будет напечатана бесконечная последовательность $k_1, k_2, \dots$, имеющая предел; 4) будет напечатана бесконечная последовательность $k_1, k_2, \dots$, не имеющая предела. В первой и четвертой ситуациях единственно естественно полагать, что функция $f(x)$, предельно вычислимая машиной, на x не определена. Тем не менее можно отличать ситуации 2) и 3) и рассматривать определения, когда

А) $f(x)$ определена как в ситуации 2), так и в ситуации 3);

Б) $f(x)$ определена в ситуации 2), но не в ситуации 3);

В) $f(x)$ определена в ситуации 3), но не в ситуации 2).

Легко видеть, что А - это основное определение предельной вычислимости, приведенное выше.

ПРЕДЛОЖЕНИЕ 6. Класс функций, предельно вычислимых в смысле Б, совпадает с классом функций, предельно вычислимых в смысле А.

ДОКАЗАТЕЛЬСТВО. из предложения I следует, что любая функция, вычислимая в смысле А, вычислима и в смысле Б.

Пусть, наоборот, функция вычислима в смысле Б на машине $\mathfrak{M}$. Работу $\mathfrak{M}$ можно легко промоделировать на машине $\mathfrak{M}'$, которая до тех пор, пока $\mathfrak{M}$ не напечатала ни одного символа, печатает попеременно 0 и I на каждом такте, а далее после того, как $\mathfrak{M}$ уже напечатала хотя бы один символ, печатает: в те такты, когда $\mathfrak{M}$ не заканчивает печатание очередного выходного значения, машина $\mathfrak{M}'$ печатает снова и снова то значение, которое $\mathfrak{M}$ напечатала последним; в те такты, когда $\mathfrak{M}$ заканчивает печатание очередного значения k_i, $\mathfrak{M}'$ прерывает на время моделирование работы $\mathfrak{M}$, печатает подряд два различных значения, например, 0 и I, потом печатает k_i и продолжает прерванное моделирование. Легко видеть, что машина $\mathfrak{M}'$ вычисляет ту же самую функцию в смысле В и А (одновременно).

Из доказательства предложения 6 вытекает

СЛЕДСТВИЕ. Любая функция, предельно вычислимая в смысле А или Б, вычислима и в смысле В.

Предложение 6 и следствие имеют место также для функционалов над произвольными классами функций. Однако даже для функций утверждение, обратное следствию предложения 5, не верно. Поэтому В определяет родственное, но несколько более слабое понятие вычислимости. Следуя [5], назовем это слабой предельной вычислимостью.

Для характеризации класса слабо предельно вычислимых функций усилим одно утверждение работы [5].

ПРЕДЛОЖЕНИЕ 7. Функция $\varphi(x)$ слабо предельно вычислима тогда и только тогда, когда одновременно I) ее область определения представима в виде разности двух множеств из Σ_2, 2) она доопределима до (частично определенной) предельно вычислимой функции, т.е. до функции с графиком, принадлежащим Σ_2.

(Отметим, что класс всех множеств, представимых в виде разности двух множеств из Σ_2, включает полностью $\Sigma_2 \cup \Pi_2$ и содержится в $\Sigma_3 \cap \Pi_3$).

ДОКАЗАТЕЛЬСТВО. НЕОБХОДИМОСТЬ. Пусть $\varphi(x)$ слабо предельно вычислима на машине $\mathfrak{M}$. Существует машина $\mathfrak{M}'$, слабо предельно вычисляющая частичную характеристическую функцию области определения функции φ – эта машина для выдачи n - го выходного значения k'_n ожидает n -ое и $(n+1)$- е выходные значения k_n, k_{n+1} машины $\mathfrak{M}$. Если $k_n \neq k_{n+1}$, то $k'_n = 1$. Если $k_n = k_{n+1}$, то $k'_n = 0$. Для доказательства необходимости условия 1) надо убедиться в том, что следующие два множества принадлежат Σ_2: а) множество всех тех x, для которых $\mathfrak{M}'$, начиная с некоторого момента, не печатает на выходе нулей; б) множество всех тех x, для которых $\mathfrak{M}'$ печатает только конечную выходную последовательность. Для доказательства необходимости условия 2) нужно перестроить машину $\mathfrak{M}$ способом, описанным в доказательстве предложения 6.

ДОСТАТОЧНОСТЬ. Пусть область определения функции представима в виде разности $B \setminus C$, где $B, C \in \Sigma_2$, и пусть функция $\varphi(x)$ доопределима до предельно вычислимой функции $\psi(x)$. По предложениям 4 и 1 частичные характеристические функции $\chi_B(x)$ и $\chi_C(x)$ множеств B и C и функция $\psi(x)$ представимы в виде пределов о.р. функций:

$\chi_B(x) = \lim_n h_1(x,n)$; $\chi_C(x) = \lim_n h_2(x,n)$; $\psi(x) = \lim_n h_3(x,n)$.

Тогда $\varphi(x)$ слабо предельно вычислима на машине $\mathfrak{M}$, которая для выдачи n - го выходного значения просматривает $h_2(x,0), h_2(x,1), h_2(x,2), \ldots$, пока среди этих значений найдется n чисел, отличных от единицы. Потом, если $h_1(x,n) = 1$, то $\mathfrak{M}$ печатает $h_3(x,n)$, а если $h_1(x,n) \neq 1$, то $\mathfrak{M}$ печатает число k_n, отличное от k_{n-1}.

СЛЕДСТВИЕ 1. (Фрейвалд, Подниекс [5]). Если область значений функции состоит из одного элемента, то функция слабо предельно вычислима тогда и только тогда, когда ее область определения представима в виде разности двух множеств из Σ_2.

СЛЕДСТВИЕ 2. Если функция всюду определена, то она слабо предельно вычислима тогда и только тогда, когда она предельно вычислима.

Перейдем теперь к формализации понятий предельно вычислимых функций в терминах о.р. функций. Естественно напрашивается мысль о замене требования общерекурсивности функции $f(x,n)$ требованием принадлежности $f(x,n)$ другим классам функций. Э.М.Голд [I] показал, что класс предельно вычислимых функций не изменяется, если потребовать примитивную рекурсивность функции $f(x,n)$. Дело обстоит иначе, если допускать частично рекурсивные (ч.р.) $f(x,n)$. Можно рассматривать два различных определения предела ч.р.функции. Э.М.Голд [I] ввел следующее определение: $\lim\limits_n \varphi(x,n)=b \Leftrightarrow \Leftrightarrow \exists n_0 \forall n \geqslant n_0$ ($\varphi(x,n)$ определена $\& \ \varphi(x,n)=b$). Можно определить предел иначе: $\lim\limits_n \varphi(x,n)=b \Longleftrightarrow$ ($\forall n \ \varphi(x,n)$ определена) $\&(\exists n_0 \forall n \geqslant n_0 \ \varphi(x,n)=b)$. Последнее определение ближе определению предела о.р. функции, так как в случаях, когда предел определен, можно вычислять последовательно $\varphi(x,0), \varphi(x,1), \varphi(x,2)$ и не надо опасаться, что какое-то из значений может оказаться неопределенным.

Э.М.Голд [I] показал, что существуют функции, представимые в виде предела ч.р. двуместной функции (в смысле первого определения), но не представимые в виде предела о.р. функции. Голд поставил вопрос о характеризации класса пределов ч.р. функций в терминах Клини-Мостовского.

ТЕОРЕМА I. Функция $\varphi(x)$ представима в виде предела ч.р. функции (в смысле определения Э.М.Голда) тогда и только тогда, когда одновременно I) ее область определения принадлежит Σ_3, 2) она доопределима до (частично определенной) предельно вычислимой функции, т.е. до функции с графиком, принадлежащим Σ_2.

ДОКАЗАТЕЛЬСТВО. НЕОБХОДИМОСТЬ условия I) по существу была доказана еще Голдом [I]. Действительно, если $\varphi(x)$ представима в виде предела ч.р. функции, то и частичная характеристическая функция области определения функции φ представима в виде такого предела. Но тогда нетрудно и саму область определения φ задать формулой с кванторной приставкой типа $\exists\forall\exists$.

Для доказательства необходимости условия 2) покажем, что для любой функции φ, представимой в виде предела ч.р. функции $\varphi(x)=\lim\limits_n \tilde{\varphi}(x,n)$, существует доопределение до предельно вычислимой (частично определенной) функции, т.е. до функции $\psi(x)=\lim\limits_l g(x,l)$, где g - ор. функции. Для определения последовательности значений $g(x,0),g(x,1),g(x,2),\ldots$ просматриваем в порядке канторовских номеров пар (n,t) результаты вычисления $\tilde{\varphi}(x,n)$ в течение t тактов машины Тьюринга. Пока ни одно из значений $\tilde{\varphi}(x,n)$ еще не оказалось определенным, полагаем $g(x,0)=0, g(x,1)=0, g(x,2)=0,\ldots$ Потом $g(x,l)$ равно значению $\tilde{\varphi}(x,m)$, где m - максимальное среди тех n, для которых существует такое t, что пара (n,t) имеет номер не больше l и вычисление $\tilde{\varphi}(x,n)$ останавливается за t тактов. Очевидно, что если предел $\lim\limits_n \tilde{\varphi}(x,n)$ существует, то $\lim\limits_l g(x,l)=\lim\limits_n \tilde{\varphi}(x,n)$.

ДОСТАТОЧНОСТЬ. Пусть для некоторой функции $\varphi(x)$ выполнены условия I) и 2). По теореме Крейсела, Шёнфильда и Ван Хао (теорема I4-ХУШ из [4]) любое отношение из Σ_3 можно представить в виде $\exists n_0 \forall n>n_0 \exists t\, R(x,n,t)$, где R - рекурсивный предикат. Пусть в таком виде представлено отношение " x принадлежит области определения функции φ ". Определим ч.р. функцию:

$$\tilde{\varkappa}(x,n)=\begin{cases} 1, & \text{если } \exists t\, R(x,n,t) \\ \text{не определена,} & \text{если } \neg\exists t\, R(x,n,t). \end{cases}$$

Тогда $\varkappa(x)=\lim\limits_n \tilde{\varkappa}(x,n)$ является частичной характеристической функцией области определения функции φ. По условию 2) функция $\varphi(x)$ доопределима до предельно вычислимой функции $\psi(x)=\lim_n g(x,n)$. Но тогда функция $\varphi(x)$ представима в виде предела $\lim_n (\chi(x,n)\cdot g(x,n))$.

СЛЕДСТВИЕ 1. Если область значений функции состоит из одного элемента, то функция представима в виде предела ч.р. функции (в смысле определения Голда) тогда и только тогда, когда ее область определения принадлежит Σ_3.

СЛЕДСТВИЕ 2. Если функция всюду определена, то она представима в виде предела ч.р. функции (в смысле определения Голда) тогда и только тогда, когда она представима в виде предела о.р. функции.

ТЕОРЕМА 2. Функция $\varphi(x)$ представима в виде предела ч.р. функции (в смысле второго определения предела ч.р. функции) тогда и только тогда, когда она слабо предельно вычислима.

ДОКАЗАТЕЛЬСТВО. НЕОБХОДИМОСТЬ. Пусть $\varphi(x)=\lim_n \tilde{\varphi}(x,n)$. Для слабого предельного вычисления φ достаточно последовательно вычислять и печатать значения $\tilde{\varphi}(x,0), \tilde{\varphi}(x,1), \tilde{\varphi}(x,2),\ldots$

ДОСТАТОЧНОСТЬ. Пусть $\varphi(x)$ слабо предельно вычислима на машине $\mathfrak{M}$. Тогда $\varphi(x)=\lim_n \tilde{\varphi}(x,n)$, где $\tilde{\varphi}(x,n)$ равно n - му выходному значению машины $\mathfrak{M}$ при работе на x.

СЛЕДСТВИЕ. Функция $\varphi(x)$ представима в виде предела ч.р. функции (в смысле второго определения предела ч.р. функции) тогда и только тогда, когда одновременно 1) ее область определения представима в виде разности двух множеств из Σ_2, 2) она доопределима до (частично определенной) предельно вычислимой функции, т.е. до функции с графиком, принадлежащим Σ_2.

Итак, после рассмотрения различных видоизменений понятия предельной вычислимости функций остались только две нестандартные концепции. Они не укладываются в нашу "физическую" модель, и, более того, не удалось найти даже удовлетворительную со всех точек зрения содержательную интерпретацию этих концепций. Тем не менее, например, слабая предельная вычислимость любопытна хотя бы тем, что при минимальном отличии от обычной предельной вычислимости, если используются детерминированные машины, она приобретает значительно большие возможности при использовании недетерминированных или вероятностных машин (см. [5, 6]).

Для функционалов совокупность неэквивалентных концепций несколько богаче.

ПРЕДЛОЖЕНИЕ 8. Функционал $F(\varphi)$ над произвольным классом $\mathfrak{T}$ частичных функций представим в виде предела (в смысле определения Голда) $F(\varphi)=\lim\limits_n \tilde{F}(\varphi,n)$, где $\tilde{F}(\varphi,n)$ - вычислимый (но не обязательно всюду определенный) функционал над $\mathfrak{T}\times N$ тогда и только тогда, когда существуют функционалы $G_1(\varphi,i,j,k)$ и $G_2(\varphi,y,i,j)$ общерекурсивные над $\mathfrak{T}\times N^3$, что одновременно 1) $F(\varphi)$ определен $\Leftrightarrow \exists i\,\forall j\,\exists k\,[G_1(\varphi,i,j,k)=1]$, 2) если $F(\varphi)$ определен и $F(\varphi)=y$, то $\exists i\,\forall j\,[G_2(\varphi,y,i,j)=1]$.

ПРЕДЛОЖЕНИЕ 9. Функционал $F(\varphi)$ над произвольным классом $\mathfrak{T}$ частичных функций представим в виде предела (в смысле второго определения предела) $F(\varphi)=\lim\limits_n \tilde{\tilde{F}}(\varphi,n)$, где $\tilde{\tilde{F}}(\varphi,n)$ - вычислимый (но не обязательно всюду определенный) функционал над $\mathfrak{T}\times N$ тогда и только тогда, когда существуют такие функционалы $G_1(\varphi,i,j)$, $G_2(\varphi,i,j)$ и $G_3(\varphi,y,i,j)$, общерекурсивные над $\mathfrak{T}\times N^2$ и над $\mathfrak{T}\times N^3$ (соответственно), что одновременно 1) $F(\varphi)$ определен $\Leftrightarrow \exists i\,\forall j\,[G_1(\varphi,i,j)=1]\ \&\ \forall i\,\exists j\,[G_2(\varphi,i,j)=1]$, 2) если $F(\varphi)$ определен и $F(\varphi)=y$, то $\exists i\,\forall j\,[G_3(\varphi,y,i,j)=1]$.

ПРЕДЛОЖЕНИЕ 10. Функционал $F(\varphi)$ над произвольным классом $\mathfrak{F}$ всюду определенных функций слабо предельно вычислим тогда и только тогда, когда $F(\varphi)$ представим в виде предела (в смысле второго определения предела) функционала $\widetilde{F}(\varphi, n)$, вычислимого, но не обязательно всюду определенного над $\mathcal{F} \times N$.

Для функционалов над многими классами частичных функций аналог предложения IO не имеет место, так как в предложении 3 без изменения доказательства функционал можно полагать не всюду определенным.

В заключение автор благодарит Я.М.Барздиня за неоднократную полезную конструктивную критику.

ЛИТЕРАТУРА

I. Gold E.M. Limiting recursion.- "Journal of Symbolic Logic", 1965, No.1.
2. Putnam M. Trial and error predicates and the solution to a problem of Mostowski.- "Journal of Symbolic Logic", 1965, 30, No.1.
3. Pour-El M.B. A comparison of five Computable operators.- "Z.math.Logic und Grundl.Math.", 1960, No.6.
4. Роджерс Х. Теория рекурсивных функций и эффективная вычислимость, М., I972.
5. Фрейвалд Р.В., Подниекс К.М. О предельных вычислениях на недетерминированных машинах Тьюринга.- Настоящий сборник, стр.25-3I.
6. Фрейвалд Р.В. Предельные вычисления на вероятностных машинах.- Настоящий сборник, стр.32-47.

ОБ ОДНОМ СВОЙСТВЕ ПРЕДЕЛЬНО ВЫЧИСЛИМЫХ ФУНКЦИОНАЛОВ

Я.М.Барздинь

Через R (соответственно, PR) обозначим класс всех I-местных общерекурсивных (частично рекурсивных) функций. Фиксируем некоторую геделевскую нумерацию φ класса PR.

Цель настоящей статьи - изучить связь между двумя классами функционалов:

I. Предельно вычислимые функционалы на R (определение см. в [I]).

2. Предельно вычислимые операции на R. Функционал Ψ называется предельно вычислимой операцией на R , если существует предельно вычислимая функция ψ (определение см. в [I]) такая, что для любого n , если $\varphi_n \in R$, то $\Psi(\varphi_n) = \psi(n)$

Для обычных функционалов имеет место теорема Крейсела-Лакомба-Шёнфильда (см., например, [2]): класс всюду определенных на R эффективных операций совпадает с классом всюду определенных рекурсивных функционалов на R . Однако для предельно вычислимых функционалов имеет место

ТЕОРЕМА I. Существует всюду определенная предельно вычислимая операция на R , которая не является предельно вычислимым функционалом на R .

ДОКАЗАТЕЛЬСТВО. Функции будем отождествлять с последовательностями натуральных чисел. Например, $\alpha 0^{\infty}$, где $\alpha = \varepsilon_1 \varepsilon_2 \dots \varepsilon_n$, $\varepsilon_i \in N$, обозначает следующую функцию $f(x)$: $f(1) = \varepsilon_1, f(2) = \varepsilon_2, \dots f(n) = \varepsilon_n, f(n+1) = 0, f(n+2) = 0, \dots$ (мы считаем, что аргумент x принимает значения, начиная с $x = 1$). Через f^n будем обозначать начальный кусок $f(1) \dots f(n)$ функции f.

Очевидно, для доказательства теоремы достаточно рассматривать какую-то одну геделевскую нумерацию. Для облегчения техники доказательства мы возьмем геделевскую нумерацию, обладающую некоторыми свойствами "оптимальности".

А именно, геделевскую нумерацию φ назовем оптимальной, если для любой вычислимой нумерации π существует общерекурсивная функция $h(n)$ и такая константа C, что для любого n $\varphi_{h(n)} = \pi_n$ и $l(h(n)) \leq l(n) + C$ (через $l(p)$ мы обозначим длину двоичной записи числа p). Такие нумерации фактически были введены А.Н.Колмогоровым [3] при определении сложности конструктивных объектов. Следуя этой идее, сложность $K(f)$ функции $f \in PR$ и сложность $K(\alpha)$ последовательности $\alpha = \varepsilon_1 \ldots \varepsilon_n$ определим соответственно

$$K(f) = \min_{p:\, \forall x\, \varphi(p,x) = f(x)} l(p), \quad K(\alpha) = \min_{p:\, \varphi(p,i) = \varepsilon_i,\, i=1,\ldots,n} l(p).$$

Очевидно, для любой другой вычислимой нумерации π существует константа C такая, что $K(f) \leq K_{\pi}(f) + C$ и $K(\alpha) \leq K_{\pi}(\alpha) + C$.

Фиксируем последовательность натуральных чисел $n_1, n_2, \ldots, n_i, \ldots$, где $n_{i+1} = n_i^3$. Пусть n_1 достаточно большое. Тогда для любого $i > 1$ имеют место следующие утверждения:

А. Какую бы двоичную (т.е. составленную из 0 и I) последовательность α длины n_i мы ни взяли, найдется двоичная последовательность β длины $n_{i+1} - n_i$ такая, что $K(\alpha\beta) > \sqrt{n_{i+1}}$.

В. Какую бы двоичную последовательность α длины n_i мы ни взяли, для любого $s \geq (n_{i+1} - n_i)$ $K(\alpha 0^s) < \sqrt{n_{i+1}}$.

Для проверки справедливости утверждения А заметим, что существует двоичное слово β длины $n_{i+1} - n_i$ такое, что $K(\beta) \geq n_{i+1} - n_i$. Далее, из оптимальности функции φ следует, что $K(\beta) \leq K(\alpha\beta) + 2l(n_i) + c'$. Отсюда $K(\alpha\beta) \geq n_{i+1} - n_i - 2l(n_i) - c'$ и, следовательно, при достаточно большом n_1 $K(\alpha\beta) > \sqrt{n_{i+1}}$.

Аналогично, используя оптимальность нумерации φ, можно убедиться в справедливости утверждения В.

В дальнейшем будем считать, что n_i выбрано столь большим, чтобы выполнялись утверждения А и В. Через $S(f)$ обозначим число различных i, при которых $K(f^{n_i}) > \sqrt{n_i}$. Через $mod_2 S$ обозначим остаток при делении числа S на два.

Искомый функционал Ψ определим следующим образом: $\Psi(f) = mod_2 S(f)$. Очевидно, Ψ всюду определен на R.

Из утверждений А и В непосредственно вытекает:

С. Какую бы двоичную последовательность α длины n_i мы ни взяли, найдутся двоичные последовательности β и γ длины $n_{i+1} - n_i$ такие, что для функций $f_1 = \alpha\beta 0^\infty$ и $f_2 = \alpha\gamma 0^\infty$ имеет место $mod_2 S(f_1) \neq mod_2 S(f_2)$, и, следовательно, $\Psi(f_1) \neq \Psi(f_2)$.

Нетрудно убедиться, что Ψ является предельно вычислимой операцией на R. На самом деле, если дан произвольный геделевский номер z общерекурсивной функции f (без ограничения общности будем считать, что $z \geq 64$, и, следовательно, $l(z) < \sqrt{n}$), то мы можем гарантировать следующее: для любого $n \geq z$ имеет место $K(f^n) \leq l(z) \leq \sqrt{n}$. Это означает, что по z мы можем эффективно найти начальный кусок f^z функции f, по которому однозначно определяется значение $\Psi(f)$. Далее, если дано f^n, то мы можем предельно вычислить $K(f^n)$ и, следовательно, проверить неравенство $K(f^n) > \sqrt{n}$. Отсюда следует, что $\Psi(f)$ можно предельно вычислить по z.

Теперь покажем, что Ψ не является предельно вычислимым функционалом. Допустим от противного, что существует машина $\mathfrak{M}$, которая предельно вычисляет Ψ. Так как $f \in R$, то без ограничения общности можно считать, что на входной ленте $\mathfrak{M}$ написаны значения f в естественном порядке: $f(1) f(2) f(3) \ldots$. Машина $\mathfrak{M}$, примененная к f, должна печатать на выходной ленте последовательность значений $k_1, k_2, k_3, \ldots,$ и эта последовательность должна стабилизироваться на значении $\Psi(f)$. Для получения противоречия мы теперь попытаемся построить функцию $f \in R$, на которой последовательность $k_1, k_2, k_3, \ldots$ не будет стабилизироваться. Конструкция будет состоять из отдельных шагов.

Машина $\mathfrak{M}$ при выдаче k_i использует определенный начальный кусок $f(1)\cdots f(s)$ функции f - это тот начальный кусок, который на входной ленте находится левее считывающей головки в момент, когда $\mathfrak{M}$ заканчивает печатать k_i.

Ш а г 0. Берем функцию 0^∞ и применяем к ней машину $\mathfrak{M}$. Так как 0^∞ - общерекурсивная функция, то машина $\mathfrak{M}$ когда-нибудь напечатает первое значение k_1. Пусть $\alpha_0^!$ - начальный кусок функции 0^∞, который использует $\mathfrak{M}$ при выдаче k_1. Дополним этот кусок справа нулями, чтобы общая длина была равна ближайшему n_{i_1} и обозначим его через α_0. Начальный кусок искомой функции f определим равным α_0.

Ш а г I. Рассматриваем всевозможные двоичные слова α длины $n_{i_1+1} - n_{i_1}$ и соответствующие функции $\alpha_0\alpha 0^\infty$. Применяем параллельно ко всем этим функциям машину $\mathfrak{M}$ (с того места, где она закончила свою работу при выдаче k_1). Из утверждения С вытекает, что обязательно найдется такое $\alpha=\alpha'$, что на $\alpha_0\alpha' 0^\infty$ машина $\mathfrak{M}$ когда-нибудь выдаст $k_{j_1} \neq k_1$. Пусть $\alpha_0\alpha' 0^s$ - начальный кусок функции $\alpha_0\alpha' 0^\infty$, который использует $\mathfrak{M}$ при выдаче k_{j_1}. Дополним этот кусок справа нулями, чтобы общая длина была равна ближайшему n_{i_2} и полученный кусок обозначим через α_1. Начальный кусок функции f доопределим равным α_1.

Ш а г m. ($m = 2,3,\ldots$) аналогичен шагу I, только вместо α_0 берется начальный кусок α_{m-1}, построенный на предыдущем шаге, и ищется $k_{j_m} \neq k_{j_{m-1}}$. Начальный кусок функции f доопределяем равным α_m.

Таким образом, нами построена общерекурсивная функция f, на которой машина $\mathfrak{M}$ выдает $k_1 \neq k_{j_1} \neq k_{j_2} \neq \ldots$ $\ldots \neq k_{j_m} \neq \ldots$ и, следовательно, последовательность $k_1, k_2, k_3, \ldots$ не стабилизируется. Таким образом, машина $\mathfrak{M}$ не может предельно вычислить функционал Ψ.

Теорема доказана.

ЗАМЕЧАНИЕ. Аналогично можно рассматривать предельно вычислимые функционалы и операции на PR. В данном случае возникает вопрос о том, имеет ли место аналог теоремы Майхилла-Шепердсона (см.[2]). Пользуясь теми же конструкциями, которые были использованы при доказательстве теоремы I, можно показать, что существует предельно вычислимая операция на PR, которая не является предельно вычислимым функционалом на PR.

ЛИТЕРАТУРА

I. Фрейвалд Р.В. Предельно вычислимые функции и функционалы. - Настоящий сборник, стр.6-I9.

2. Роджерс Х. Теория рекурсивных функций и эффективная вычислимость, М., I972.

3. Колмогоров А.Н. Три подхода к определению понятия количества информации. - "Проблемы передачи информации", I965, I, № I.

О ПРЕДЕЛЬНЫХ ВЫЧИСЛЕНИЯХ НА НЕДЕТЕРМИНИРОВАННЫХ МАШИНАХ ТЬЮРИНГА

Р.В.Фрейвалд, К.М.Подниеко

Уже неоднократно [I, 2] изучались вычисления на недетерминированных машинах Тьюринга, т.е. на машинах, программы которых могут содержать команды с одинаковыми левыми частями, но с разными правыми. Если процесс работы машины приводит к ситуации, которую описывает общая левая часть таких команд, то возникает возможность различных р е а л и з а ц и й работы машины.

Для представления множеств недетерминированные машины применяются в двух различных смыслах.

Машина $\mathfrak{M}$ п е р е ч и с л я е т множество A, если для тех и только тех x, которые принадлежат A, существует реализация работы машины $\mathfrak{M}$, начинающаяся со значения аргумента x и кончающаяся результатом 1.

Машина $\mathfrak{M}$ р а с п о з н а е т множество A, если для каждого x существует реализация работы машины $\mathfrak{M}$, начинающаяся со значения аргумента x, кончающаяся результатом $\chi_A(x)$ (где $\chi_A(x)=1$, если $x \in A$; $\chi_A(x)=0$, если $x \bar{\in} A$), и, кроме того, невозможны реализации, приводящие к неверному результату.

Легко видеть, что A перечислимо на недетерминированной машине, если и только если A рекурсивно перечислимо и A распознаваемо на недетерминированной машине, если и только если A рекурсивно.

Х.Патнам [3] и Э.М.Голд [4] рассматривали т.н. предельные вычисления. Говорят, что рекурсивная функция $a(n)$ имеет предел b ($\lim\limits_{n} a(n)=b$) или, другими словами, что последовательность ее значений стабилизируется на b, если $\exists n_0 \forall n > n_0\, a(n)=b$.

В соответствии с этим множество A называется п р е д е л ь н о р е к у р с и в н ы м, если для некоторой общерекурсивной функции $g(x,n)$ и всех x :

$$x \in A \to \lim_n g(x,n) = 1,\ x \bar{\in} A \to \lim_n g(x,n) = 0 .$$

Аналогично, A п р е д е л ь н о   п е р е ч и с л и м о, если

$$x \in A \leftrightarrow \lim_n g(x,n) = 1 .$$

для общерекурсивной g .

По существу, здесь рассматриваются машины Тьюринга с двумя лентами: рабочей и выходной. В начале работы на рабочей ленте записано значение аргумента x , выходная лента пуста. На рабочей ленте находится одна обычная читающая-пишущая головка, на выходной ленте - одна только-пишущая головка, которая может стоять на месте или двигаться вправо, но не может двигаться влево. Во время работы машины эта последняя головка печатает б е с к о н е ч н у ю последовательность нулей и единиц, соответствующую значениям $g(x,n)$ при $n = 0,1,2,\ldots$.

ТЕОРЕМА I. (Патнам, Голд) а) Множество A предельно перечислимо, если и только если $A \in \Sigma_2$
б) A предельно рекурсивно, если и только если $A \in \Sigma_2 \cap \Pi_2$.
(Определения классов Σ_2, Π_2 см. [5], § I4.I).

Упомянутое представление предельного перечисления с помощью двуленточных машин заставляет признать естественным также и следующее понятие.

Множество A с л а б о   п р е д е л ь н о   п е р е ч и с л и м о, если существует машина $\mathfrak{M}$ такая, что $x \in A$ если и только если при значении аргумента x $\mathfrak{M}$ печатает на выходе бесконечную последовательность нулей и единиц, стабилизирующуюся на 1 . (Таким образом, слабая предельная перечислимость отличается от предельной перечислимости тем, что здесь не требуется, чтобы машина $\mathfrak{M}$ печатала на выходе бесконечную последовательность п р и л ю б о м x).

ТЕОРЕМА 2. Множество A слабо предельно перечислимо, если и только если оно представимо в виде $A = B \setminus C$, где множества $B, C \in \Sigma_2$.

(В частности, класс слабо предельно перечислимых множеств включает полностью $\Sigma_2 \cup \Pi_2$, а сам содержится полностью в $\Sigma_3 \cap \Pi_3$).

Для доказательства необходимости в теореме 2 следует убедиться, что следующие два множества входят в Σ_2:

I) Множество B тех x, для которых машина $\mathfrak{M}$, слабо перечисляющая A, начиная с некоторого момента, не печатает на выходе нулей.

2) Множество C тех x, для которых машина $\mathfrak{M}$ печатает на выходе только конечную последовательность.

Очевидно, что $A = B \setminus C$.

Для доказательства достаточности в теореме 2 следует воспользоваться теоремой I.

Авторы задались целью изучить предельные вычисления на недетерминированных машинах. Используемые для определения таких вычислений машины отличаются от описанных выше двуленточных машин только недетерминированностью. Аналогично непредельному случаю недетерминированные машины предельного вычисления (НД МПВ) могут применяться для представления множеств в нескольких различных смыслах.

НДМПВ $\mathfrak{M}$ слабо перечисляет множество A, если для тех и только тех x, которые принадлежат A, существует реализация работы машины $\mathfrak{M}$, начинающаяся со значения x, при которой на выходной ленте печатается бесконечная последовательность, стабилизирующаяся на 1.

НДМПВ $\mathfrak{M}$ слабо распознает A, если для каждого x существует реализация работы машины $\mathfrak{M}$, начинающаяся со значения x, при котором на выходной ленте печатается бесконечная последовательность, стабилизирующаяся на $\chi_A(x)$, и, кроме того, не существует реализаций, стабилизирующихся на неверном результате.

Если потребовать дополнительно, чтобы машина $\mathfrak{M}$ при любом x и при любой реализации печатала на выходе обязательно бесконечную последовательность, получается, соответственно, понятия (н е с л а б о г о) перечисления и распознавания на НДМПВ.

Оказывается, возможности "слабых предельных вычислений" на детерминированных и недетерминированных машинах значительно отличаются. (Ср.теоремы 2,3; см.также теорему I, поскольку слабое предельное распознавание на детерминированной машине совпадает с предельной рекурсивностью).

ТЕОРЕМА 3. а) Множество A слабо перечислимо на НДМПВ, если и только если $A \in \Sigma_1^1$.
б) A слабо распознаваемо на НДМПВ, если и только если $A \in \Sigma_1^1 \cap \Pi_1^1$ (т.е. если A является гиперарифметическим множеством).
(Определения классов Σ_1^1, Π_1^1 см. [5], § I6.I).

Утверждение б) выводится из а) посредством довольно просто доказываемого аналога теоремы Поста; если A и дополнение к A слабо перечислимы на НДМПВ, то A слабо распознаваемо на НДМПВ.

Доказательство необходимости утверждения а) становится достаточно простым, если предварительно несколько видоизменить определение недетерминированной машины. Грубо говоря,"элемент недетерминированности" следует вынести из программы в особый вход машины. (Это напоминает сосредоточение "элемента случайности" при синтезе вероятностных машин в специальном датчике случайных чисел, см. [6]).

"Машина с лентой недетерминированности" получается из НДМПВ $\mathfrak{M}$, если добавить особую входную ленту, на которой может быть закодирована любая бесконечная последовательность $\{s(n)\}$ натуральных чисел. На ленте расположена одна только-читающая головка. Если в такте t работа $\mathfrak{M}$ приводит к ситуации, когда есть выбор из нескольких команд, новая машина обращается к входной ленте, читает $s(t)$ и выбирает (если возможно) команду с этим номером.

Если теперь в определениях представления множеств заменить "НДМПВ" на "машина с лентой недетерминированности" и "существует реализация работы машины" - на "существует запись на входной ленте", то, как легко видеть, мы получим понятия, эквивалентные исходным.

Достаточность а) можно доказать, исходя непосредственно из Σ_1^1 -формы множества A. Некоторые упрощения вносит, однако, предварительный переход к форме:

$$x \in A \longleftrightarrow \exists Y \forall y \exists z R(x, y, z, Y)$$

где Y - переменная для множеств, а R - рекурсивный предикат (см. [5], упражнение 16-10).

ЗАМЕЧАНИЕ. В доказательстве достаточности а) легко добиться выполнения дополнительного свойства: при любой реализации выходная последовательность (конечная или бесконечная) состоит только из единиц.

В случае обычных (неслабых) предельных вычислений, возможности детерминированных и недетерминированных машин, однако, совпадают (ср. теоремы 1, 4).

ТЕОРЕМА 4. а) Множество A перечислимо на НДМПВ, если и только если $A \in \Sigma_2$.

б) A распознаваемо на НДМПВ, если и только если $A \in \Sigma_2 \cap \Pi_2$.

Утверждение б) выводится из а) посредством подходящего аналога теоремы Поста.

Достаточность утверждения а) очевидна в силу теоремы 1.

Для доказательства необходимости а) следует показать сначала, что всякое A, перечислимое на НДМПВ, представимо в виде:

$$x \in A \longleftrightarrow \exists Y \exists y \forall z R(x,y,z,Y)$$

где Y - переменная для множеств, R - рекурсивный предикат. Затем можно показать, что отсюда следует $A \in \Sigma_2$ (ср. [5], упражнение 16-9).

Теорема 3 дает любопытную характеристику гиперарифметических множеств (а также Σ_1^1 - множеств) в машинных терминах. Уже ранее машинное представление гиперарифметических множеств (вместе с Π_1^1 -множествами) было получено в терминах обобщенных машин Клини [7] (см.также [5], § 16.5). Однако эта характеристика и наша значительно отличаются. В некотором смысле ситуация здесь аналогична случаю $\Sigma_2 \cap \Pi_2$ -множеств, которые также допускают различные описания в терминах машин. С одной стороны, это - множества, распознаваемые на машинах Тьюринга с креативным оракулом, а с другой - предельно рекурсивные множества.

Можно рассмотреть также в ы ч и с л е н и е ф у н к ц и й н а Н Д М П В. Полученный при этом класс слабо вычислимых частичных функций совпадает с классом всех Σ_1^1 -функций, а класс слабо вычислимых всюду определенных функций - с классом всех гиперарифметических функций.

(Неслабые) вычисления функций на НДМПВ дают класс всех Σ_2 -функций.

Некоторая "неестественность" понятия Σ_1^1 -вычислимости (отмеченная в [5], § 16.5) проявляется в нашей интерпретации так: хотя каждая НДМПВ слабо перечисляет некоторое множество (и таким образом мы получаем нумерацию этих множеств, которая оказывается изоморфной нумерации Σ_1^1 -множеств, построенной в [5], § 16.1), не всякая НДМПВ вычисляет функцию (т.е. нумерацию функций мы не получаем). Заметим, что обычный путь - через теорему об однозначности - также не может дать нумерацию Σ_1^1 -функций. Эта теорема в случае Σ_1^1 просто не имеет места, ибо из нее следовал бы (по методу [5], § 5.7) принцип редукции, опровергнутый для Σ_1^1 в 5 ; § 16.5.

Вопрос о возможности "вычислимых" (т.е. сводимых к нумерации Σ_1^1 -множеств, данной в [5], § 16.1) нумераций для Σ_1^1 -функций остается, по-видимому, открытым.

ЗАМЕЧАНИЕ. Легко видеть, что оформулированные теоремы 1-4 имеют место равномерно (в смысле [5]): переходы от машин к формулам и обратно здесь всегда эффективно выполнимы.

ЛИТЕРАТУРА

1. Myhill I. Linear bounded automata - Wright air Development, Tech.Note, 1960.
2. Savitch. Relationship between nondeterministic and deterministic tape complexities - "Journal of Computer and System Sciences", 1970, 4, No.2.
3. Putnam H. Trial and error predicates and the solution to a problem of Mostowski - "Journal of Symbolic Logic", 1965, 30, No.1.
4. Gold E.M. Limiting recursion - "J.Symbolic Logic", 1965, 30, No.1.
5. Роджерс Х. Теория рекурсивных функций и эффективная вычислимость, М., 1972.
6. де Леу К., Мур Э.Ф., Шеннон К., Шапиро Н. - Вычислимость на вероятностных машинах - Сборник "Автоматы", М., 1956.
7. Клини С. Функционалы конечных типов, вычислимые на машинах Тьюринга - Сборник "Математическая логика и ее применение", М., 1965.

ПРЕДЕЛЬНЫЕ ВЫЧИСЛЕНИЯ НА ВЕРОЯТНОСТНЫХ МАШИНАХ

Р.В.Фрейвалд

В статье де Леу, Мура, Шеннона и Шапиро [I] было введено понятие вероятностной машины. В настоящей работе вероятностные машины используются в качестве средств предельного вычисления функций. В [2] приведены определения предельной вычислимости и слабой предельной вычислимости функций в случае, когда вычисления проводятся на детерминированных машинах. Ниже будут введены вероятностные аналоги этих понятий и описано, как изменяются при этом классы вычислимых фнукций.

Статья [I] начинается с вопроса: имеются ли задачи, которые может решить машина со случайным элементом, но не может решить детерминированная машина? В [I] было показано, что многое зависит от вероятностей элементарных действий случайного элемента. Если эти вероятности не являются конструктивными действительными числами (определение см., например, упр. I5-34 в [3]), то вероятностные машины легко могут решать такие задачи. Поэтому ниже (следуя примеру многих других авторов) рассматриваются только такие вероятностные машины, которые отличаются от детерминированных машин Тьюринга (и их модификаций) лишь наличием простейшего датчика случайных чисел, который выдает символы 0 и 1 с равной вероятностью $\frac{1}{2}$ по схеме бернуллиевских испытаний, т.е. в разные моменты, независимо друг от друга.

В [I] рассматривалось несколько уточнений понятия вероятностной перечислимости множества. Было доказано, что перечислимы с вероятностью, превышающей некоторое конструктивное действительное число, только рекурсивно перечислимые множества, т.е. множества, перечислимые на детерминированных машинах. Эти результаты породили уверенность, будто любую задачу, которую может решить вероятностная машина с большой конструктивной вероятностью, может решить и подходящая детерминированная машина.

Я.М.Барздинь [4] показал, что это все же не совсем так. Им было доказано существование такого иммунного множества $A \in \Pi_1$ (определение иммунности и классов иерархии Клини-Мостовского см. [3]), что для каждого $\varepsilon > 0$ существует вероятностная машина, которая при каждой реализации работы перечисляет некоторое множество, и с вероятностью больше $1-\varepsilon$ оказывается, что перечисленное множество является бесконечным подмножеством множества A.

Позже были найдены и другие примеры задач, разрешимых на вероятностных, но не на детерминированных машинах (см. [5]). Однако все эти примеры характерны тем, что результатом работы машины является множество натуральных чисел с определенным свойством, притом при разных реализациях работы машины могут получаться разные множества.

Ниже доказано, что слабая предельная вычислимость имеет некоторые особенности (по сравнению с рекурсивной вычислимостью), вследствие которых аналог результатов [I] не имеет места. Построены примеры к о н к р е т н ы х функций, слабо предельно вычислимых на вероятностных машинах, но не на детерминированных машинах (см. следствия I и 2 предложения I и следствия 3 и 4 теоремы 3). С другой стороны, доказано, что любая функция, (неслабо) предельно вычислимая на вероятностной машине, предельно вычислима и на подходящей детерминированной машине (теорема I).

Формально вероятностной машиной предельного вычисления (ВМПВ) будем называть машину Тьюринга с простейшим бернуллиевским датчиком случайных чисел и двумя лентами: рабочей и выходной. В начале работы на рабочей ленте записано значение аргумента x , выходная лента пуста. На рабочей ленте находится одна обычная читающая-пишущая головка, на выходной - одна только-пишущая головка, которая может стоять на месте или двигаться вправо, но не может двигаться влево. Во время работы машины эта головка может печатать на выходной ленте записи натуральных чисел.

ОПРЕДЕЛЕНИЕ I. Функцию $\varphi(x)$ назовем предельно вычислимой с вероятностью больше p, если существует ВМПВ $\mathfrak{M}$, при любой реализации работы которой печатается бесконечная выходная последовательность, притом, при работе $\mathfrak{M}$ на x, если $\varphi(x)$ определена, то с вероятностью больше p печатается последовательность, стабилизирующаяся на $\varphi(x)$, и, кроме того, для каждого y, отличного от $\varphi(x)$ (и для всех y, если $\varphi(x)$ не определена) вероятность печати последовательности, стабилизирующейся на y, не превосходит p.

ОПРЕДЕЛЕНИЕ 2. Функцию $\varphi(x)$ назовем слабо предельно вычислимой с вероятностью больше p, если существует ВМПВ $\mathfrak{M}$, при работе которой на x, если $\varphi(x)$ определена, то с вероятностью больше p печатается бесконечная последовательность, стабилизирующаяся на $\varphi(x)$, и, кроме того, для каждого y, отличного от $\varphi(x)$, (и для всех y, если $\varphi(x)$ не определена) вероятность печатания бесконечной последовательности, стабилизирующейся на y, не превосходит p.

ОПРЕДЕЛЕНИЕ 3. Множество A назовем предельно рекурсивным (слабо предельно рекурсивным) с вероятностью больше p, если характеристическая функция множества A предельно вычислима (слабо предельно вычислима) с вероятностью больше p.

ОПРЕДЕЛЕНИЕ 4. Множество A назовем предельно перечислимым (слабо предельно перечислимым) с вероятностью больше p, если частичная характеристическая функция множества A предельно вычислима (слабо предельно вычислима) с вероятностью больше p.

ТЕОРЕМА I. Если p - конструктивное действительное число $(0 \leqslant p < 1)$, то функция φ (неслабо) предельно вычислима с вероятностью больше p тогда и только тогда, когда φ предельно вычислима на детерминированной машине.

ДОКАЗАТЕЛЬСТВО. ДОСТАТОЧНОСТЬ очевидна.

НЕОБХОДИМОСТЬ. Пусть ВМПВ $\mathfrak{M}$ предельно вычисляет φ с вероятностью больше p.

Совокупность α^b всех реализаций работы $\mathfrak{M}$ на x при которых выходная последовательность стабилизируется на b, можно представить как объединение $\bigcup_{i=1} \alpha_i^b$ совокупностей α_i^b всех реализаций работы $\mathfrak{M}$ на x, при которых все элементы выходной последовательности, напечатанные не раньше i - го такта, равны b.

Так как p -конструктивное действительное число, то существует эффективный процесс $\mathfrak{A}$, который, имея в качестве аргументов программу машины $\mathfrak{M}$, натуральные числа x, b, i и рациональное q, останавливается в тех и только тех случаях, когда вероятностная мера α_i^b при работе $\mathfrak{M}$ на x строго меньше $p+q$.

Поскольку $\alpha_1^b \subseteq \alpha_2^b \subseteq \alpha_3^b \subseteq \ldots$, то вероятностная мера α^b строго больше p тогда и только тогда, когда существует α_i^b, вероятностная мера которого больше p. Поэтому φ можно предельно перечислить следующей детерминированной машиной $\mathfrak{M}'$.

Обозначим через $r_1, r_2, r_3, \ldots$ произвольную строго монотонно убывающую к нулю рекурсивную последовательность рациональных чисел. Машина $\mathfrak{M}'$ берет тройки натуральных чисел (b, i, j) в порядке, например, канторовских номеров и в соответствии с тройкой (b, i, j) запускает процесс $\mathfrak{A}$ на программе $\mathfrak{M}, x, b, i, r_j$. Пока процесс не остановится, $\mathfrak{M}'$ в каждый момент печатает выходное значение b. Если процесс останавливается, $\mathfrak{M}'$ печатает значение, отличное от b, и потом берет следующую тройку (b', i', j'). Таким образом, выходная последовательность машины $\mathfrak{M}'$ стабилизируется на b тогда и только тогда, когда существуют такие i и j, что вероятностная мера α_i^b не меньше $p + r_j > p$.

СЛЕДСТВИЕ I. Если p - конструктивное действительное число $(0 \leq p < 1)$, то множество A (неслабо) предельно перечислимо с вероятностью больше p тогда и только тогда, когда $A \in \Sigma_2$.

СЛЕДСТВИЕ 2. Если p - конструктивное действительное число $(0 \leq p < 1)$, то множество A (неслабо) предельно рекурсивно с вероятностью больше p тогда и только тогда, когда $A \in \Sigma_2 \cap \Pi_2$.

ПРЕДЛОЖЕНИЕ I. Если p - конструктивное действительное число $(0 \leq p < 1)$, то множество A слабо предельно перечислимо с вероятностью больше p тогда и только тогда, когда $A \in \Sigma_3$.

ДОКАЗАТЕЛЬСТВО. ДОСТАТОЧНОСТЬ. Сначала докажем достаточность при $p = 0$. Пусть R - общерекурсивный предикат, такой, что

$$x \in A \Longleftrightarrow \exists i\, \forall j\, \exists k\ R(x,i,j,k).$$

Построим ВМПВ $\mathfrak{M}$, слабо предельно перечисляющую A. Сначала $\mathfrak{M}$, еще не обращая внимания на x, подсчитывает, сколько раз подряд бернуллиев датчик выдает 0. Результатом этого счета с вероятностью 2^{-1} будет 0, с вероятностью 2^{-2} будет $1, \ldots$, с вероятностью 2^{-i-1} будет $i, \ldots$ Пусть результатом оказалось i_0. Далее, $\mathfrak{M}$ вычисляет $R(x,i_0,0,0), R(x,i_0,0,1), R(x,i_0,0,2), \ldots$ до тех пор, пока какое-то из этих значений предиката окажется "истинно". После этого $\mathfrak{M}$ печатает первое выходное значение 1 и начинает вычислять $R(x,i_0,1,0), R(x,i_0,1,1), R(x,i_0,1,2), \ldots$. Когда среди этих значений оказывается "истинно", $\mathfrak{M}$ печатает второе выходное значение 1, переходит к вычислению $R(x,i_0,2,0), R(x,i_0,2,1), R(x,i_0,2,2), \ldots$ и т.д. Если $\exists i\, \forall j\, \exists k\ R(x,i,j,k)$ и i_1 - наименьшее значение, при котором $\forall j\, \exists k\, R(x,i_1,j,k)$, то бесконечная последовательность единиц будет напечатана с вероятностью не меньше 2^{-i_1-1}. Если же $\neg \exists i\, \forall j\, \exists k\, R(x,i,j,k)$, то таких реализаций работы $\mathfrak{M}$, при которых была бы напечатана бесконечная выходная последовательность, не существует вообще.

Остается рассмотреть случай $p>0$. Так как p - конструктивное действительное число, то существует такой процесс $\mathcal{B}$, реализуемый на обычной машине Тьюринга с простейшим бернуллиевским датчиком, что $\mathcal{B}$ останавливается с вероятностью, равной p. Требуемая для случая $p>0$ ВМПВ $\mathfrak{M}$ параллельно осуществляет процесс $\mathcal{B}$ и процесс, описанный в случае $p=0$. Если реализация процесса $\mathcal{B}$ останавливается, то моделирование процесса, описанного в случае $p=0$, прекращается, и $\mathfrak{M}$ начинает в каждый момент печатать символ 1. Таким образом, вероятность печати машиной $\mathfrak{M}$ бесконечной последовательности единиц строго больше p тогда и только тогда, когда $x \in A$. Символы, отличные от 1, $\mathfrak{M}$ не печатает.

НЕОБХОДИМОСТЬ. Для облегчения дальнейших рассуждений будем считать, что значение бернуллиевского датчика считывается один раз в течение каждого такта. Некоторые считывания тогда являются фиктивными, и реализации работы машины не будут отличаться. Тем не менее будем в подсчетах эти реализации учитывать как различные. Таким образом, возможны ровно 2^t реализаций первых t тактов работы машины.

Доказательство опирается на тот факт, что счетное объединение событий вероятности 0 есть событие вероятности 0. Поэтому условие слабой предельной перечислимости с вероятностью больше p можно описать так: существует рациональное положительное число q и существует натуральное n, что для всех натуральных z существует такое натуральное t, что среди всех 2^t реализаций первых t тактов работы $\mathfrak{M}$ на x имеется строго больше, чем $(p+q)\cdot 2^t$ таких реализаций, при которых машина за первые t тактов печатает не менее z выходных значений, и все они (допускается исключение лишь для первых n) равны единице.

Так как p - конструктивное действительное число, и неравенство в предыдущем предложении строгое, то предыдущее предложение описывает рекурсивный предикат, снабженный кванторной приставкой типа $\exists\forall\exists$.

СЛЕДСТВИЕ I. Существует множество, которое при любом конструктивном действительном p $(0 \leq p < 1)$ слабо предельно перечислимо с вероятностью больше p, но не перечислимо слабо предельно на детерминированных машинах.

ДОКАЗАТЕЛЬСТВО. По теореме 2 из [6] множество слабо предельно перечислимо на детерминированных машинах тогда и только тогда, когда оно представимо в виде разности двух множеств из Σ_2. Все такие множества содержатся в $\Sigma_3 \cap \Pi_3$, однако $\Sigma_3 \cap \Pi_3 \subsetneqq \Sigma_3$.

СЛЕДСТВИЕ 2. Существует частично определенная функция, которая при любом конструктивном действительном p $(0 \leq p < 1)$ слабо предельно вычислима с вероятностью больше p, но не вычислима слабо предельно на детерминированных машинах.

ДОКАЗАТЕЛЬСТВО. Таким свойством обладают частичные характеристические функции множеств из следствия I.

ПРЕДЛОЖЕНИЕ 2. Если p - конструктивное действительное число, и частичная функция φ слабо предельно вычислима с вероятностью больше p, то график функции φ принадлежит Σ_3.

ДОКАЗАТЕЛЬСТВО. Пусть ВМПВ $\mathfrak{M}$ слабо предельно вычисляет φ с вероятностью больше p. Легко перестроить $\mathfrak{M}$ в машину, слабо предельно перечисляющую с вероятностью больше p график функции φ. Вследствие предложения I, в таком случае график φ принадлежит Σ_3.

ТЕОРЕМА 2. Функция φ слабо предельно вычислима с вероятностью больше 0 тогда и только тогда, когда ее график принадлежит Σ_3.

ДОКАЗАТЕЛЬСТВО. НЕОБХОДИМОСТЬ следует из предложения 2.

ДОСТАТОЧНОСТЬ. Из предложения 1 следует, что график функции φ можно слабо предельно перечислить с вероятностью больше 0 на подходящей ВМПВ $\mathfrak{M}$, выходной алфавит которой состоит только из одной буквы 1. Машину $\mathfrak{M}$ можно перестроить в ВМПВ $\mathfrak{M}'$, слабо предельно вычисляющую φ с вероятностью больше p. Сначала $\mathfrak{M}$, еще не обращая внимания на x, подсчитывает, сколько раз подряд бернуллиев датчик выдает 0. Результатом этого счета с вероятностью 2^{-1} будет 0, с вероятностью 2^{-2} будет 1 и т.д. Пусть результатом оказалось i. Тогда, далее, $\mathfrak{M}'$ моделирует работу $\mathfrak{M}$ на паре $\langle x, i \rangle$. В каждый момент, когда $\mathfrak{M}$ должна печатать символ 1, $\mathfrak{M}'$ на время прерывает моделирование, печатает вместо 1 число i и потом продолжает прерванное моделирование. Теорема доказана.

Наша дальнейшая цель - для любого конструктивного действительного p описать класс функций, слабо предельно вычислимых с вероятностью больше p. Из предложения 2 следует, что для слабой предельной вычислимости функции с вероятностью больше некоторого конструктивного действительного p необходимо, чтобы график φ принадлежал Σ_3. Одно достаточное условие дает

ПРЕДЛОЖЕНИЕ 3. Если область значений функции φ состоит из не более чем k элементов, p - конструктивное действительное число, $0 \le p < \frac{1}{k}$ и график функции φ принадлежит Σ_3, то φ слабо предельно вычислима с вероятностью больше p.

ДОКАЗАТЕЛЬСТВО опустим, так как оно вытекает из приведенного ниже доказательства теоремы 3.

Достаточное условие предложения 3 можно усилить, вместо ограничения числа элементов области значений функции φ допустив лишь, что для каждого x "достаточно эффективно" указывается множество, состоящее из не более чем k элементов, к которому принадлежит значение $\varphi(x)$, если оно определено.

Перед теоремой 3 докажем лемму. Следуя [3] (§ 5.6), через $\mathcal{D}_u$ обозначим конечное множество с каноническим индексом u (канонический индекс задает список всех элементов множества с явным указанием числа элементов).

ЛЕММА. Существует такая общерекурсивная функция $g(k,n,x,i,j)$, что I) если $k=0$ или $k=1$, то $g(k,n,x,i,j)=0$; 2) при любых $k\geq 2,n,x,i,j$ $\mathcal{D}_{g(k,n,x,i,j)}$ состоит из не более чем $(k-1)$ элементов; 3) при любых $k\geq 2,n,x,i,j$ $\mathcal{D}_{g(k,n,x,i,j+1)}\subseteq \mathcal{D}_{g(k,n,x,i,j)}$ и $\mathcal{D}_{g(k,n,x,i,j)}\subseteq$ $\subseteq\mathcal{D}_{g(k,n,x,i+1,j)}$; 4) при любых $k\geq 2,n,x$, если ВМПВ с номером n при значении аргумента x слабо предельно вычисляет результат y с вероятностью больше p, где $\frac{1}{k}\leq p<1$, то $y\in \lim\limits_{i}\lim\limits_{j}\mathcal{D}_{g(k,n,x,i,j)}$

ДОКАЗАТЕЛЬСТВО. Для определения $g(k,n,x,i,j)$ рассмотрим все реализации работы ВМПВ с номером n на x в течение первых $i+j$ тактов. В силу оговорки о подсчете числа реализаций (см. доказательство предложения I), таких реализаций ровно 2^{i+j}. Сначала разделим их на классы $K_{-1},K_0,K_1,K_2,\ldots$. Реализация попадает в класс K_{-1}, если до i-го такта не было напечатано ни одного выходного значения (или было начато печатание какого-то одного натурального числа, но еще не было закончено). Реализация попадает в класс $K_z(z\geq 0)$, если последнее выходное значение, полностью напечатанное к i-му такту, равно z

Классом K_{-1} мы не будем интересоваться. Рассмотрим остальные классы. Выбросим из каждого K_z все реализации, при которых в течение $(i+1)$-го, $(i+2)$-го,...,$(i+j)$-го тактов полностью было напечатано хотя бы одно выходное значение, отличное от z. Получится класс K'_z.

Определим $g(k,n,x,l,j)$ при $k=0,1$ равным нулю и при $k\geqslant 2$ как канонический индекс конечного множества всех тех t, при которых K'_t содержит более $\frac{1}{k}\cdot 2^{l+j}$ реализаций. Очевидно, тогда условия I), 2), 3) леммы выполнены. Условия I), 2), 3) обеспечивают существование предела $\lim_j \mathcal{D}_{g(k,n,x,l,j)}$ при любых k,n,x,l, включение $\lim_j \mathcal{D}_{g(k,n,x,l,j)} \subseteq \lim_j \mathcal{D}_{g(k,n,x,l+1,j)}$ при любых k,n,x,l и существование предела $\lim_l \lim_j \mathcal{D}_{g(k,n,x,l,j)}$ при любых k,n,x. Докажем, далее, условие 4). Пусть ВМПВ с номером n на значении аргумента x слабо предельно вычисляет результат y с вероятностью больше $p\geqslant\frac{1}{k}$. Заметим, что если вероятностная мера совокупности $\mathcal{L}$ всех реализаций, при которых печатается бесконечная выходная последовательность, стабилизирующаяся на y, строго больше $\frac{1}{k}$, то существует такое l_0, что строго больше $\frac{1}{k}$ уже вероятностная мера совокупности $\mathcal{L}_{l_0}$ всех реализаций, при которых последнее значение, напечатанное до l_0-го такта, равно y, а после l_0-го такта не печатаются выходные значения, отличные от y. Поэтому такой y принадлежит $\mathcal{D}_{g(k,n,x,l,j)}$ при любых $l\geqslant l_0$ и любых j. Следовательно, $y\in\lim_l \lim_j \mathcal{D}_{g(k,n,x,l,j)}$.

ТЕОРЕМА 3. Пусть p - конструктивное действительное число и $k\geqslant 1$ - такое натуральное число, что $\frac{1}{k+1}\leqslant p<\frac{1}{k}$. Функция φ слабо предельно вычислима с вероятностью больше p тогда и только тогда, когда график φ принадлежит Σ_3 и существует такая общерекурсивная функция $h(x,t)$, что I) при любых x,t $\mathcal{D}_{h(x,t)}$ состоит из не более, чем k элементов, 2) при любом x, если $\varphi(x)$ определена, то существует t_0, такое, что для всех $t>t_0$ $\varphi(x)\in\mathcal{D}_{h(x,t)}$.

ДОКАЗАТЕЛЬСТВО. НЕОБХОДИМОСТЬ. Принадлежность графика φ классу Σ_3 следует из предложения 2. Пусть функцию φ слабо предельно вычисляет с вероятностью больше p ВМПВ с номером n. Пусть g - функция, существование которой утверждает лемма. Тогда в качестве $h(x,t)$ можно взять $g(\kappa+1,n,x,t,0)$. Действительно, предел $\lim\lim \mathcal{D}_{g(\kappa+1,n,x,t,0)}$ существует для любых κ,n,x. Но тогда существуют i_0 и j_0, что для всех $j>j_0$ $\mathcal{D}_{g(\kappa+1,n,x,t,0)}=$ $=\lim\lim \mathcal{D}_{g(\kappa+1,n,x,t,0)}$. Из условия 3) леммы тогда вытекает, что в качестве t_0 для условия 2) теоремы 7 можно взять i_0+j_0.

ДОСТАТОЧНОСТЬ. По функции h определим κ вспомогательных функций $h_1,h_2,\ldots,h_\kappa$. Для определения $h_1(x,0),h_2(x,0),\ldots,h_\kappa(x,0)$ рассматривается множество $\mathcal{D}_{h(x,0)}$. Пусть $\mathcal{D}_{h(x,0)}=\{y_1,y_2,\ldots,y_s\}$, где $s\le\kappa$. Тогда определяется $h_1(x,0)=y_1, h_2(x,0)=y_2,\ldots,h_s(x,0)=y_s$, $h_{s+1}(x,0)=b+1,\ldots,h_\kappa(x,0)=b+\kappa-s$, где b - максимальное число в $\{y_1,y_2,\ldots,y_s\}$. Пусть, по индукции, уже определены $h_1(x,t-1),h_2(x,t-1),\ldots,h_\kappa(x,t-1)$. Для определения $h_1(x,t),h_2(x,t),\ldots,h_\kappa(x,t)$ рассматривается $\mathcal{D}_{h(x,t)}=\{z_1,z_2,\ldots,z_m\}, m\le\kappa$. Пусть $\{z_1,z_2,\ldots,z_m\}\cap\{h_1(x,t-1),h_2(x,t-1),\ldots,h_\kappa(x,t-1)\}=\{u_1,u_2,\ldots,u_q\}$, притом $u_1=h_{i_1}(x,t-1),\ldots,u_q=h_{i_q}(x,t-1)$. Тогда $h_{i_1}(x,t)=h_{i_1}(x,t-1)$, $\ldots, h_{i_q}(x,t)=h_{i_q}(x,t-1)$, а остальным функциям присваиваются значения из $\{z_1,z_2,\ldots,z_m\}\setminus\{u_1,u_2,\ldots,u_q\}$ в порядке возрастания. Если после этого некоторые функции еще не определены, они получают значения $b+1,b+2,\ldots$, где b - максимальное число в $\{z_1,z_2,\ldots z_m\}\cup\{h_1(x,t-1),\ldots,h_\kappa(x,t-1)\}$.

Так как $p<\frac{1}{\kappa}$, по предложению I существует ВМПВ $\mathfrak{M}$, которая слабо предельно перечисляет график φ с вероятностью больше $\kappa\cdot p$, притом выходной алфавит $\mathfrak{M}$ состоит только из одной буквы 1. Тогда для слабого предельного вычисления φ с вероятностью больше p подходит следующая ВМПВ $\mathfrak{M}'$. Сначала $\mathfrak{M}'$ выбирает одно из чисел $1,2,\ldots\kappa$ таким образом, что вероятность выбора для каждого из этих чисел равна $\frac{1}{\kappa}$. Пусть результатом выбора является число τ. Тогда $\mathfrak{M}'$, с одной стороны, вычисляет последовательность $h_\tau(x,0), h_\tau(x,1), h_\tau(x,2),\ldots$ и, с другой стороны, моделирует работу $\mathfrak{M}$ на паре $\langle x,h_\tau(x,0)\rangle$ до тех пор, пока впервые обнаружится такое t_1, что $h_\tau(x,t_1)\ne$

$\neq h_\tau(x,t_1-1)$. При этом, в каждый момент, когда $\mathfrak{M}$ должна печатать символ 1 , $\mathfrak{M}'$ на время прерывает моделирование, печатает вместо 1 число $h_\tau(x,0)$ и потом продолжает прерванное моделирование. Если такое t_1 обнаружится, то $\mathfrak{M}$ печатает подряд два различных выходных значения (например, 0 и 1), бросает моделирование работы $\mathfrak{M}$ на паре $<x,h_\tau(x,0)>$ и вместо этого начинает моделирование работы $\mathfrak{M}$ на паре $<x,h_\tau(x,t_1)>$. Если обнаружится такое $t_2>t_1$, что $h_\tau(x,t_2)\neq h_\tau(x,t_2-1)$, то $\mathfrak{M}'$ снова печатает подряд два различных выходных значения и переходит к моделированию работы $\mathfrak{M}$ на паре $<x,h_\tau(x,t_2)>$ и т.д.

Если предел $\lim\limits_t h_\tau(x,t)$ не существует, то бесконечно много раз будут напечатаны подряд два различных значения - следовательно, выходная последовательность не стабилизируется. Если предел $\lim\limits_t h_\tau(x,t)$ существует, то условная вероятность (при условии, что выбрано именно число τ) печатания бесконечной последовательности, стабилизирующейся на $\lim\limits_t h_\tau(x,t)$, больше $k\cdot p$, если пара $<x,\lim\limits_t h_\tau(x,t)>$ принадлежит графику функции φ (т.е. если $\lim\limits_t h_\tau(x,t)=\varphi(x)$), и не превосходит $k\cdot p$, если $<x,\lim\limits_t h_\tau(x,t)>$ не принадлежит графику функции φ . При этом, печатание бесконечной последовательности, стабилизирующейся на значении, отличном от $\lim\limits_t h_\tau(x,t)$,невозможно.

Подсчитаем теперь безусловные вероятности выдачи различных результатов. Если $\varphi(x)$ определена, то по условию 2) теоремы существует такой τ , что $\exists t_0 \forall t\geq t_0$ $h_\tau(x,t)=\varphi(x)$. Вероятность выбора именно этого τ равна $\frac{1}{k}$, и, если такой выбор сделан, то с условной вероятностью большей, чем $k\cdot p$, будет напечатана бесконечная последовательность, стабилизирующаяся на $\varphi(x)$.

Если две реализации работы $\mathfrak{M}'$ на x отличаются уже выбором числа τ , то при этих реализациях не могут быть напечатаны бесконечные последовательности, стабилизирующиеся на одном и том же числе. Следовательно, для любого y отличного от $\varphi(x)$,вероятность печатания бесконечной последовательности, стабилизирующейся на y , не превосходит $\frac{1}{k} \cdot k \cdot p = p$. Теорема доказана.

СЛЕДСТВИЕ I. Пусть p_1 и p_2 - конструктивные действительные числа и $0 \leqslant p_1 \leqslant p_2 < 1$. Тогда любая функция, слабо предельно вычислимая с вероятностью больше p_2 , слабо предельно вычислима и с вероятностью больше p_1.

СЛЕДСТВИЕ 2. Пусть p_1 и p_2 - такие конструктивные действительные числа, для которых существует натуральное $k \geqslant 1$ такое, что $\frac{1}{k+1} \leqslant p_1 \leqslant p_2 < \frac{1}{k}$. Тогда любая функция слабо предельно вычислима с вероятностью больше p_1 тогда и только тогда, когда она слабо предельно вычислима с вероятностью больше p_2.

СЛЕДСТВИЕ 3. Существует всюду определенная функция, которая при любом конструктивном действительном p $(0 \leqslant p < \frac{1}{2})$ слабо предельно вычислима с вероятностью больше p , но не вычислима слабо предельно на детерминированных машинах.

СЛЕДСТВИЕ 4. Если p - конструктивное действительное число и $\frac{1}{2} \leqslant p < 1$, то всюду определенная функция f слабо предельно вычислима с вероятностью больше p тогда и только тогда, когда f предельно вычислима на детерминированной машине.

Если с вероятностью большей, чем $p \geqslant \frac{1}{2}$, машина печатает бесконечную последовательность, стабилизирующуюся на каком-то y , то все остальные возможные результаты (печатание только конечной последовательности, печатание последовательности, стабилизирующейся на другом значении) могут иметь суммарную вероятность не больше $1-p \leqslant \frac{1}{2}$. Следовательно, любая функция, слабо предельно вычислимая с вероятностью больше $p \geqslant \frac{1}{2}$ (где p - не обязательно конструктивное число), слабо предельно вычислима с вероятностью больше $\frac{1}{2}$. Поэтому в следствии 4 требование конструктивности числа p лишнее.

СЛЕДСТВИЕ 5. Если $\frac{1}{2} \leq p < 1$, то всюду определенная функция f слабо предельно вычислима с вероятностью больше p тогда и только тогда, когда f предельно вычислима на детерминированной машине.

СЛЕДСТВИЕ 6. Если $\frac{1}{2} \leq p < 1$, то множество A слабо предельно рекурсивно с вероятностью больше p тогда и только тогда, когда $A \in \Sigma_2 \cap \Pi_2$.

СЛЕДСТВИЕ 7. Если $0 \leq p < \frac{1}{2}$, то множество A слабо предельно рекурсивно с вероятностью больше p тогда и только тогда, когда $A \in \Sigma_3 \cap \Pi_3$.

ПРЕДЛОЖЕНИЕ 4. Можно эффективно для любого натурального $k \geq 2$ найти номер (в главной нумерации всех частичных Σ_3 -функций) такой всюду определенной функции f, принимающей не более k различных значений, что график f принадлежит $\Sigma_3 \cap \Pi_3$, и, если p - такое действительное число (не обязательно конструктивное), что $p \geq \frac{1}{k}$, то f не является слабо предельно вычислимой с вероятностью больше p.

ДОКАЗАТЕЛЬСТВО. Определим вспомогательную функцию $h(x,i,j)$ как наименьшее натуральное число, не принадлежащее множеству $\mathcal{D}_{g(k,x,x,i,j)}$, где g - функция, определенная в лемме. Легко видеть, что значение функции h не превосходит $k-1$. Из доказанного в лемме следует, что при любых x,i существует предел $\lim_j h(x,i,j)$ и при любом x существует повторный предел $f(x)=\lim_i \lim_j h(x,i,j)$. Из этих свойств легко вытекает, что график функции f принадлежит $\Sigma_3 \cap \Pi_3$. Условие 4) леммы гарантирует, что f не является слабо предельно вычислимой с вероятностью большей p.

СЛЕДСТВИЕ I. Существует такая всюду определенная функция f_1, график которой принадлежит $\Sigma_3 \cap \Pi_3$, что f_1 не является слабо предельно вычислимой с вероятностью больше $p>0$ (даже если p - неконструктивное).

ДОКАЗАТЕЛЬСТВО. Пусть функции $f_2, f_3, f_4, \dots$ - функции, которые в смысле предложения 4 соответствуют числам 2, 3, 4,... Пусть $c(x,y)$, $l(x), r(x)$ - канторовские нумерационные функции, осуществляющие взаимно однозначное соответствие между $N \times N$ и N. Тогда $f_1(x) = f_{2+l(x)}(r(x))$.

СЛЕДСТВИЕ 2. Существует такая всюду определенная функция f, график которой принадлежит Π_3, что f не является слабо предельно вычислимой с вероятностью больше p ни при каком $p \in [0,1)$.

ДОКАЗАТЕЛЬСТВО. Релятивизацией рассуждения, приведенного в § 5 работы [7] можно показать, что существует всюду определенная функция $g(x)$, график которой принадлежит Π_3, но не Σ_3. В качестве $f(x)$ можно взять

$$f(x)=\begin{cases} g(n), \text{если } x=2n \\ f_1(n), \text{если } x=2n+1 \end{cases}$$

где f_1 - функция из следствия I.

Из следствий I и 2 и предложения 4 вытекает

ТЕОРЕМА 4. Пусть p_1 и p_2 - два конструктивных действительных числа, $0 \leq p_1 \leq p_2 < 1$. Если существует такое натуральное число k, что $p_1 < \frac{1}{k} \leq p_2$, то класс $\mathfrak{F}_{p_2}$ всех функций, слабо предельно вычислимых с вероятностью больше p_2, собственно содержится в классе $\mathfrak{F}_{p_1}$ всех функций, слабо предельно вычислимых с вероятностью больше p_1. В противном случае $\mathfrak{F}_{p_1} = \mathfrak{F}_{p_2}$.

ЛИТЕРАТУРА

I. Де Леу К., Мур Э.Ф., Шеннон К.Э., Шапиро Н. Вычислимость на вероятностных машинах.- Сборник "Автоматы", М., 1956.
2. Фрейвалд Р.В. Предельно вычислимые функции и функционалы.- Настоящий сборник,стр.6-19.
3. Роджерс Х. Теория рекурсивных функций и эффективная вычислимость, М., 1972.
4. Барздинь Я.М. О вычислимости на вероятностных машинах.- "ДАН СССР", 1969, 189, № 4.
5. Звонкин А.К., Левин Л.А. Сложность конечных объектов и обоснование понятий информации и случайности с помощью теории алгоритмов.- "УМН", 1970, 25, № 6.
6. Фрейвалд Р.В., Подниекс К.М. О предельных вычислениях на недетерминированных машинах Тьюринга.- Настоящий сборник,стр. 25-31.
7. Подниекс К.М. О сводимостях классов функций.- Сборник "Уравнения математической физики и теории алгоритмов", Рига, 1972.

О ЧАСТОТНЫХ ВЫЧИСЛЕНИЯХ НА БЕСКОНЕЧНЫХ ВЫБОРКАХ

Е.Б.Кинбер

Настоящая работа посвящена изучению некоторых концепций частотных и предельных частотных эффективных вычислений на бесконечных последовательностях натуральных чисел. В § I показано, что в случае, когда требуемая частота истинных значений функции должна достигаться "начиная с некоторого места", частотно предельно вычислимы только те функции, которые предельно вычислимы в обычном смысле (см. [I]). Отсюда следует аналогичный результат для обычных частотных вычислений. В § 2 рассматривается случай, когда нужная частота истинных значений функции достигается лишь "на бесконечно многих начальных фрагментах". В этой ситуации уже в обычном смысле можно частотно эффективно вычислять нерекурсивные функции. Аналогичное утверждение имеет место для предельных частотных вычислений.

Перейдем к точным определениям. Основные обозначения и терминология, относящиеся к частотным вычислениям в обычном смысле, заимствованы из [2].

Введем хеммингову метрику на множестве всех двоичных последовательностей γ длины n : $\tau(\gamma_1, \gamma_2) =$ числу позиций, в которых γ_1 и γ_2 отличаются. Частотное отклонение $\frac{\tau(\gamma_1,\gamma_2)}{n}$ обозначим через $\rho(\gamma_1, \gamma_2)$. Пусть теперь G и D — две бесконечные двоичные последовательности. Через $\rho_n(G,D)$ мы обозначим $\rho(\gamma, \delta)$, где γ и δ - начальные куски длины n последовательностей G и, соответственно, D. Далее, число $\underline{\rho}(G,D) = \varliminf_n \rho_n(G,D)$ назовем нижним отклонением G от D и число $\overline{\rho}(G,D) = \varlimsup_n \rho_n(G,D)$, соответственно, верхним отклонением. Очевидно, $0 \leqslant \underline{\rho}(G,D) \leqslant \overline{\rho}(G,D) \leqslant 1$.

Выборкой мы назовем любую бесконечную последовательность попарно неравных натуральных чисел: $J = i(1), i(2), \ldots, i(k), \ldots$.

Класс всех выборок обозначим через Ω_∞. Для фиксированных $n_1, n_2, \dots, n_k$ $\Omega_\infty(n_1; n_2, \dots, n_k)$ - это класс всех выборок $J \in \Omega_\infty$, для которых $l(1)=n_1, l(2)=n_2, \dots, l(k)=n_k$.

Бесконечные двоичные последовательности условно назовем предикатами. В дальнейшем мы не будем различать предикат Γ и множество, характеристической функцией которого он является. На предикаты мы будем переносить терминологию, относящуюся к соответствующим множествам; так, например, будем говорить: рекурсивный предикат, рекурсивно перечислимый (р.п.) предикат и т.п.

Через T обозначим эффективный оператор, определенный на Ω_∞ и преобразующий любую выборку $J \in \Omega_\infty$ в некоторую бесконечную двоичную последовательность TJ. Такой эффективный оператор можно представить в виде некоторого условного алгоритма $\mathfrak{M}$, например, в виде двухленточной машины Тьюринга. На одной ленте заранее записана некоторая выборка J. На другой ленте, первоначально пустой, машина $\mathfrak{M}$, начав работу, постепенно записывает следующие друг за другом элементы последовательности TJ.

Зафиксируем $\varepsilon > 0$ и эффективный оператор T. Введем следующие обозначения: $\widetilde{K}(T, \Omega_\infty, \varepsilon)$ - класс предикатов Γ, удовлетворяющих условию:

$$(1)\quad \forall J \in \Omega_\infty : \bar{\rho}(\Gamma J, TJ) < \varepsilon$$

$\underset{\sim}{K}(T, \Omega_\infty, \varepsilon)$ - класс предикатов Γ, удовлетворяющих условию:

$$(2)\quad \forall J \in \Omega_\infty : \bar{\rho}(\Gamma J, TJ) < \varepsilon .$$

Теперь при фиксированном $\varepsilon > 0$ мы определим класс $\widetilde{K}(\Omega_\infty, \varepsilon)$ следующим образом:

$$(3)\quad \Gamma \in \widetilde{K}(\Omega_\infty, \varepsilon) \iff \exists \text{ эфф.оператор } T \, \forall J \in \Omega_\infty : \bar{\rho}(\Gamma J, TJ) < \varepsilon$$

Если в этом определении заменить $\bar{\rho}$ на $\underline{\rho}$, то мы получим определение класса $\underset{\sim}{K}(\Omega_\infty, \varepsilon)$. Другие, более слабые концепции частотной вычислимости получаются, если в

(3) отказаться от требования равномерности по T. Другими словами, требуется лишь, чтобы для каждой выборки $J \in \Omega_\infty$ существовал эффективный оператор T, для которого выполнялось бы условие $\bar{\rho}(\Gamma J, TJ) < \varepsilon$ или $\underline{\rho}(\Gamma J, TJ) < \varepsilon$. Так мы получим определения классов $\bar{K}(\Omega_\infty, \varepsilon)$ и, соответственно, $\underline{K}(\Omega_\infty, \varepsilon)$. Обозначим, кроме того, через $\underline{K}(\Omega_\infty)$ класс предикатов Γ, удовлетворяющих условию

(4) $\forall J \in \Omega_\infty \ \exists$ эфф.оператор $T : \underline{\rho}(\Gamma J, TJ) = 0$.

В настоящей работе получены ответы на следующие вопросы, поставленные Б.А.Трахтенбротом в [2]: содержат ли классы $K(\Omega_\infty, \varepsilon)$, $\bar{K}(\Omega_\infty, \varepsilon)$, $\underline{K}(\Omega_\infty, \varepsilon)$ и $\underline{K}(\Omega_\infty)$ неэффективные предикаты при $\varepsilon \in (0, \frac{1}{2}]$.

Оказывается, что класс $\bar{K}(\Omega_\infty, \varepsilon)$ (и, следовательно, $\tilde{K}(\Omega_\infty, \varepsilon)$) при любом $\varepsilon \in (0, \frac{1}{2}]$ неэффективных предикатов не содержит; в то же время в классах $\underline{K}(\Omega_\infty, \varepsilon)$ и $\underline{K}(\Omega_\infty, \varepsilon)$ такие предикаты имеются.

ЗАМЕЧАНИЕ. Из определений непосредственно следует, что при любом $\varepsilon > 0$ $\underline{K}(\Omega_\infty) \subseteq \underline{K}(\Omega_\infty, \varepsilon)$ и $\underline{K}(\Omega_\infty, \varepsilon) \subseteq \underline{K}(\Omega_\infty, \varepsilon)$. Как показал Я.М.Барздинь в [6], класс $\underline{K}(\Omega_\infty, \varepsilon)$ при любом $\varepsilon \in (0, \frac{1}{2}]$ содержит все р.п. предикаты. При этом, как легко следует из теоремы I [7], существуют р.п. предикаты, которые не входят в $\underline{K}(\Omega_\infty)$. Пусть V - множество всех р.п. предикатов, R - множество рекурсивных предикатов. Пользуясь простыми диагональными аргументами, легко показать, что при любом $\varepsilon \in (0, \frac{1}{2}]$ $V \subseteq \underline{K}(\Omega_\infty, \varepsilon)$. Используя все эти факты, а также следствие из теоремы 2 данной статьи, получаем следующую систему включений: $\forall \varepsilon \in (0, \frac{1}{2}]$:

$$R = \tilde{K}(\Omega_\infty, \varepsilon) = \bar{K}(\Omega_\infty, \varepsilon) \subset \underline{K}(\Omega_\infty) \subset \underline{K}(\Omega_\infty, \varepsilon);$$

$$R = \bar{K}(\Omega_\infty, \varepsilon) \subset \underline{K}(\Omega_\infty, \varepsilon) \subset \underline{K}(\Omega_\infty, \varepsilon);$$

$$V \subseteq \underline{K}(\Omega_\infty,\varepsilon);\ \underline{K}(\Omega_\infty) \not\subseteq V;\ V \not\subseteq \underline{K}(\Omega_\infty);$$
$$\underline{K}(\Omega_\infty,\varepsilon) \not\subseteq V;\ V \not\subseteq \underline{K}(\Omega_\infty,\varepsilon).$$

Описание классов $\tilde{K}(\Omega_\infty, \varepsilon)$ получается как следствие из более общего результата § I о предельной частотной вычислимости - частотного аналога предельных эффективных вычислений (см. [I], [4], [5]).

Пусть эффективный оператор T преобразует произвольную выборку $J \in \Omega_\infty$ в бесконечную последовательность $(TJ)_1$, $(TJ)_2, \ldots, (TJ)_k, \ldots$ двоичных последовательностей. Такой эффективный оператор можно задать при помощи трехленточной машины Тьюринга $\mathfrak{M}$. На первой ленте машины $\mathfrak{M}$ записана выборка J. Вторая лента машины двумерна и ограничена, например, сверху и слева, третья лента - рабочая. Начав работу на пустой второй ленте, $\mathfrak{M}$ постепенно записывает на ней элементы последовательностей $(TJ)_k$, $k = 1, 2, \ldots$, печатая на n - м такте работы $l(n)$ -й элемент последовательности $(TJ)_{r(n)}$ ($l(n)$ и $r(n)$ - соответственно левый и правый элементы пары с номером n в канторовской нумерации); при этом элементы последовательности $(TJ)_k$ располагаются в k - й строке ленты. Пусть $l(k,r)$ - r -й член последовательности $(TJ)_k$, $l(r) = \lim\limits_k l(k,r)$ и TJ - последовательность $l(1), l(2), \ldots, l(r), \ldots$. Взяв в определениях (I), (2), (3) и последующих именно эту последовательность TJ, мы получим определения классов предикатов, частотно-вычислимых в пределе: $\tilde{L}(T,\Omega_\infty,\varepsilon)$, $\tilde{\Gamma}(T,\Omega_\infty,\varepsilon)$, $\tilde{L}(\Omega_\infty,\varepsilon)$, $\underline{L}(R_\infty,\varepsilon)$, $\tilde{\Gamma}(\Omega_\infty,\varepsilon)$, $\underline{L}(\Omega_\infty,\varepsilon)$

Отметим, что $\lim\limits_k l(k,r)$ может не существовать. В таком случае мы считаем, что $l(r)$ в TJ не определено, и в (I), (2), (3) и т.д. ΓJ на r - м месте обязательно отличается от TJ. Оператор T подобного типа в дальнейшем будем называть предельно эффективным.

§ I. О классах $T(\Omega_\infty,\varepsilon)$ и $\bar{T}(\Omega_\infty,\varepsilon)$.

ТЕОРЕМА I. Если $0<\varepsilon\leqslant\frac{1}{2}$, то класс $T(\Omega_\infty,\varepsilon)$ содержит только предельно рекурсивные предикаты.

Нам понадобятся две леммы, из которых утверждение теоремы следует немедленно. Но прежде мы приведем некоторые соображения и определения, необходимые для формулировки и доказательства этих лемм.

Рассмотрим следующую концепцию частотного вычисления, впервые представленную в [9]. Пусть рекурсивный оператор P перерабатывает n-ку натуральных чисел $(x_1,x_2,\ldots x_n)$ в n-ку нулей и единиц $(y_1,y_2,\ldots,y_n)$. Будем говорить, что предикат $\Gamma(x)$ (m,n)-вычислим $(m\leqslant n)$ оператором P, если каждой n-ке $(x_1,x_2,\ldots,x_n)$, где $x_i\neq x_j$, $i\neq j$, оператор P сопоставляет такую n-ку значений $(y_1,y_2,\ldots,y_n)$, что в системе равенств

$$\Gamma(x_1)=y_1,\ \Gamma(x_2)=y_2,\ldots,\Gamma(x_n)=y_n$$

по крайней мере m истинны.

Как было доказано в [3], если $\frac{m}{n}>\frac{1}{2}$, то любой предикат Γ, (m,n)-вычислимый оператором P, является рекурсивным. Используя теорему 5 из [IO], указанное утверждение можно несколько усилить. А именно, утверждение остается справедливым и в том случае, когда в n-ке $(y_1,y_2,\ldots,y_n)=P(x_1,x_2,\ldots,x_n)$ некоторые y_i могут быть неопределены; если y_i не определено, то считаем, что $y_i\neq\Gamma(x_i)$. Такой оператор P можно представить в виде n-ки $(\varphi_1,\varphi_2,\ldots,\varphi_n)$ частично-рекурсивных функций от n переменных. Если каждая из этих функций частично-рекурсивна с креативным оракулом $\varnothing'$, то легко показать, что предикат Γ будет рекурсивным с оракулом $\varnothing'$ (см. [8]). А это означает, что $\Gamma\in\Delta_2$ (или, другими словами, Γ предельно рекурсивен, см. [I]).

Используя результаты и технику работ [4] и [5], легко показать, что частично-рекурсивные с оракулом $\varnothing'$ предикаты от n переменных - это в точности те, которые можно представить в виде $\lim_k \varphi(k, x_1, x_2, \dots, x_n)$, где $\varphi(k,x_1,\dots,x_n)$ - общерекурсивный предикат и $\lim$ не определен в том случае, когда φ не определен. Поэтому n-ку $(\psi_1, \psi_2, \dots, \psi_n)$ частично-рекурсивных с $\varnothing'$ функций мы можем представить в виде оператора $\widetilde{P}$, который каждой n-ке чисел $(x_1, x_2, \dots, x_n)$ сопоставляет последовательность n-ок нулей и единиц $\{(y_1, \dots, y_n)_i\}_{i=0,1,\dots}$ такую, что

$$\lim_i (y_j)_i = \psi_j(x_1, x_2, \dots, x_n),\ 1 \leq j \leq n.$$

Нашей целью будет показать, что если $\Gamma \in T(\Omega_\infty, \varepsilon)$, то существует $\widetilde{P}$, "предельно" (m,n)-вычисляющий Γ: если

$$\widetilde{P}: (x_1, x_2, \dots, x_n) \longrightarrow \{(y_1, y_2, \dots, y_n)_i\}_{i=0,1,\dots}$$

и $\lim_i (y_j)_i = z_j \ (1 \leq j \leq n)$, то в системе равенств

$$\Gamma(x_1) = y_1,\ \Gamma(x_2) = y_2, \dots, \Gamma(x_n) = y_n$$

по крайней мере m истинны. Как следует из сказанного выше, в таком случае $\Gamma \in \Delta_2$.

Итак, пусть $0 < \varepsilon \leq \frac{1}{2}$ и $\Gamma \in T(\Omega_\infty, \varepsilon)$. Пусть, далее, $\{T_k\}_{k=0,1,\dots}$ - главная нумерация всех предельно эффективных операторов, $\mathfrak{M}_j$ - машина Тьюринга, соответствующая оператору T_j. Напомним, что каждая машина $\mathfrak{M}_j$ имеет три ленты, причем вторая лента двумерна и ограничена слева и сверху.

Работу любой машины $\mathfrak{M}_j$ на произвольной выборке $J \in \Omega_\infty$ можно представить следующим образом. Машина $\mathfrak{M}_j$, печатая элементы $i(k,\tau)$, одновременно просматривает на первой ленте все более длинные начальные фрагменты выборки J. Элементы выборки обрабатываются на третьей ленте, и получаемая таким образом информация влияет на выбор очередного $i(k,\tau)$.
Пусть $J_j(i(k,\tau))$ - начальный фрагмент выборки J, который просматривает $\mathfrak{M}_j$ перед тем, как напечатать $i(k,\tau)$. Через $i_j(k,\tau)$ обозначим число $i(k,\tau)$,

которое выдает машина $\mathfrak{M}_j$.

В дальнейшем неоднократно будут рассмотрены различные наборы натуральных чисел любой длины. При этом мы всегда будем считать, что различные компоненты в этих наборах не равны. Под продолжением произвольного набора чисел $(a_1, a_2, \dots, a_m)$ будем понимать любой набор $(a_1, a_2, \dots, a_m, a_{m+1}, \dots, a_n)$, где $m<n$.

ОПРЕДЕЛЕНИЕ ДОПУСТИМОСТИ ПРОДОЛЖЕНИЙ. Продолжение $(a_1, a_2, \dots, a_n)$ произвольного набора чисел $(a_1, a_2, \dots, a_m)$ назовем j -допустимым, если для каждого a_τ $(1 \leq \tau \leq m)$ выполняется одно и только одно из следующих условий:

(I) а) $\forall J \in \Omega_\infty(a_1, a_2, \dots a_n)\ \exists \lim\limits_{k} i_j(k,\tau)$

б) если $J \in \Omega_\infty(a_1, a_2, \dots, a_n)$ и
$J' \in \Omega_\infty(a_1, a_2, \dots, a_n)$, то
$\lim\limits_{k} i_j(k,\tau) = \lim\limits_{k} i'_j(k,\tau)$

(2) $\forall J \in \Omega_\infty(a_1, a_2, \dots, a_n)\ \forall k\ \exists t$
$\exists (a_1, a_2, \dots, a_s)$ - продолжение набора $J_j(k,\tau)$
$\forall J' \in \Omega_\infty(a_1, a_2, \dots, a_s)(i'_j(k+t,\tau) = i_j(k,\tau))$.

Пусть $a = (a_1, a_2, \dots, a_n)$ - j - допустимое продолжение набора $(a_1, a_2, \dots, a_m)$ и $J \in \Omega_\infty(a_1, a_2, \dots, a_n)$ - произвольная выборка. Обозначим через $[\mathfrak{M}_j J_m^{(a)}]$ такой набор $(\alpha_1, \alpha_2, \dots, \alpha_m)$, в котором для всех $\tau\ (1 \leq \tau \leq m)$:

$$\alpha_\tau = \begin{cases} \lim\limits_{k} i_j(k,\tau), & \text{если } a_\tau \text{ обладает свойством (I)} \\ \text{не определено, в противном случае} \end{cases}$$

СВОЙСТВО $R_j(\Gamma, \varepsilon)$. Будем говорить, что наоор $(a_1, a_2, \dots, a_m)$ обладает свойством $R_j(\Gamma, \varepsilon)$, если существует его j -допустимое продолжение $a = (a_1, a_2, \dots, a_n)$, для которого

(I.I) $\exists J \in \Omega_\infty(a_1, a_2, \dots, a_m) : \rho_m(\Gamma J, [\mathfrak{M}_j J_m^{(a)}] \geq \varepsilon)$

Как легко следует из определения, $[\mathfrak{M}_j\, \mathfrak{I}_m^{(a)}]\ \exists \mathfrak{I}$ в (I.I) можно заменить на $\forall \mathfrak{I}$. Действительно, $[\mathfrak{M}_j\, \mathfrak{I}_m^{(a)}]$ зависит только от a, но не зависит от $\mathfrak{I} \in \Omega_\infty(a_1, a_2, \ldots, a_n)$.

ЛЕММА I. Существует набор чисел $(b_1, b_2, \ldots, b_v)$ и число j такие, что никакое продолжение набора $(b_1, b_2, \ldots, b_v)$ не обладает свойством $R_j(\Gamma, \varepsilon)$.

ДОКАЗАТЕЛЬСТВО. Основная часть доказательства представляет собой процесс построения набора $(b_1, b_2, \ldots, b_v)$, который ведется по шагам. Допустим в качестве предположения индукции, что шаги $0, 1, \ldots, j-1$ уже проделаны.

<u>Шаг j</u>. К этому шагу мы перейдем лишь в том случае, если процесс построения $(b_1, b_2, \ldots, b_v)$ не был закончен на шаге $j-1$.

Пусть $(a_1, a_2, \ldots, a_s)$ - начальный фрагмент набора $(b_1, b_2, \ldots, b_v)$, построенный к этому моменту. Выясняем, существует ли его продолжение $(a_1, a_2, \ldots, a_{s_1})$, для которого выполняется $R_j(\Gamma, \varepsilon)$. Возможны два ответа:

I). Да. Пусть $a = (a_1, a_2, \ldots, a_{s_2})$ - j-допустимое продолжение набора $(a_1, a_2, \ldots, a_{s_1})$, существование которого гарантируется свойством $R_j(\Gamma, \varepsilon)$. Выделим в $\{a_1, a_2, \ldots, a_{s_1}\}$ подмножество $\{a_{u_1}, a_{u_2}, \ldots, a_{u_p}\}$ чисел, обладающих свойством (2) (из определения j-допустимости). Пусть $i_j(k_1, u_1), i_j(k_2, u_2), \ldots, i_j(k_p, u_p)$ - - последние числа, которые напечатает $\mathfrak{M}_j$ в столбцах $u_1, u_2, \ldots, u_p$, просмотрев на первой ленте только фрагмент $(a_1, a_2, \ldots, a_{s_2})$.

Выберем теперь настолько длинное продолжение $(a_1, a_2, \ldots, a_{s_3})$ набора $(a_1, a_2, \ldots, a_{s_2})$, что просмотрев на первой ленте $(a_1, a_2, \ldots, a_{s_3})$, машина $\mathfrak{M}_j$ на второй ленте напечатает числа $i_j(k_z + t_z, u_z) \neq i_j(k_z, u_z)$ $(1 \leq z \leq p)$. Как нетрудно убедиться, возможность выбора такого набора

$(a_1, a_2, \ldots, a_{\Delta_3})$ обеспечивается выполнением условия (2) для всех чисел $a_{u_1}, a_{u_2}, \ldots, a_{u_p}$.

Набор $(a_1, a_2, \ldots, a_{\Delta_3})$ возьмем в качестве начального фрагмента набора $(b_1, b_2, \ldots, b_v)$ и перейдем к шагу $j+1$.

II) Нет. Полагаем $(b_1, b_2, \ldots, b_v) = (a_1, a_2, \ldots, a_\Delta)$ и останавливаем процесс.

Покажем, что построенная нами последовательность $b_1, b_2, \ldots, b_k, \ldots$ конечна. Допустим, от противного, что это не так. Пусть $n_1, n_2, \ldots, n_i, \ldots$ - геделевы номера произвольного фиксированного предельно эффективного оператора Γ. Рассмотрим произвольный шаг n_i. Так как последовательность $b_1, b_2, \ldots, b_i, \ldots$ бесконечна, то на этом шаге мы встретимся с ситуацией I). Пусть $(b_1, b_2, \ldots, b_m)$ - тот фрагмент, который обладает свойством $R_{n_i}(\Gamma, \varepsilon)$. Тогда существует такое n_i-допустимое продолжение $b = (b_1, b_2, \ldots, b_m, b_{m+1}, \ldots, b_r)$, что

$$\rho_m(\Gamma J, [\mathfrak{M}_{n_i} J_m^{(b)}]) \geq \varepsilon,$$

где $J = b_1, b_2, \ldots, b_k, \ldots$.

Пусть $\alpha(m)$ - начальный фрагмент длины m последовательности $\lim_k (\mathfrak{M}_{n_i} J)_k$. Покажем, что $\alpha(m) = [\mathfrak{M}_{n_i} J_m^{(b)}]$.

Пусть произвольное b_p $(p \leq m)$ обладает свойством (2) (из определения n_i-допустимости). Как нетрудно убедиться, выборка J строится именно таким образом, чтобы для подобного b_p в последовательности $i_{n_i}(1,p), i_{n_i}(2,p), \ldots, i_{n_i}(k,p), \ldots$ было бесконечно много нулей и единиц. Это означает, что $\lim_k i_{n_i}(k,p)$ не определен. Если же b_p обладает свойством (I), то $\lim_k i_{n_i}(k,p)$, по определению $[\mathfrak{M}_{n_i} J_m^{(b)}]$, совпадает с числом, стоящим на p-м месте в $[\mathfrak{M}_{n_i} J_m^{(b)}]$. Итак, $[\mathfrak{M}_{n_i} J_m^{(b)}] = \alpha(m)$ и, следовательно,

$$(I.2)\ \rho_m(\Gamma J, \alpha(m)) \geq \varepsilon.$$

Так как n_i выбран произвольно, то (I.2) выполняется для бесконечно многих m. Итак, для любого предельно эффективного оператора $T: \bar{\rho}(\Gamma J, \lim_{k} (TJ)_k) \geq \varepsilon$.

Это, очевидно, противоречит тому, что $\Gamma \in \Im(\Omega_\infty, \varepsilon)$. Следовательно, $(b_1, b_2, \ldots, b_v)$ - конечный набор.

Предположим, что процесс построения $(b_1, b_2, \ldots, b_v)$ был закончен на шаге j. Легко видеть, что в этом случае никакое продолжение набора $(b_1, b_2, \ldots, b_v)$ не обладает свойством $R_{j+1}(\Gamma, \varepsilon)$. Лемма доказана.

ЛЕММА 2. Если существуют j и набор чисел $(b_1, b_2, \ldots, b_v)$ такие, что $(b_1, b_2, \ldots, b_v)$ не имеет продолжений, обладающих свойством $R_j(\Gamma, \varepsilon)$, то существуют такие m и $n \geq m$, что Γ предельно (m,n)-вычислим некоторым оператором $\mathcal{P}$ и $\frac{1}{2} < \frac{m}{n}$.

ДОКАЗАТЕЛЬСТВО. Пусть $(x_1, x_2, \ldots, x_s)$ - произвольное продолжение набора $(b_1, b_2, \ldots, b_v)$. Так как $(x_1, x_2, \ldots, x_s)$ не обладает свойством $R_j(\Gamma, \varepsilon)$, то для любого его j-допустимого продолжения $x = (x_1, x_2, \ldots, x_{s_1})$:

$$(I.3)\quad \forall J \in \Omega_\infty(x_1, x_2, \ldots, x_{s_1}): \rho_s(\Gamma J, [\mathfrak{M}_j J_s^{(x)}]) < \varepsilon.$$

В дальнейшем вместо "j-допустимый" будем говорить просто "допустимый" и вместо $\mathfrak{M}_j$ будем употреблять символ $\mathfrak{M}$.

Нетрудно видеть, что для $\varepsilon = \frac{1}{2}$ $R_j(\Gamma, \frac{1}{2})$ в формулировке леммы I можно заменить на $R_j(\Gamma, \varepsilon')$, где $\varepsilon' < \frac{1}{2}$. Для этого в доказательстве леммы I на каждом шаге нужно проверять выполнение свойства $R_j(\Gamma, \varepsilon_j)$, где $\varepsilon_k < \varepsilon_{k+1}$ и $\lim_{k} \varepsilon_k = \frac{1}{2}$. Если выборка $J = (b_1, b_2, \ldots, b_v)$ будет бесконечной, то для любого j и для бесконечно многих r_i будет выполняться неравенство:

$$\rho_{r_i}(\Gamma J, \lim_{k} (\mathfrak{M} J)_k) \geq \varepsilon_j$$

и $\lim_{i} \varepsilon_i = \frac{1}{2}$. Но тогда $\Gamma \notin \Im(\Omega_\infty, \frac{1}{2})$ - противоречие.

Итак, будем считать, что $\varepsilon < \frac{1}{2}$. Пусть $q_s =$

$=\max\left\{p \mid \frac{p}{s} < \varepsilon\right\}$. Очевидно, что (I.3) можно заменить на

$$(I.4)\ \forall J \in \Omega_\infty(x_1, x_2, \ldots, x_s): \rho_s(\Gamma J, [\mathfrak{M} J_s^{(x)}]) \leq \frac{q_s}{s}$$

Поскольку при любом s $\frac{q_s}{s} \leq \varepsilon < \frac{1}{2}$, то s можно выбрать столь большим, что $\frac{q_s}{s-v} < \frac{1}{2}$. Пусть $s - v = n$ и $n - q_s = m$; тогда $\frac{n-m}{n} = \frac{q_s}{s-v} < \frac{1}{2}$ и, следовательно, $\frac{m}{n} > \frac{1}{2}$.

Определим оператор $\widetilde{P}$ на всех n -ках натуральных чисел, которые не входят в $\{b_1, b_2, \ldots, b_v\}$. Этого вполне достаточно, так как значения Γ на $b_1, b_2, \ldots, b_v$ мы можем считать заранее известными. Зафиксируем произвольную n -ку $(x_1, x_2, \ldots, x_n)$. Пусть $x = (b_1, b_2, \ldots, b_v, x_1, x_2, \ldots, x_\eta)$ - допустимое продолжение набора $(b_1, b_2, \ldots, b_v, x_1, x_2, \ldots x_n)$. Тогда:

$$\forall J \in \Omega_\infty(x): \rho_{v+n}(\Gamma J, [\mathfrak{M} J_{v+n}^{(x)}]) \leq \frac{q_{v+n}}{v+n}$$

Пусть ΓJ_1^n - фрагмент последовательности ΓJ от $v+1$ -го члена по $v+n$ -й. Аналогично определим $[\mathfrak{M} J_{v+n}^{(x)}]_1^n$. Тогда в силу выбора m и n :

$$\forall J \in \Omega_\infty(x): \rho(\Gamma J_1^n, [\mathfrak{M} J_{v+n}^{(x)}]_1^n) \leq \frac{n-m}{n}.$$

Мы будем строить последовательность значений $\{(y_1, y_2, \ldots, y_n)_k\}_{k=0,1,\ldots}$ оператора $\widetilde{P}$ на $(x_1, x_2, \ldots, x_n)$ таким образом, чтобы выполнялось равенство $\lim\limits_k (y_1, y_2, \ldots, y_n)_k = [\mathfrak{M} J_{v+n}^{(x)}]_1^n$.

Ясно, что тогда Γ будет предельно (m, n) -вычислим оператором $\widetilde{P}$.

В дальнейшем мы будем рассматривать только такие продолжения $(c_1, c_2, \ldots, c_\tau)$ набора $(b_1, b_2, \ldots, b_v, x_1, x_2, \ldots, x_n)$, что, просмотрев на первой ленте кортеж $(c_1, c_2, \ldots, c_\tau)$, машина $\mathfrak{M}$ на второй ленте печатает в одном из столбцов $v+1, v+2, \ldots, v+n$ какое-нибудь число. Имея машину $\mathfrak{M}$, можно указать некоторый эффективный процесс перечисления всех таких наборов $(c_1,$

$c_2, \dots, c_r)$. Зафиксируем произвольный такой процесс. Пусть $<e_0(x)>$ - x - е в порядке перечисления продолжение набора $(b_1, b_2, \dots, b_v, x_1, \dots, x_n)$. Пусть уже определено $<e_0(x), e_1(y), \dots, e_r(z)>$. Тогда через $<e_0(x), e_1(y), \dots, e_r(z), e_{r+1}(u)>$ обозначим u-е в порядке перечисления продолжение набора $<e_0(x), e_1(y), \dots, e_r(z)>$.

Перейдем к процессу определения $\bar{P}(x_1, x_2, \dots, x_n)$ На некоторых шагах в этом процессе к произвольному числу из $(x_1, x_2, \dots, x_n)$ может быть приписан плюс; в произвольный момент рядом с числом может стоять любое конечное множество плюсов.

<u>Шаг 0</u>. Находим такой x, что, просмотрев на первой ленте $<e_0(x)>$, машина $\mathfrak{M}$ печатает по крайней мере по одному из чисел в каждом столбце $v+1, v+2, \dots, v+n$ второй ленты. В качестве $(y_1, y_2, \dots, y_n)_0$ берем последние числа в этих столбцах, которые $\mathfrak{M}$ напечатает, просмотрев на первой ленте $<e_0(x)>$. Переходим к шагу I.

<u>Шаг к</u> . Предположим, что на шаге $k-1$ мы находили набор $<e_0(x), e_1(y), \dots, e_{i-1}(u), e_i(v), \dots, e_r(z)>$. Пусть t_i - шаг, на котором мы находим $<e_0(x), e_1(y), \dots, e_{i-1}(u), e_i(1)>$ $(0 \leq i \leq r)$. Для каждого $i \leq min(n-1, r)$ проверяем, верно ли, что в течение шагов $t_i, t_i+1, \dots, t_{i+1}-1$ плюс был приписан менее, чем к $n-1$ элементам из $\{x_1, x_2, \dots, x_n\}$ (если $i=r$, то проверка происходит только до шага k). Допустим сначала, что на все $min(n-1, r)+1$ вопросов получен положительный ответ. Тогда, если $r<n$ то находим набор $<e_0(x), e_1(y), \dots, e_r(z), e_{r+1}(1)>$, а если $r \geq n$, то находим $<e_0(x), e_1(y), \dots, e_r(z+1)>$. В качестве $(y_1, y_2, \dots, y_n)_k$ берем набор последних чисел, которые $\mathfrak{M}$ напечатает в столбцах $v+1, v+2, \dots, v+n$ второй ленты, просмотрев на первой ленте найденный выше набор. Затем переходим к шагу $k+1$.

Предположим теперь, что для некоторого i на наш вопрос получен отрицательный ответ; пусть l - наименьшее среди таких. В этом случае в качестве $(y_1, y_2, \dots, y_n)_k$ мы берем набор последних чисел, которые $\mathfrak{M}$ напечатает в столбцах $v+1, v+2, \dots, v+n$ второй ленты, просмотрев на первой ленте набор $\langle e_0(x), e_1(y), \dots, e_{l-1}(u), e_l(v+1)\rangle$. Далее приписываем плюс рядом со всеми числами из $(x_1, x_2, \dots, x_n)$, против которых $(y_1, y_2, \dots, y_n)_{t_l}$ отличается от $(y_1, y_2, \dots, y_n)_k$. Переходим к шагу $k+1$.

Определение $\widetilde{P}(x_1, x_2, \dots, x_n)$ завершено.

Заметим, что на произвольном шаге k мы всегда находим такой набор $\langle e_0(x), e_1(y), \dots, e_\tau(z)\rangle$, в котором $\tau \leqslant n$. Отсюда следует, что существуют числа $s \leqslant n$ и k_0 такие, что на любом шаге $k \geqslant k_0$ мы всегда находим только те наборы $\langle e_0(x), e_1(y), \dots, e_\tau(z)\rangle$, в которых $\tau \geqslant s$. Другими словами, на любом шаге $k \geqslant k_0$ мы всегда находим набор чисел, который является продолжением некоторого фиксированного набора $\langle e_0(x), e_1(y), \dots, e_{s-1}(u)\rangle$ (если $s=0$, то таким набором является $(b_1, b_2, \dots, b_v, x_1, x_2, \dots, x_n)$).

Покажем, что набор $\langle e_0(x), e_1(y), \dots, e_{s-1}(u)\rangle$ является допустимым продолжением для $(b_1, \dots, b_v, x_1, \dots, x_n)$. Как легко следует из описания шага k, на любом шаге $k \geqslant k_0$ плюс может быть приписан не более чем к $n-s$ фиксированным числам $x_{l_1}, x_{l_2}, \dots, x_{l_{n-s}}$ из $(x_1, x_2, \dots, x_n)$. При этом, как следует из определения числа s, к каждому из чисел $x_{l_1}, x_{l_2}, \dots, x_{l_{n-s}}$ на шагах $k \geqslant k_0$ плюс приписывается бесконечно много раз. Предположим, что $\{x_{l_1}, x_{l_2}, \dots, x_{l_{n-s}}\} \neq \varnothing$. Поскольку произвольный x_{l_j} получает плюс бесконечно много раз, то для каждого набора $\langle e_0(x), e_1(y), \dots, e_{s-1}(u), e_s(v)\rangle$ должно существовать такое продолжение $\langle \dots \rangle$, что последние значения, которые $\mathfrak{M}$ напечатает в столбце $v+l_j$ второй ленты после просмотра $\langle e_0(x), e_1(y), \dots, e_{s-1}(u), e_s(v)\rangle$ и, соответственно, $\langle \dots \rangle$ отличаются. Отсюда легко следует, что числа $x_{l_1}, x_{l_2}, \dots, x_{l_{n-s}}$ обладают свойством (2) из определения допустимости.

Рассмотрим теперь произвольное число $x_j \notin \{x_{i_1}, x_{i_2}, \ldots, x_{i_{n-s}}\}$. Нетрудно видеть, что все числа, которые $\mathfrak{M}$ будет печатать в $v+j$ столбце второй ленты, просматривая на первой ленте произвольное продолжение набора $< e_0(x), e_1(y), \ldots, e_{s-1}(u) >$, одинаковы. А это и означает, что x_j обладает свойством (I).

Итак, $< e_0(x), e_1(y), \ldots, e_{s-1}(u) >$ является допустимым продолжением набора $(b_1, b_2, \ldots, b_v, x_1, \ldots, x_n)$. Далее, из определения последовательности $\{(y_1, y_2, \ldots, y_n)_i\}_{i=0,1,\ldots}$ легко следует, что

$$\lim_{\kappa} (y_1, y_2, \ldots, y_n)_\kappa = [\mathfrak{M} J^{(e)}_{v+n}]^n_1$$

для $e = < e_0(x), e_1(y), \ldots, e_{s-1}(u) >$ и любой выборки $J \in \Omega_\infty(e)$, что и т.д. Лемма доказана.

Из лемм I и 2 следует утверждение теоремы.

СЛЕДСТВИЕ I. Если $0 < \varepsilon \leq \frac{1}{2}$, то класс $\widetilde{L}(\Omega_\infty, \varepsilon)$ содержит только предикаты из Δ_2.

ДОКАЗАТЕЛЬСТВО. Очевидно.

СЛЕДСТВИЕ 2.[*] При любом $\varepsilon \in (0, \frac{1}{2}]$ классы $\widetilde{K}(\Omega_\infty, \varepsilon)$ и $\widetilde{K}(\Omega_\infty, \varepsilon)$ содержат только рекурсивные предикаты.

ДОКАЗАТЕЛЬСТВО. Если в доказательствах лемм I и 2 мы возьмем $\Gamma \in \widetilde{K}(\Omega_\infty, \varepsilon)$ и главную нумерацию обычных эффективных операторов, то, как нетрудно убедиться, в лемме 2 у нас получится такая последовательность $\{(y_1, y_2, \ldots, y_n)_i\}_{i=0,1,\ldots}$, что $(y_1, y_2, \ldots, y_n)_1 = (y_1, y_2, \ldots, y_n)_2 = \ldots = (y_1, y_2, \ldots, y_n)_i = \ldots$ Отсюда легко следует, что $\Gamma(x)$ (m, n) -вычислим рекурсивным оператором. Следовательно, Γ рекурсивен.

В заключение параграфа отметим, что не существует

---

*) Утверждение следствия 2 независимо от автора было доказано М.И.Дехтярем.

эффективной процедуры, посредством которой по оператору T, для которого $\widetilde{K}(T,\Omega_\infty,\varepsilon)\neq\varnothing$, можно было бы построить алгоритм для обычного вычисления некоторого предиката $\Gamma\in\widetilde{K}(T,\Omega_\infty,\varepsilon)$. Это следует из теоремы 2 [IO]. Тем не менее, можно показать, что для любого рационального $\varepsilon<\frac{1}{2}$ существует эффективная процедура, которая по любому эффективному оператору T, для которого $\widetilde{K}(T,\Omega_\infty,\varepsilon)\neq\varnothing$, доставляет такой эффективный оператор T', что

$$\forall\Gamma\in\widetilde{K}(T,\Omega_\infty,\varepsilon)\,\forall J\in\Omega_\infty:\bar{\rho}(\Gamma J,T'J)=0$$

§ 2. О классах $\underset{\sim}{K}(\Omega_\infty,\varepsilon)$ и $\underset{\sim}{k}(\Omega_\infty)$.

Введем следующее обозначение: $|M|$ - мощность множества M. Далее через $\underset{\sim}{K}(\Omega_\infty)$ обозначим класс предикатов Γ, удовлетворяющих следующему условию:

(2.I) $\exists$ эфф.оператор $T\,\forall J\in\Omega_\infty:\underset{\sim}{\rho}(\Gamma J,TJ)=0$.

Сравнение (2.I) с (4) и с определением класса $\underset{\sim}{K}(\Omega_\infty,\varepsilon)$ показывает, что $\underset{\sim}{K}(\Omega_\infty)\subseteq\underset{\sim}{k}(\Omega_\infty)$ и $\underset{\sim}{K}(\Omega_\infty)\subseteq\underset{\sim}{K}(\Omega_\infty,\varepsilon)$ при любом $\varepsilon>0$. В этом параграфе мы покажем, что класс $\underset{\sim}{K}(\Omega_\infty)$ содержит неэффективные предикаты. Очевидно, что в таком случае классы $\underset{\sim}{K}(\Omega_\infty,\varepsilon)$ и $\underset{\sim}{k}(\Omega_\infty)$ также содержат неэффективные предикаты.

Зафиксируем однозначную вычислимую нумерацию $P(x)$ всех кортежей натуральных чисел $(a_1,a_2,\dots,a_n)$, где $a_i\neq a_j$, $i\neq j$, любой длины. Пусть $l(P(x))$ - длина набора $P(x)$, $\max P(x)=\max\{a\mid a\in P(x)\}$. Если кортеж $P(x)=(a_1,a_2,\dots,a_n)$ является начальным фрагментом кортежа $P(y)=(a_1,\dots,a_m,a_{m+1},\dots,a_n)$, то будем записывать это следующим образом: $P(x)\lhd P(y)$.

ТЕОРЕМА 2. Класс $\underset{\sim}{K}(\Omega_\infty)$ содержит неэффективные предикаты.

ДОКАЗАТЕЛЬСТВО. Пусть $\{W_x\}_{x=0,1,\dots}$ - главная нумерация всех р.п. множеств. Зафиксируем некоторый эффективный процесс параллельного перечисления всех W_x. Через W_x^k обозначим множество элементов из W_x, которые были перечислены за k шагов. Выберем монотонно возрастающую общерекурсивную функцию f, удовлетворяющую следующему условию:

$$\lim_n \frac{f(n)}{f(n+1)} = 0 .$$

Основная часть доказательства заключается в построении такого эффективного оператора T, что для него и для некоторого неэффективного Γ выполняется (2.I).

Мы будем использовать приоритетный список, состоящий из всех натуральных чисел, расположенных в возрастающем порядке, и множество маркеров $\boxed{0}, \boxed{1}, \ldots, \boxed{k}, \ldots$. Построение оператора T ведется по шагам. На некоторых шагах с любым числом из списка может быть ассоциирован один из маркеров. В качестве специальной отметки будем использовать также минус, который на произвольном шаге может быть приписан рядом с любым числом из списка; получив на некотором шаге минус, число сохраняет его и в дальнейшем. Для любой выборки $J \in \Omega_\infty$ через $(TJ)^\tau$ обозначим начальный фрагмент длины τ последовательности TJ.

<u>Шаг 0</u>. Ассоциируем $\boxed{0}$ с числом 0. Переходим к шагу I.

<u>Шаг k</u>. Этот шаг разделяется на два этапа.

I) Пусть $d = \min\{a \mid a > \max_{z \leq k}(\max P(z))$ & у a нет минуса$\}$. Ассоциируем $\boxed{k}$ с $d+1$. Пусть $e_0^k, e_1^k, \ldots, e_k^k$ – текущие позиции маркеров $\boxed{i}$, где $0 \leq i \leq k$, и X_k – подмножество элементов из $\{e_0^k, e_1^k, \ldots, e_k^k\}$, рядом с которыми в этот момент нет минуса. Перечислим $W_0^k, W_1^k, \ldots, W_k^k$. Если не найдется ни одного числа $e_j^k \in X_k \cap W_j^k$, то переходим к этапу II. Предположим, что по крайней мере одно $e_i^k \in X_k \cap W_i^k$ найдется; пусть e_i^k – наименьшее среди них. Передвигаем маркеры, ассоциированные в этот момент с числами из $X_k \setminus \{e_0^k, e_1^k, \ldots, e_i^k\}$ к числам $d+2$,

$d+3$ и т.д., сохраняя порядок. Затем приписываем минус рядом со всеми числами z из $\{e_i^k, e_i^k+1, \dots, d, d+1\}$, которые его к этому моменту еще не имели. Переходим к этапу II.

II) Пусть $P(u)$ - произвольный набор, удовлетворяющий следующим условиям:

1) $u<k \mathbin{\&} P(u) \lhd P(k)$

2) $l(P(u)) = f(n)$ для некоторого n ,

3) $\dfrac{|\{a \mid a \in P(k) \mathbin{\&} a > \max P(u)\}|}{\max P(u)} \geqslant \dfrac{f(n+1)}{f(n)}$.

Мы намерены определить $(TJ)^{f(n)}$ для всех выборок $J \in \Omega_\infty(P(k))$. Пусть $\alpha_1 \alpha_2 \dots \alpha_z$ - уже определенный к этому моменту фрагмент последовательности TJ для некоторой выборки $J \in \Omega_\infty(P(k))$; будем считать, что $\alpha_1 \alpha_2 \dots \alpha_z$ имеет среди всех таких максимальную длину. Пусть далее $P(u) = (x_1, x_2, \dots, x_{f(n)})$. Для любой выборки $J \in \Omega_\infty(P(k))$ полагаем $(TJ)^{f(n)} = \alpha_1 \alpha_2 \dots \alpha_z \alpha_{z+} \dots \alpha_{f(n)}$, где для $i > z$

$$\alpha_i = \begin{cases} 0, & \text{если рядом с } x_i \text{ приписан минус} \\ 1, & \text{в противном случае.} \end{cases}$$

Переходим к шагу $k+1$.

По индукции легко доказывается, что оператор T определяется однозначно для каждой выборки $J \in \Omega_\infty$. Нетрудно видеть также, что T определен на всех выборках $J \in \Omega_\infty$. В самом деле, если некоторый кортеж $P(u)$ длины $f(n)$ является начальным фрагментом выборки J , то обязательно найдется такой кортеж $P(k)$, что $J \in \Omega_\infty(P(k))$ и для $P(k)$ выполняется 3). Следовательно, $(TJ)^{f(n)}$ будет определено уже на шаге k . Из определения T вытекает также, что T - эффективный оператор. Полагаем: $\Gamma(x)=0 \iff$ к x приписывается минус.

Пусть $x \in A \iff \Gamma(x) = 1$. Покажем, что множество A не является р.п. По индукции нетрудно доказать, что каждый маркер в нашем процессе двигается не более чем конечное число раз. Пусть $h(\kappa)$ - заключительная позиция маркера $\boxed{\kappa}$. Легко проверяется, что

$$\Gamma(h(\kappa))=0 \Longleftrightarrow h(\kappa)\in W_\kappa .$$

Отсюда следует, что A не является р.п. и, следовательно, предикат Γ не эффективен.

Покажем, что $\Gamma\in \underset{\sim}{K}(\Omega_\infty)$. Для этого нам нужно проверить, что для оператора T выполняется условие (2.I).

Пусть $J\in\Omega_\infty$. Если TJ отличается от ΓJ не более чем в конечном числе позиций, то (2.I) для выборки J выполняется очевидным образом. Предположим, что TJ отличается от ΓJ в бесконечно многих местах. Пусть, например, TJ отличается от ΓJ в n - м месте. Обозначим через $(x_1,x_2,\dots,x_n)$ начальный фрагмент выборки J длины n . Заметим, что обязательно $\Gamma(x_n)=0$. Действительно, если $\Gamma(x_n)=1$, то число x_n никогда не получит в нашем процессе минус. Но в таком случае для любой выборки $J'\in\Omega_\infty(x_1,x_2,\dots,x_n)$ n -ый член в последовательности TJ' должен быть равен единице; это противоречит нашему предположению.

Итак, $\Gamma(x_n)=0$. Пусть κ - шаг, на котором x_n получает минус. Легко видеть, что $(TJ)^n$ впервые полностью определено на некотором шаге $\tau<\kappa$: в противном случае n -й член в TJ был бы равен нулю. Пусть $P(u)$ - максимальный начальный фрагмент выборки J , на котором $(TJ)^{l(P(u))}$ определено к шагу κ , и пусть $l(P(u))=$ $=f(m)$. Из нашей конструкции следует, что существует набор $P(v)$ с номером $v<\kappa$ такой, что $J\in\Omega_\infty(P(v))$ и для $P(u)$ и $P(v)$ (на месте $P(\kappa)$) выполняется свойство 3). На шаге κ вместе с x_n минус получают все числа z , удовлетворяющие условию $x_n\leq z\leq \max P(v)$. В частности, минус будет приписан ко всем числам из множества $Y=\{a\mid a\in P(v)\ \&\ a>\max P(u)\}$. Пусть $x\in Y$ - $n+i$ - й член выборки J . Так как $n+i$ -й член последовательности TJ к шагу κ еще не был определен, то в силу определения T он обязательно будет равен нулю. Но $\Gamma(x)$, по определению Γ, также равно нулю. Пусть Z -множество элементов из $P(v)$, не входящих в Y. Мы получаем, что

$$\rho_{l(P(v))}(\Gamma J, TJ) \leq \frac{|Z|}{l(P(v))} \leq \frac{|Z|}{|Y|}$$

С другой стороны, так как $|Z| \leq \max P(u)$, то

$$\frac{|Z|}{|Y|} \leq \frac{\max P(u)}{|Y|} \leq \frac{f(m)}{f(m+1)}.$$

Отсюда следует, что

$$\rho_{l(P(v))}(\Gamma J, TJ) \leq \frac{f(m)}{f(m+1)}.$$

Так как TJ отличается от ΓJ на бесконечно многих местах, то неравенство

$$\rho_{n_l}(\Gamma J, TJ) \leq \frac{f(m_l)}{f(m_l+1)}$$

выполняется для бесконечно многих пар (n_l, m_l) таких, что $[(l<j) \Longrightarrow (n_l<n_j) \,\&\, (m_l<m_j)]$. В силу выбора функции f $\rho(\Gamma J, TJ)=0$. Теорема доказана.

СЛЕДСТВИЕ. В классе $\underset{\sim}{K}(\Omega_\infty)$ содержится р.п. предикат, который не является рекурсивным.

ДОКАЗАТЕЛЬСТВО. Легко видеть, что дополнение множества A, определенного в доказательстве теоремы, является р.п.

ЗАМЕЧАНИЕ I. Как доказано в [4], любой предикат $\Gamma \in \Delta_2$ можно представить в виде $\lim\limits_n \varphi(n,x)$, где φ - двухместный общерекурсивный предикат. Под вычислимой нумерацией некоторого множества предикатов из Δ_2 мы будем понимать вычислимую нумерацию таких двухместных общерекурсивных предикатов. Взяв в доказательстве теоремы 2 вместо $\{W_x\}_{x=0,1,\ldots}$ вычислимую нумерацию класса $\mathfrak{A}$ предикатов из Δ_2 и изменив соответствующим образом конструкцию, можно доказать следующее усиление теоремы 2: для любого класса $\mathfrak{A}$ предикатов из Δ_2, имеющего вычислимую нумерацию, существует предикат $\Gamma \in \underset{\sim}{K}(\Omega_\infty) \setminus \mathfrak{A}$.

ЗАМЕЧАНИЕ 2. Изменив нужным образом рассуждения в доказательстве теоремы 2, нетрудно показать, что классы $\underset{\sim}{L}(\Omega_\infty)$ и $\underset{\sim}{\perp}(\Omega_\infty)$ содержат предикаты, не являющиеся предельно рекурсивными.

Автор выражает признательность Р.В.Фрейвалду за постановку задачи и внимание к работе.

ЛИТЕРАТУРА

I. Фрейвалд Р.В. Предельно вычислимые функции и функционалы.- Настоящий сборник, стр. 6-19.
2. Трахтенброт Б.А. Частотные вычисления.- Труды Мат.ин-та им.В.А.Стеклова АН СССР, 1973, II3.
3. Трахтенброт Б.А. О частотном вычислении функции.- "Алгебра и логика", 1963, 2, № I.
4. Putnam M. Trial and error predicates and the solution to a problem of Mostowski.- "Journal of Symbolic Logic", 1965, 30, No.1.
5. Gold E.M. Limiting recursion.- "Journal of Symbolic Logic", 1965, 30, No.1.
6. Барздинь Я.М. О частотном решении алгоритмически неразрешимых массовых проблем.- "ДАН СССР", 1970, I7I, № 5.
7. Барздинь Я.М. Сложность и точность решения начальных кусков проблемы вхождения в рекурсивно перечислимое множество.- "ДАН СССР", I97I, I99, № 2.
8. Роджерс Х. Теория рекурсивных функций и эффективная вычислимость, М., 1972.
9. McNaughton R. The theory of automata, a survey, Advances in computers, 1961, 2.
I0. Кинбер Е.Б. Частотные вычисления общерекурсивных предикатов и частотное перечисление множеств.- "ДАН СССР", 1972, 205, № I.

СРАВНЕНИЕ РАЗЛИЧНЫХ ТИПОВ ПРЕДЕЛЬНОГО СИНТЕЗА И ПРОГНОЗИРОВАНИЯ ФУНКЦИЙ

К.М.Подниекс

1. Соглашения. φ_n - фиксированная геделевская нумерация всех 1-местных частично-рекурсивных функций (ч.р.ф.). R - класс всех 1-местных общерекурсивных функций (о.р.ф.). В дальнейшем классами называются только множества о.р.ф. Класс U называется эффективно перечислимым классом, если существует о.р.ф. $\alpha(i)$ такая, что $U=\{\varphi_{\alpha(i)} \mid i=0,1,2,\ldots\}$.

$\subset$ - строгое включение, $\subseteq$ - нестрогое.

$<x_1 \ldots x_n>$ - эффективная нумерация всех конечных кортежей натуральных чисел; в качестве номеров использованы все натуральные числа.

Стратегия - это любая (частичная) функция типа $N \to N$. Особо выделяются ч.р. и о.р. стратегии.

Если всюду определенную функцию φ представлять как последовательность значений $\varphi(0), \varphi(1), \varphi(2), \ldots$, то понятны обозначения $i\varphi$, $0^k 1 0^\infty$, $\bar{\alpha} 0^k \varphi$ и т.п. (здесь i, k - натуральные числа, $\bar{\alpha}$ - кортеж натуральных чисел, φ - всюду определенная функция). Например, $0^k 1 0^\infty$ обозначает функцию, которая равна нулю на всех x, за исключением $x=k$.

2. Предельный синтез. Предельным синтезом называется восстановление "в пределе" геделевского номера функции φ по данной последовательности ее значений: $\varphi(0), \varphi(1), \varphi(2), \ldots$. Для этой цели мы используем только о.р. стратегии. Если F - стратегия, а φ - всюду определенная функция, то значения $F(<\varphi(0) \ldots \varphi(n)>)$ при $n=0,1,2,\ldots$ называются гипотезами. Гипотеза p считается верной, если $\varphi_p=\varphi$, т.е. p - некоторый геделев номер функции φ.

Первое понятие предельного синтеза под названием "identification in the limit" изучалось Голдом [I, 2]. В наших терминах оно определяется следующим образом. Говорят, что о.р. стратегия F предельно синтезирует функцию φ в смысле GN, если последовательность гипотез $F(<\varphi(0)\ldots\varphi(n)>)(n=0,1,\ldots)$ имеет верную гипотезу в качестве предела, т.е. если она "стабилизируется" на некотором геделевом номере функции φ. (Символ GN означает "геделев номер").

Если о.р. стратегия F синтезирует в смысле GN каждую функцию класса U, то говорят, что F синтезирует класс U в смысле GN. В этом случае мы пишем: $U\in GN$ (т.е. символ GN понимается как семейство всех классов, предельно синтезируемых в смысле GN).

Результаты, полученные Голдом [I]:

а) Если класс U содержится в эффективно перечислимом классе о.р.ф., то $U\in GN$.

б) $R\bar{\in} GN$ (этот результат значительно усилен нашей теоремой I).

Отметим также один результат Я.М.Барздиня [3]

а) Существует класс U такой, что $U\in GN$, однако U не содержится ни в одном эффективно перечислимом классе о.р.ф. (В качестве U можно взять класс V из доказательства теоремы I в [4]).

Таким образом, семейство GN весьма нетривиально.

Следующее понятие предельного синтеза под названием "matching in the limit" рассматривалось Фелдманом [5] для языков. В нашем случае это означает следующее. Говорят, что о.р. стратегия F предельно синтезирует функцию φ в смысле GN^{∞}, если последовательность $F(<\varphi(0)\ldots\varphi(n)>)$ $(n=0,1,\ldots)$ состоит, начиная с некоторого места, только из верных гипотез. (Знак ∞ у символа GN указывает, что допускается даже бесконечное число различных гипотез на одной функции).

Если о.р. стратегия F синтезирует в смысле GN^{∞} каждую функцию класса U, то говорят, что F синтезирует

U в смысле GN^{∞}. Запись $U \in GN^{\infty}$ определяется аналогично GN.

Очевидно: $GN \subseteq GN^{\infty}$, так как все, что синтезируемо в смысле GN, синтезируемо и в смысле GN^{∞}. С другой стороны, Барздинь [4] доказал, что существует класс U такой, что $U \in GN^{\infty}$, однако $U \bar{\in} GN$. Таким образом: $GN \subset GN^{\infty}$.

Последним в нашей схеме является понятие "частотного" синтеза. Пусть ε - действительное число, $0 < \varepsilon \leq 1$. Говорят, что о.р. стратегия F предельно синтезирует функцию φ в смысле $GN(\varepsilon)$, если в последовательности $F(<\varphi(0)\dots\varphi(n)>)$ нижняя частота верных гипотез не меньше ε, т.е.

$$\varliminf_{n} \frac{card\{x \mid x < n \,\&\, \varphi_F(<\varphi(0)\dots\varphi(x)>) = \varphi\}}{n} \geq \varepsilon.$$

Семейство $GN(\varepsilon)$ определяется аналогично GN и GN^{∞}. Очевидно: $GN^{\infty} \subseteq GN(1)$, $\varepsilon > \delta \longrightarrow GN(\varepsilon) \subseteq GN(\delta)$.

ТЕОРЕМА I. Если $\varepsilon > 0$, то $R \bar{\in} GN(\varepsilon)$.

Эта теорема является упомянутым усилением результата Голда ($R \bar{\in} GN$) : оказывается, что класс всех о.р.ф. нельзя предельно синтезировать, например, даже с частотой 10^{-6}. Поэтому все семейства $GN(\varepsilon)$ (а также GN^{∞}) нетривиальны вместе с GN. Теорема I легко следует из теоремы 3.

ТЕОРЕМА 2. Если $\varepsilon > \frac{1}{2}$, то $GN(\varepsilon) = GN^{\infty}$.

Эта теорема выражает обычную ситуацию "детерминизации": если частота синтеза превосходит $\frac{1}{2}$, то можно построить стратегию, которая синтезирует тот же класс "в абсолютном смысле". Теорема 2 следует из теоремы 4, если предварительно установить, что при $\varepsilon > \frac{1}{2}$ имеет место $GN(\varepsilon) \subseteq NV''$ (см.дальше). Это несложно.

Теорему 2 нельзя "усилить": оказывается, что $GN^{\infty} \subset$

$GN\left(\frac{1}{2}\right)$ (это следует из теорем 2, 3). Из теоремы 3 следует, что $GN\left(\frac{1}{2}\right) \subset GN\left(\frac{1}{3}\right) \subset GN\left(\frac{1}{4}\right) \subset \ldots$ (можно предположить, что $GN(\varepsilon) \subset GN(\delta)$ для всех $\varepsilon, \delta : 0 < \delta < \varepsilon \leq \frac{1}{2}$)

ТЕОРЕМА 3. Если $q \geq 2$ - натуральное число и $\delta > 0$, то $GN\left(\frac{1}{q} + \delta\right) \subset GN\left(\frac{1}{q}\right)$.

ДОКАЗАТЕЛЬСТВО. Определим "почти-равенство" двух частичных функций φ, ψ так:

$$\varphi \overset{n}{=} \psi \longleftrightarrow \exists x_0 \forall x > x_0 \;\; \varphi(x) = \psi(x).$$

Соответственно: i - почти-геделев номер функции φ, если $\varphi_i \overset{n}{=} \varphi$. Для любых $0 < p \leq q$ определяется класс функций:

$$C_{pq} = \{\varphi \mid \varphi = i_1 \ldots i_q \varphi' \;\&\; \varphi' \in R \;\&\; \varphi_{i_k} \overset{n}{=} \varphi \text{ для } \geq p \text{ значений } k\}$$

Таким образом, если $\varphi \in C_{pq}$, то из первых q значений этой функции не менее p будут ее почти-геделевыми номерами.

1. Покажем сначала, что $C_{pq} \in GN\left(\frac{p}{q}\right)$. Обозначим для данной φ через j_{sn} (где $0 \leq s \leq q-1; n \geq 0$) некоторый геделев номер функции

$$\eta_{sn}(x) = \begin{cases} \varphi(x), \text{ если } x \leq n \\ \varphi_{\varphi(s)}(x), \text{ если } x > n \end{cases}$$

Тогда гипотеза $F(<\varphi(0) \ldots \varphi(n)>)$ полагается равной j_{sn}, если n при делении на q дает в остатке s. Очевидно, если $\varphi_{\varphi(s)} \overset{n}{=} \varphi$, то для всех достаточно больших n, которые при делении на q дают в остатке s, гипотеза $F(<\varphi(0) \ldots \varphi(n)>)$ будет верной. Поэтому на $\varphi \in C_{pq}$ стратегия F дает верные гипотезы с нижней частотой $\frac{p}{q}$, что и требовалось.

2. Теперь мы должны установить, что $C_{pq} \notin GN\left(\frac{p}{q} + \delta\right)$ при $p = 1, q \geq 2$, $\delta > 0$. Допустим противное: существует о.р. стратегия H, которая на всех функциях из C_{pq} дает вер-

ные гипотезы с частотой $\geq \frac{p}{q} + \delta$. Сразу же перейдем к подклассу C_{pq}, зафиксировав $i_q = a$, где $\varphi_a = 0^\infty$. Всякая функция вида $i_1 \dots i_{q-1} a \bar{\alpha} 0^\infty$ ($\bar{\alpha}$ - произвольный кортеж чисел) входит в C_{pq}, следовательно, стратегия H должна ее синтезировать с частотой $\geq \frac{p}{q} + \delta$.

Для построения "контрпримера" (такой о.р.ф., которая входит в C_{pq}, но, тем не менее H дает на ней верные гипотезы лишь с нижней частотой $\leq \frac{p}{q}$) мы должны будем рассмотреть действие H на различных кортежах $\bar{\alpha}$. Эти кортежи воспринимаются как начальные куски функций, поэтому H даст на $\bar{\alpha}$ всего $|\bar{\alpha}|$ гипотез ($|\bar{\alpha}|$ - длина кортежа $\bar{\alpha}$). Гипотезу i, которую H выдает где-нибудь на $\bar{\alpha}$, мы называем "приятной", если значение $\varphi_i(|\bar{\alpha}|)$ определено. ("Приятной" - потому, что такую гипотезу легко опровергнуть, переходя к кортежу $\bar{\alpha} y$, где $y \neq \varphi_i(|\bar{\alpha}|)$, что облегчает построение контрпримера).

Кортеж $i_1 \dots i_{q-1} a$, содержащий $q-1$ переменных, обозначим через τ и введем следующий предикат $(0 \leq l \leq q)$:

$$B(\tau, \bar{\alpha}, l) \equiv \exists \bar{\beta} \left(H \text{ на } \tau\bar{\alpha}\bar{\beta} \text{ не менее } \tfrac{l}{q}|\tau\bar{\alpha}\bar{\beta}| \text{ раз дает "приятные" гипотезы}\right)$$

Этот предикат рекурсивно перечислим, следовательно, для произвольных $\tau, \bar{\alpha}, l$ его истинность можно разрешить, задав единственный вопрос оракулу $\emptyset'$ (см. [7], гл.14).

Введем еще один предикат $(0 \leq l \leq q)$:

$$A(\tau, l) \equiv \forall \bar{\alpha} B(\tau, \bar{\alpha}, l).$$

Он входит в Π_2, поэтому его истинность решается при помощи оракула $\emptyset''$.

Поскольку на любой о.р.ф. φ со свойством $\varphi = \tau\bar{\alpha}0^\infty$ стратегия H дает верные гипотезы с нижней частотой $> \frac{p}{q}$, то $A(\tau, p)$ должно быть истинно для всех τ (напомним, что последняя компонента τ суть a, $\varphi_a = 0^\infty$, поэтому $\varphi \in C_{pq}$). Очевидно также, что при $l < q$: $\neg A(\tau, l) \rightarrow \neg A(\tau, l+1)$. Поэтому условие:

$$[\frac{l_0}{p} - \text{целое} \,\&\, p \leq l_0 < q - p \,\&\, A(\Gamma, l_0) \,\&\, \neg A(\Gamma, l_0 + p)] \vee$$
$$\vee [l_0 = q - p \,\&\, A(\Gamma, l_0)] \qquad (*)$$

определяет единственное число l_0, для каждого Γ (заметим, что $p \leq q - p$ при $\frac{p}{q} \leq \frac{1}{2}$). Здесь l_0 принимает одно из $]\frac{q}{p}[-1$ возможных значений; какое именно – это можно узнать, задав подходящие вопросы оракулу $\varnothing''$.

Теперь мы приступаем непосредственно к построению контрпримера, опровергающего предполагаемое свойство стратегии H. Сначала мы построим такой "контрпример" для каждого кортежа $\Gamma = l_1 \ldots l_{q-1}\, a$, затем применим подходящим образом теорему о неподвижной точке; в результате окажется, что один из "контрпримеров" входит в C_{pq}, это – противоречие, доказывающее теорему 3.

Если дано Γ, мы с помощью оракула $\varnothing''$ определяем l_0 из условия $(*)$. Дальше различаются два случая.

а) Выполняется первый член дизъюнкции $(*)$. Воспользуемся сначала тем, что $A(\Gamma, l_0 + p)$ ложно:

$\exists \bar\alpha \forall \bar\beta$ (H на $\Gamma\bar\alpha\bar\beta$ менее $\frac{l_0+p}{q}|\Gamma\bar\alpha\bar\beta|$ раз дает "приятные" гипотезы),

т.е. существует $\bar\alpha_0$ такой, что

$\forall \bar\beta$ (H на $\Gamma\bar\alpha_0\bar\beta$ более $(1 - \frac{l_0+p}{q})|\Gamma\bar\alpha_0\bar\beta|$ раз дает "неприятные" гипотезы).

Для каждого $\bar\beta$ все эти "неприятные" гипотезы заведомо неверны (не являются номерами о.р.ф.). Найти указанный кортеж $\bar\alpha$ можно, перебирая по порядку всевозможные $\bar\alpha$ и задавая оракулу $\varnothing'$ вопросы об истинности $B(\Gamma, \bar\alpha, l_0 + p)$.

Теперь начнем с этого $\bar\alpha_0$ построение контрпримера – некоторый о.р.ф. φ'. Так как $A(\Gamma, l_0)$ истинно, то по $\bar\alpha_0$ эффективно найдется $\bar\beta_0$ такой, что H на $\Gamma\bar\alpha_0\bar\beta_0$ не менее $\frac{l_0}{q}|\Gamma\bar\alpha_0\bar\beta_0|$ раз дает "приятные" гипотезы. Но тогда легко подобрать число γ_0 так, что на кортеже $\Gamma\bar\alpha_0\bar\beta_0\gamma_0$ не менее $\frac{l_0}{q}|\Gamma\bar\alpha_0\bar\beta_0|$ из этих "приятных" гипотез окажутся опровергнутыми. Затем берем $\bar\alpha_1 = \bar\alpha_0\bar\beta_0\gamma_0$ и эту процедуру повторяем, и так далее. В результате определена о.р.ф. $\varphi' = \Gamma\bar\alpha_0\bar\beta\gamma_0\bar\beta_1\gamma_1 \cdots$.

Заметим, что на любом начальном куске вида $\tau\bar{\alpha}_0 - \bar{\beta}_0\gamma_0\ldots\bar{\beta}_n$ частота "неприятных" гипотез, даваемых стратегией H, больше $1-\frac{l_0+p}{q}$. К этому можно прибавить частоту тех "приятных" гипотез, которые опровергнуты значением γ_n (следующим за $\bar{\beta}_n$). Эта частота $\geqslant \frac{l_0}{q}$, поэтому общая частота неверных гипотез на выбранном куске больше $1-\frac{l_0+p}{q}+\frac{l_0}{q}=1-\frac{p}{q}$. Так будет при любом n, потому нижняя частота в е р н ы х гипотез, которые H дает на φ', не превосходит $\frac{p}{q}$. Значит - φ' является контрпримером.

б) Выполняется второй член дизъюнкции (*). Так как здесь $A(\tau, q-p)$ истинно, то для каждого $\bar{\alpha}$ можно эффективно построить $\bar{\beta}$ такой, что H на $\tau\bar{\alpha}\bar{\beta}$ не менее $\frac{q-p}{q}\times|\tau\bar{\alpha}\bar{\beta}|$ раз дает "приятные" гипотезы. Эти гипотезы можно опровергнуть значением γ, следующим за $\bar{\beta}$. Итерации этого процесса дают, как и в предыдущем случае, функцию φ', на которой стратегия H дает верные гипотезы с нижней частотой $\leqslant\frac{p}{q}$. Таким образом, и здесь φ' - контрпример.

Итак, отправляясь от $\tau=i_1\ldots i_{q-1}\,a$, мы сначала задали несколько вопросов оракулу $\varnothing''$, получив в результате одно из $]\frac{q}{p}[-1$ возможных значений числа l_0 из условия (*). Затем, задав еще несколько вопросов, на этот раз - оракулу $\varnothing'$, мы сумеем построить в конце концов о.р.ф. $\varphi'=i_1\ldots i_{q-1}\,a\,\varphi''$, на которой стратегия H дает верные гипотезы с нижней частотой $\leqslant\frac{p}{q}$. Остается каким-то спосособом применить теорему о неподвижной точке, "заставив" одно из $i_s\,(s\leqslant q-1)$ стать почти-геделевым номером для φ'. Тогда получится, что $\varphi'\in C_{pq}$ - противоречие с предположением, что H синтезирует C_{pq} в смысле $GN(\frac{p}{q}+\delta)$.

Все предыдущее построение дает, по существу, некоторую функцию $\varphi'(\tau,x)$, которая вычислима с оракулом $\varnothing'$ и $\varnothing''$ (контрпримером для данного τ тогда будет функция $\lambda x\varphi'(\tau,x)$). От оракула $\varnothing''$ мы освобождаемся так: переходим от одной функции $\varphi'(\tau,x)$ к $]\frac{q}{p}[-1$ функциям $f_s(\tau,x)$ (где $1\leqslant s\leqslant]\frac{q}{p}[-1$), именно: $f_s(\tau,x)$ вычисляется по возможности так же, как $\varphi'(T,x)$, с использованием оракула $\varnothing'$, где это необходимо, однако вместо того, чтобы задавать вопросы $\varnothing''$, мы уже с самого начала "полагаем", что значение l_0 в условии (*) равно t_s (s-му значению из всех возможных). В тех случаях, когда (для данного T) это предположение неверно, $\lambda x f_s(T,x)$ будет только частичной функцией (например, поиск кортежа $\bar{x}_0$ в случае (a) может оказаться бесконечным). Однако, если для данного T число t_s совпадает с l_0 из условия (*), то $\lambda x f_s(T,x)=\lambda x \varphi'(T,x)$.

Освободимся теперь от оракула $\varnothing'$. Заменим его некоторой процедурой перечисления (креативного) множества $\varnothing'$! Если для определения ответов оракула $\varnothing'$ мы пользуемся тем, что уже перечислено к данному моменту, то ошибок при вычислении таким способом $f_s(T,x)$ нельзя избежать полностью. Однако в случае, когда (для данного T) число t_s совпадает с l_0 из условия (*), при вычислении в с е й функции $\lambda x f_s(T,x)$ оракулу $\varnothing'$ задается не более, чем конечное число вопросов (поиск $\bar{x}_0$ в случае (a)). Ошибки в ответах на эти вопросы (а также отошедший от "верного" направления процесс вычисления) можно "в пределе" скорректировать, поэтому, начиная с некоторого x_0, все значения функции $\lambda x f_s(T,x)$ будут вычислены правильно (когда дойдет очередь до них, ошибки в ответах $\varnothing'$ уже будут исправлены). Таким образом, мы вместо $f_s(T,x)$ вычислим некоторую ч.р.ф. $g_s(T,x)$ со свойством:

если t_s равно l_0 из условия (*), составленного для T, то

$$\lambda x g_s(T,x) \overset{\text{п}}{=} \lambda x f_s(T,x) = \lambda x \varphi'(T,x). \qquad (**)$$

По существу, мы имеем теперь $\left]\frac{q}{p}\right[-1 = q-1$ функций g_s от q аргументов $i_1,\dots,i_{q-1},x$:

$$g_s = g_s(i_1,\dots,i_{q-1},a,x).$$

Начинаем применять теорему о неподвижной точке: существует о.р.ф. $h_1(i_2,\dots,i_{q-1})$ такая, что

$$\lambda x g_1(h_1(\dots),i_2,\dots,i_{q-1},a,x) = \varphi_{h_1(\dots)}$$

Далее: $\lambda x g_2(h_1(h_2\ldots), h_2(\ldots), l_3, \ldots, l_{q-1}, a, x) = \varphi_{h_2(\ldots)}$,
наконец: $\lambda x g_{q-1}(h_1, \ldots, h_{q-1}, a, x) = \varphi_{h_{q-1}}$, где h_{q-1} – число. Поэтому кортеж чисел $\tau = h_1 \ldots h_{q-1} a$ обладает свойством $(1 \leq s \leq q-1): \lambda x g_s(\tau, x) = \varphi_{h_s}$. Если для этого кортежа составить условие $(*)$ и взять s таким, что $t_s = l_0$ из этого условия, то соответствующий контрпример будет обладать (в силу $(**)$) свойством:

$$\lambda x \varphi'(\tau, x) = h_1 \ldots h_{q-1} a \varphi \triangleq \varphi_{h_1} \vee \varphi_{h_2} \vee \ldots \dot{\vee} \varphi_{h_{q-1}}.$$

Итак, одно из первых q значений функции $\lambda x \varphi'$ суть ее почти-геделев номер, поэтому $\lambda x \varphi' \in C_{1q}$. Противоречие с предположением, что $C_{1q} \in GN(\frac{1}{q} + \delta)$ посредством стратегии H.

ЗАМЕЧАНИЕ I. Если в определениях типов предельного синтеза заменить о.р. стратегии на ч.р. (тогда $F(<\varphi(0)\ldots \ldots \varphi(n)>) =$ "неопределено" считается "неверной гипотезой"), то, как легко видеть, расширения семейств GN, GN^{∞}, $GN(\varepsilon)$ $(0 < \varepsilon \leq 1)$ не происходит. В случае GN это можно доказать простой процедурой "выжидание" (см. дальше первую часть доказательства теоремы 5) а в остальных случаях значение "неопределено" можно включить в гипотезу (например, выдать номер пустой функции).

3. П р о г н о з и р о в а н и е. Прогнозированием называется предсказание значения $\varphi(n+1)$ по данным значениям $\varphi(0), \ldots, \varphi(n)$ функции φ. Здесь мы будем пользоваться как ч.р., так и о.р. стратегиями. Если F -стратегия, а φ - (всюду определенная) функция, то значения $F(<\varphi(0)\ldots\varphi(n)>)$ при $n = 0, 1, 2, \cdots$ назовем п р о г н о з а м и. Прогноз считается верным, если он равен $\varphi(n+1)$, в противном случае прогноз считается оши-

бочным. (Если значение $F(<\varphi(0)\dots\varphi(n)>)$ неопределено, то это также считается ошибкой).

Говорят, что стратегия F прогнозирует функцию φ, если в последовательности $F(<\varphi(0)\dots\varphi(n)>)$ $(n=0,1,\dots)$ не более чем конечное число прогнозов ошибочны.

Если существует о.р. стратегия, прогнозирующая каждую функцию класса U, то мы пишем $U\in NV$ (символ NV означает "следующее значение"). Если существует ч.р. стратегия, прогнозирующая все функции класса U, мы будем писать $U\in NV''$.

Очевидно, $NV\subseteq NV''$, но здесь не хватает еще промежуточного NV'. $U\in NV'$ истинно, если и только если существует ч.р. стратегия, которая прогнозирует все функции класса U, и при этом для всех $\varphi\in U$ и всех n значение $F(<\varphi(0)\dots\varphi(n)>)$ определено. (Здесь случай $F(<\varphi(0)\dots\varphi(n)>)$ -неопределено" принимается сразу за "бесконечное число ошибок", это довольно естественно: сколько можно ожидать появление о д н о г о прогноза?) Очевидно, $NV\subseteq NV'\subseteq NV''$.

Прогнозирования в смысле NV, NV' введены Я.М. Барздинем [3, 6]; в частности, им получены результаты:

а) $U\in NV$ если и только если U содержится в некотором эффективно перечислимом классе о.р.ф.

б) Существует $U\in NV'$, который не содержится ни в одном эффективно перечислимом классе о.р.ф. (в качестве U можно взять класс $\mathcal{V}$ из доказательства теоремы I в [4]).

Таким образом: $NV\subset NV'$. Из теорем 4, 5 легко следует, что $NV'\subset NV''$.

4. С в я з ь м е ж д у п р о г н о з и р о в а н и е м и с и н т е з о м. Оказывается, что прогнозирование посредством ч.р. стратегий (в смысле NV'') "по силе" эквивалентно предельному синтезу в смысле GN^{∞}.

ТЕОРЕМА 4. $NV''=GN^{\infty}$

ДОКАЗАТЕЛЬСТВО. I) $U \in GN^{\infty} \longrightarrow U \in NV''$ доказывается очень просто.

2) Покажем, что $U \in NV'' \longrightarrow U \in GN^{\infty}$. Пусть $U \in NV''$ посредством ч.р. стратегии H. Определим следующую стратегию F: $F(<\varphi(0)\ldots\varphi(n)>)=j_n$, где j_n - некоторый геделев номер следующей функции η_n (по существу, η_n - это "экстраполяция" начального куска $\varphi(0)\ldots\varphi(n)$ посредством прогнозирующей стратегии H):

$$\eta_n(x)=\varphi(x), \text{ если } x \leqslant n.$$

$$\eta_n(n+1)=H(<\varphi(0)\ldots\varphi(n)>).$$

$$\eta_n(n+k+1)=\begin{cases} H(\varphi(0)\ldots\varphi(n)\eta_n(n+1)\ldots\eta_n(n+k)), & \text{если все входящие сюда значения } \eta_n \text{ определены,} \\ \text{неопределено,} & \text{в противном случае.}\end{cases}$$

Если $\varphi \in U$, то H прогнозирует φ, т.е. начиная с некоторого n функция η_n совпадает с φ, для этих n гипотезы $F(<\varphi(0)\ldots\varphi(n)>)$ будут верны. Итак, $U \in GN^{\infty}$ посредством F. Теорема 4 доказана.

Из упомянутых результатов Голда и Барздиня легко следует, что $NV \subset GN$. Но теорема 5 показывает, что даже прогнозирование в смысле NV' "слабее" предельного синтеза в смысле GN.

ТЕОРЕМА 5. $NV' \subset GN$.

ДОКАЗАТЕЛЬСТВО. I) Сначала покажем, что $NV' \subseteq GN$. Пусть $U \in NV'$ посредством ч.р. стратегии H. Построим следующую стратегию F (определение j_x и η_x см. в конце доказательства теоремы 4):

$$F(<\varphi(0)\ldots\varphi(n)>)=\begin{cases} j_k, \text{ где } k=\min\{x \mid x\leqslant n \,\&\, \forall y\leqslant n[\eta_x(y)=\varphi(y)]\} \\ \text{если значения } \eta_x(y) \text{ определены для всех } x,y\leqslant n, \\ \text{неопределено, в противном случае.}\end{cases}$$

Ясно, что F "пока что" лишь ч.р. стратегия (для $U \in GN$ требуется о.р.), тем не менее, если $\varphi \in U$, то:

а) $F(<\varphi(0)\ldots\varphi(n)>)$ определено для всех n (ибо H прогнозирует U в смысле NV', поэтому для всех n прогнов $H(<\varphi(0)\ldots\varphi(n)>)$ определен и, таким образом, все значения $\eta_x(y)$ также определены).

б) Начиная с некоторого $n=n_0$, будет $\eta_{n_0}=\varphi$ (ибо η_n экстраполирует $\varphi(0)\ldots\varphi(n)$ посредством стратегии H, которая прогнозирует функцию φ). Но тогда при достаточно большом n гипотеза $F(<\varphi(0)\ldots\varphi(n)>)$ меняться больше не будет: она стабилизируется на числе j_{n_0} - геделевском номере функции $\eta_{n_0}=\varphi$.

Теперь легко определить о.р. стратегию F', которая синтезирует класс U в смысле GN. Именно, если даны значения функции φ, зафиксируем некоторую процедуру параллельного вычисления значений $F(<\varphi(0)\ldots\varphi(n)>)$ для всех n, причем в момент времени t разрешается использовать только значения $\varphi(0)\ldots\varphi(t)$.

$$F'(<\varphi(0)\ldots\varphi(t)>)=\begin{cases} i, & \text{если } i \text{ - значение } F(<\varphi(0)\ldots\varphi(x)>), \text{ вычисленное до момента } t \text{ в последнюю очередь,}\\ 0, & \text{если до момента } t \text{ никаких значений не вычислено.}\end{cases}$$

Ясно, что F' будет о.р. стратегией, даже если F нигде не определена, и если последовательность $F(<\varphi(0)\ldots\varphi(n)>)$ стабилизируется на числе j, то и $F'(<\varphi(0)\ldots\varphi(n)>)$ стабилизируется на j. Поэтому из а), б) следует, что $U\in GN$ посредством F', что и требовалось доказать.

2) Построим теперь класс $U\in GN\setminus NV'$ (это построение существенно использует одну идею М.И.Аугустона).

$$U=\{\varphi\mid\varphi=\bar{\alpha}\,i\,\varphi'(0)\,i\,\varphi'(1)\,i\ldots\ \&\ \bar{\alpha} \text{ - кортеж четной длины } \&\ \varphi'\in R\ \&\ \varphi_i=\varphi\}$$

Из теоремы о неподвижной точке легко следует, что для всякого $\bar{\alpha}$ и всякой φ' найдется такое i, что $\varphi\in U$. Очевидно, о.р. стратегия F со свойством $F(<\varphi(0)\ldots\varphi(2n+1)>)=F(<\varphi(0)\ldots\varphi(2n)>)=\varphi(2n)$ синтезирует U в смысле GN.

Предположим теперь, что $U\in NV'$ посредством ч.р. стратегии H. В качестве начальных кусков функций класса U

выступают произвольные кортежи натуральных чисел. Но тогда, если H прогнозирует U именно в смысле NV', то любое значение $H(<\bar{x}>)$ должно быть определено, т.е. H является на самом деле о.р. стратегией. Итак, оказалось, что $U \in NV$. Отсюда, по теореме Барздиня (формулировку см. в п.3^0) класс U содержится в некотором эффективно перечислимом классе о.р.ф. Пусть $\{\tau_i\}$ -вычислимая нумерация этого класса. Определим новую нумерацию $\tau_i'(x)=\tau_i(2x)$. Тогда из упомянутого свойства класса U (каждая функция которого находится среди $\tau_0, \tau_1, \ldots$) следует, что $\{\tau_i'\}$ содержит все о.р.ф., что невозможно.

Поэтому $U \bar{\in} NV'$ и теорема 5 доказана.

Все предыдущие результаты сводятся в схему: $NV \subset NV' \subset$ $\subset GN \subset GN^{\infty} = NV'' = GN(\frac{1}{2}+\varepsilon) \subset GN(\frac{1}{2}) \subset GN(\frac{1}{3}) \subset \ldots$

ЗАМЕЧАНИЕ 2. (Совместно с Е.Б.Кинбером). Наряду с частотным синтезом можно интересоваться и частотным прогнозированием. Но здесь почти все результаты доказываются очень легко.

Для класса о.р.ф. U будем писать $U \in NV(\varepsilon)$, если существует о.р. стратегия F такая, что для любой функции $\varphi \in U$ в последовательности прогнозов F нижняя частота верных прогнозов не меньше ε. Очевидно: $NV \subseteq NV(1)$, $\varepsilon > \delta \rightarrow NV(\varepsilon) \subseteq NV(\delta)$.

Введем для каждого $\varepsilon \in (0,1)$ класс:

$$U_\varepsilon = \{\varphi \mid \varphi \in R \,\&\, \underline{\lim}\, \frac{card\{x \mid x \leq n \,\&\, \varphi(x)=0\}}{n} \geq \varepsilon\}.$$

Очевидно, $U_\varepsilon \in NV(\varepsilon)$ посредством о.р. стратегии $F(x) \equiv 0$. Можно показать, что $U_1 \in NV(1) \setminus NV$ (для этого следует доказать от противного, что U_1 не содержится ни в одном эффективно перечислимом классе о.р.ф.). Таким образом: $NV \subset NV(1)$. Непосредственно (построением о.р.ф.-контрпримера) доказывается, что $U_\varepsilon \bar{\in} NV(\delta)(\delta > \varepsilon)$, таким образом: $\varepsilon > \delta \rightarrow NV(\varepsilon) \subset NV(\delta)$.

Можно ввести также понятие $NV'(\varepsilon)$, заменив в предыдущем определении о.р. стратегии на ч.р.. Нетрудно показать, что $R \in NV''(1)$, т.е. что существует ч.р. стратегия, которая на любой о.р.ф. делает верные прогнозы с частотой 1. Здесь, таким образом, $NV''(\varepsilon) = NV''(1)$ для всех ε.

Для понятия $NV'(\varepsilon)$ доказываются те же результаты, что для $NV(\varepsilon)$, используя те же классы U_ε. Но взаимная связь семейств $NV(\varepsilon), NV'(\delta)$ совершенно неясна, в первую очередь, неясно, будет ли $NV(1) \subset NV'(1)$.

ЛИТЕРАТУРА

1. Gold E.M. Limiting recursion.- "Journal of Symbolic Logic", 1965, 3, No.1.
2. Gold E.M. Language identification in the limit.- "Information and Control", 1967, 10, No.5.
3. Barzdin J.M. Prognostication of automata and functions - Information Processing 71, North-Holland, 1972.
4. Барздинь Я.М. Две теоремы о предельном синтезе.- Настоящий сборник, стр.82-88.
5. Feldman J. Some decidability results on grammatical inference and complexity.- "Information and Control", 1972, 20, No.3.
6. Барздинь Я.М., Фрейвалд Р.В. Прогнозирование общерекурсивных функций.- "ДАН СССР", 1972, 206, № 3.
7. Роджерс Х. Теория рекурсивных функций и эффективная вычислимость. М., 1972.

ДВЕ ТЕОРЕМЫ О ПРЕДЕЛЬНОМ СИНТЕЗЕ ФУНКЦИЙ

Я.М.Барздинь

Определения и обозначения см. в [I].

I°. Естественно спросить: если два класса о.р.ф. V, V' синтезируемы в смысле GN, то будет ли синтезируемо объединение $V \cup V'$? Тот же вопрос возникает и для синтеза в смысле GN^{∞}.

Легко видеть, что если один из классов V, V' конечный, то ответ положительный. Также легко судить, что ответ положительный, если оба класса эффективно перечислимы. Однако в общем случае имеет место

ТЕОРЕМА I.*) Существуют классы о.р.ф. V, V' такие, что $V, V' \in GN$ (более того, V' - эффективно перечислимый класс), однако, $V \cup V' \bar{\in} GN^{\infty}$.

ДОКАЗАТЕЛЬСТВО. Пусть $c(x,y)$ - эффективная нумерация пар, $x = lc(x,y), y = rc(x,y)$. Если $\alpha = (x_1, \ldots, x_n)$ - кортеж натуральных чисел, то $c(j;\alpha) = (c(j,x_1), \ldots, c(j,x_n))$. Так как функции мы отождествляем с последовательностями натуральных чисел, то отсюда понятно обозначение вида $c(j;\varphi)$, где φ - функция.

Возьмем

$$V = \{c(l;\varphi) \mid \varphi \in R \,\&\, \varphi_l = \varphi\},$$

$$V' = \{\tilde{\alpha}\, 0^{\infty} \mid \tilde{\alpha} - \text{кортеж натуральных чисел}\}.$$

Эффективная перечислимость класса V' очевидна, поэтому согласно [4], $V' \in GN$. Далее, легко видеть, что класс V синтезирует в смысле GN следующая о.р. стратегия F:

$$F(<\varphi(0), \ldots, \varphi(n)>) = f(l\varphi(0)),$$

где о.р.ф. $f(l)$ определена условием $c(l;\varphi) = \varphi_{f(l)}$.

---

*) Эта теорема была сформулирована без доказательства в [2, 3]. Независимо она была доказана также Л. и М.Блюмами [6].

Покажем теперь, что $V \cup V' \bar{\in} GN^{\infty}$. Предположим противное, что некоторая о.р. стратегия H синтезирует $V \cup V'$ в смысле GN^{∞}. Тогда для любого натурального числа j и любого кортежа натуральных чисел $\bar{\alpha}$ можно эффективно построить кортеж 0^{κ} такой, что гипотеза $H(<c(j;\bar{\alpha}0^{\kappa})>)=i$, где $\varphi_i(|\bar{\alpha}|+\kappa)=c(j,0)$ (т.е. гипотеза "считает", что за $c(j;\bar{\alpha}0^{\kappa})$) должно следовать $c(j,0)$, $|\bar{\alpha}|$ - длина кортежа $\bar{\alpha}$). Это вытекает из того, что $c(j;\bar{\alpha}0^{\infty}) \in V \cup V'$, и, следовательно, стратегия H, начиная с некоторого места $c(j;\bar{\alpha}0^{k})$, должна выдавать в качестве гипотез геделевы номера функции $c(j;\bar{\alpha}0^{\infty})$.

Пользуясь такой возможностью находить 0^{κ} по $j,\bar{\alpha}$, мы строим теперь эффективно для каждого j следующую функцию f_j:

$$c(j;f_j)=c(j;0^{\kappa_0}10^{\kappa_1}10^{\kappa_2}1\ldots),$$

где 0^{κ_0} найдено для $\bar{\alpha}_0=\varnothing$, 0^{κ_1} - для $\bar{\alpha}_1=c(j;0^{\kappa_0}1)$, 0^{κ_2} - для $\alpha_2=c(j;0^{\kappa_0}10^{\kappa_1}1)$ и так далее. Очевидно, стратегия H не может синтезировать в смысле GN^{∞} ни одну из функций $c(j;f_j)$. Действительно, для любого t за $c(j;\bar{\alpha}_t 0^{\kappa_t})$ следует $c(j,1)$, тогда как гипотеза $H(<c(j;\bar{\alpha}_t 0^{\kappa_t})>)$ "считает", что за $c(j;\bar{\alpha}_t 0^{\kappa_t})$ следует $c(j,0)$.

Теорема о неподвижной точке дает j_0 такое, что $f_{j_0}=\varphi_{j_0}$; тогда $c(j_0;f_{j_0})=c(j_0;\varphi_{j_0})\in V$. Однако H эту функцию не синтезирует в смысле GN^{∞}. Таким образом, $V\cup V'\bar{\in} GN^{\infty}$.

Теорема доказана.

Построенные выше классы V и V' обладают еще тем свойством, что они прогнозируемы равномерно на всех выборках (определения см. в [7]). Таким образом, имеет место теорема (М.И.Аугустон):

Существуют классы V и V', которые прогнозируемы равномерно на всех выборках, но объединение которых не прогнозируемо в указанном смысле.

2°. Непосредственно из определений следует, что $GN \subseteq$ $\subseteq GN^{\infty}$. Возникает вопрос: имеет ли место строгое включение $GN \subset GN^{\infty}$? Ответ на этот вопрос дает следующая

ТЕОРЕМА 2. Существует класс о.р.ф. U , который синтезируем в смысле GN^{∞}, но не синтезируем в смысле GN .

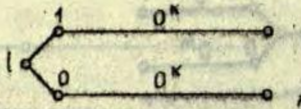
ДОКАЗАТЕЛЬСТВО. Приводимое построение класса U является в некотором смысле диагональной конструкцией. Функции φ_i рассматриваются как ч.р. стратегии F_i, и каждому i сопоставляется некоторый класс U_i , состоящий из одной или двух о.р.ф., который стратегия F_i не может синтезировать в смысле GN . Затем искомый класс U определяется как $U = \bigcup_{i=0}^{\infty} U_i$ и доказывается, что $U \in GN^{\infty}$.

Сначала перейдем от ч.р. стратегий F_i к эквивалентным о.р. стратегиям F_i' . Для каждой функции φ_i вводится некоторая процедура P_i параллельного вычисления гипотез $F_i(<\varphi(0),\dots,\varphi(n)>)$ для всех n , причем в момент времени t разрешается пользоваться только значениями $\varphi(0),\dots,\varphi(t)$. Тогда о.р. стратегия F_i' определяется так:

$$F_i'(<\varphi(0),\dots,\varphi(t)>) = \begin{cases} j, & \text{если } j \text{ - последняя гипотеза } F_i(<\varphi(0),\dots,\varphi(x)>) \text{, выданная до момента } t \text{;} \\ 0, & \text{если до момента } t \text{ ни одной гипотезы не выдано.} \end{cases}$$

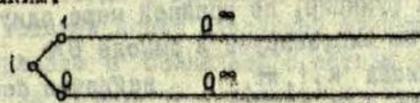
Ясно, что F_i' будет о.р. стратегией, даже если F_i нигде не определена, но если последовательность $\{F_i(<\varphi(0),\dots,\varphi(n)>)\}$ стабилизируется на некотором числе m , то и $\{F_i'(<\varphi(0),\dots,\varphi(n)>)\}$ тоже стабилизируется на этом же m . Таким образом, F_i' синтезирует в смысле GN по крайней мере все те функции, которые синтезирует в смысле GN F_i . Поэтому, если мы построим класс U_i , который не синтезирует в смысле GN стратегия F_i' , то этот класс не будет синтезировать в смысле GN также и стратегия F_i .

Класс U_i строится в виде дерева функций. Сначала это дерево "растет" по параметру k в виде



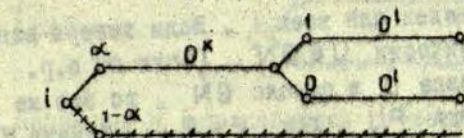
,

пока k не станет таким, что одна из гипотез $F_i'(<i10^k>)$, $F_i'(<i00^k>)$ отличается от гипотезы $F_i'(<i>)$. Если такого k не существует, то дерево функций класса U_i "вырастет" таким:

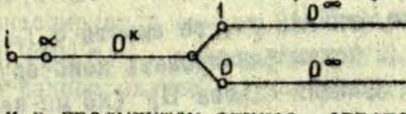


и мы положим $U_i = \{i10^\infty, i00^\infty\}$. В этом случае все гипотезы стратегии F_i' на обоих функциях класса U_i равны $F_i'(<i>)$, т.е. эта стратегия одну из функций не синтезирует.

Но если мы нашли число k с упомянутым свойством (т.е. $F_i'(<i\alpha 0^k>) \neq F_i'(<i>)$, где $\alpha = 0$ или 1), то дальнейший "рост" дерева происходит по параметру l в виде:

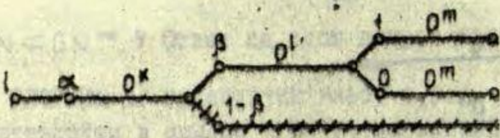


(зачеркнутая ветвь отбрасывается), пока l не станет таким, что одна из гипотез $F_i'(<i\alpha 0^k 10^l>)$, $F_i'(<i\alpha 0^k 00^l>)$ отличается от гипотезы $F_i'(<i\alpha 0^k>)$. Если такого l не существует, то дерево класса U_i "вырастет" таким:



Здесь, как и в предыдущем случае, стратегия F_i' по крайней мере одну из функций класса U_i не синтезирует.

Если же найдется l с упомянутым свойством (т.е. $F_i'(<i\alpha 0^k \beta 0^l>) \neq F_i'(<i\alpha 0^k>)$, где $\beta = 0$ или 1), то дальнейший "рост" происходит по параметру m в виде:



Аналогичным образом это продолжается дальше. Зачеркнутые ветви отбрасываются.

Таким образом, если процесс появления чисел $k, l, m, \ldots$ оборвется на каком-то шаге, то класс U_i будет определен и будет состоять из двух функций, по крайней мере одну из которых стратегия F_i' не синтезирует в смысле GN.

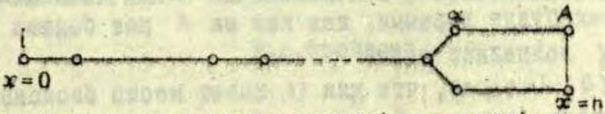
Если же таких чисел $k, l, m, \ldots$ найдется бесконечно много, то наше дерево "вырастет" таким:

где последовательность $i\alpha 0^k \beta 0^l \gamma 0^m \delta \ldots$ рекурсивна. Тогда мы положим $U_i = \{ i\alpha 0^k \beta 0^l \gamma 0^m \delta \ldots \}$. Ясно, что эту единственную функцию класса U_i стратегия F_i' не синтезирует в смысле GN (F_i' на ней бесконечно много раз меняет гипотезу).

Итак, U_i определен для всех i. Если теперь взять $U = \bigcup_{i=0}^{\infty} U_i$, то автоматически $U \notin GN$ (если бы о.р. стратегия F синтезировала U в смысле GN, то это же должна делать и стратегия F_{i_0}', где $F = F_{i_0}$, однако F_{i_0}' не синтезирует даже U_{i_0}).

Остается показать, что $U \in GN^{\infty}$. Мы должны построить о.р. стратегию H, которая на каждой $\varphi \in U$ выдает последовательность гипотез, которые, начиная с некоторого места, все верны.

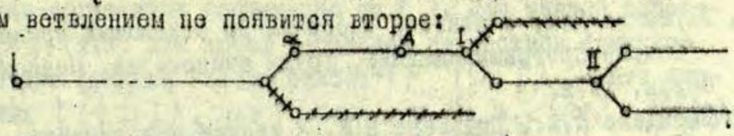
Если даны значения функции φ, то вместе с $i = \varphi(0)$ мы знаем стратегию F_i' и можем фиксировать некоторую процедуру порождения дерева функций класса U_i (но мы не знаем заранее, какой из случаев нам встретится - конечный или бесконечный). Чтобы определить гипотезу $H(<\varphi(0), \ldots, \varphi(n)>)$, дерево U_i $(i = \varphi(0))$ "выращивается" сначала до "высоты" большей n:



Если начальный кусок $\varphi(0),\ldots,\varphi(n)$ "отклоняется" от этого дерева в сторону, то $\varphi\bar{\in} U$, т.е. можно положить, например, $H(<\varphi(0),\ldots,\varphi(n)>)=0$. Допустим теперь, что $\varphi(0),\ldots,\varphi(n)$ не отклоняется, например, совпадает с ветвью iA, тогда в качестве гипотезы $H(<\varphi(0),\ldots,\varphi(n)>)$ берется геделев номер следующей функции η.

Сначала η совпадает с ветвью iA, чтобы определить η за точкой A, дерево U_i "выращивается" дальше. Пока ветвь αA "растет" без ветвлений, функция η следует по ней. Если ветвь αA со временем отбрасывается, то η остается неопределенной для дальнейших значений аргумента.

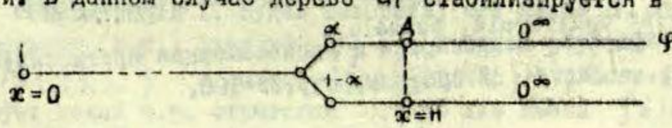
Если же на продолжении ветви αA встретилось ветвление, то η не вычисляется до тех пор, пока за этим первым ветвлением не появится второе:



Тогда значения η определяются по ветви $i\alpha A\,I\,II$ (если II так и не появится, то η останется функцией с конечной областью определения). Но за точкой II значения η не определяются, пока не появится третье ветвление; в этом случае дальше η пойдет от II к III, и так далее.

Стратегия H определена. Проверим, что она действительно синтезирует в смысле GN^{∞} любую функцию φ класса U.

а) $\varphi(0)=i$ такое, что для U_i имеет место конечный случай. В данном случае дерево U_i стабилизируется в виде:



Но тогда гипотезы стратегии H в точке A и во всех дальнейших точках будут верными, так как за A нет больше ветвлений, и η совпадает с φ.

б) $\varphi(0)=i$ такое, что для U_i имеет место бесконечный случай. В данном случае φ совпадает с единственной "выживающей" ветвью дерева U_i, причем в процессе построения этого дерева за каждым ветвлением следует новое ветвление. Поэтому уже с самого начала η совпадает с φ, т.е. все гипотезы стратегии H на φ верны.

Теорема доказана.

В заключение отметим, что аналог теоремы 2 для предельной идентификации языков получается значительно проще (см.[5]).

ЛИТЕРАТУРА

1. Подниекс К.М. Сравнение различных типов предельного синтеза и прогнозирования функций. - Настоящий сборник, стр.68-81.
2. Барздинь Я.М. О синтезе программ по отдельным примерам. - Теория программирования, Труды симпозиума, Новосибирск, 1972.
3. Барздинь Я.М., Подниекс К.М. К теории индуктивного вывода. - Труды симпозиума "Mathematical foundations of computer science", High Tatras, Czechoslovakia, 1973.
4. Gold E.M. Limiting recursion. "The Journal of Symbolic Logic", 1965, 30, No.1.
5. Feldman J. Some decidability results on grammatical inference and complexity. - "Information and Control", 1972, 20, No.3.
6. Blum L., Blum M. Inductive inference: a recursion theoretic approach. Memorandum No.ERL-M386, March 1973, Univ.of California, Berkeley.
7. Фрейвалд Р.В. Равномерная и неравномерная прогнозируемость. - Настоящий сборник, стр.89-100.

РАВНОМЕРНАЯ И НЕРАВНОМЕРНАЯ ПРОГНОЗИРУЕМОСТЬ

Р.В.Фрейвалд

Пусть зафиксированы геделевская нумерация всех частично рекурсивных (ч.р.) функций $\{\varphi_n\}$ и эффективная канторовская нумерация $<x_1,\dots,x_n>$ всех конечных кортежей натуральных чисел.

В [I] рассматривались, в частности, задачи прогнозирования следующего значения (для любого момента t по значениям аргументов $\{x_1,\dots,x_t,x_{t+1}\}$ и значениям функции $\{f(x_1),\dots,f(x_t)\}$ требуется определить следующее значение функции $f(x_{t+1})$) и прогнозирования геделевского номера (по значениям аргументов $\{x_1,\dots,x_t\}$ и значениям функции $\{f(x_1),\dots,f(x_t)\}$ требуется определить геделевский номер функции f или другой функции, которая на элементах всей бесконечной последовательности $\Omega=\{x_1,\dots,x_t,\dots\}$ значений аргументов неотличима от f). Для каждой стратегии $F(x,y)$ (т.е. для любой частичной двуместной функции) введем две функции "числа ошибок": $F^{NV}(\Omega,f)$ - это число таких t $(t\geq 1)$, что $F(<x_1,\dots,x_t,x_{t+1}>,<f(x_1),\dots,f(x_t)>)\neq$ $\neq f(x_{t+1})$, если $\Omega=\{x_1,x_2,\dots\}$; $F^{GN}(\Omega,f)$ - это число таких t $(t\geq 1)$, что $F(<x_1,\dots,x_{t+1}>,<f(x_1),\dots,f(x_{t+1})>)\neq$ $\neq F(<x_1,\dots,x_t>,<f(x_1),\dots,f(x_t)>)$ при условии, что $\lim\limits_t F(<x_1,\dots x_t>,$ $<f(x_1),\dots,f(x_t)>)$ равен геделевскому номеру функции, совпадающей на последовательности Ω с f. Если это условие не выполнено, а также если при каком-то t $F(<x_1,\dots,x_t>,<f(x_1),\dots,f(x_t)>)$ не определено, то $F^{GN}(\Omega,f)=\infty$. Аналогично, если при каком-то t не определено $F(<x_1,\dots,x_t,x_{t+1}>,<f(x_1),\dots,f(x_t)>)$, то $F^{NV}(\Omega,f)=\infty$.

ОПРЕДЕЛЕНИЕ I. Будем говорить, что класс U общерекурсивных (о.р.) функций п р о г н о з и р у е м н а $\Omega_0=\{0,1,2,3,\dots\}$ в смысле $\lambda(\lambda\in\{GN,NV\})$, если существует такая ч.р. стратегия , что для любой $f\in U$ $F^{\lambda}(\Omega_0,f)<\infty$.

ОПРЕДЕЛЕНИЕ 2. Будем говорить, что класс U о.р.

функций прогнозируем на всех выборках в смысле $\lambda(\lambda\in\{GN,NV\})$, если для любой последовательности Ω натуральных чисел существует такая ч.р. стратегия F, что для любой $f\in U$ $F^{\lambda}(\Omega,f)<\infty$.

ОПРЕДЕЛЕНИЕ 3. Будем говорить, что класс U о.р. функций прогнозируем равномерно на всех бесконечных выборках в смысле λ $(\lambda\in\{GN,NV\})$, если существует такая ч.р. стратегия F, что для любой последовательности Ω, содержащей бесконечно много различных натуральных чисел и для любой $f\in U$ $F^{\lambda}(\Omega,f)<\infty$.

ОПРЕДЕЛЕНИЕ 4. Будем говорить, что класс U о.р. функций прогнозируем равномерно на всех выборках в смысле $\lambda(\lambda\in\{GN,NV\})$, если существует такая ч.р. стратегия F, что для любой последовательности Ω натуральных чисел и для любой $f\in U$ $F^{\lambda}(\Omega,f)<\infty$.

Легко убедиться, что при любом $\lambda\in\{GN,NV\}$ из прогнозируемости в смысле λ по какому-то одному из указанных четырех определений вытекает прогнозируемость в смысле λ по всем предшествующим определениям. Например, из прогнозируемости, равномерной на всех бесконечных выборках, следует (неравномерная) прогнозируемость на всех выборках. Дело в том, что существуют стратегии F_1 и F_2, прогнозирующие в смысле GN и, соответственно, NV, любые классы о.р. функций на любых последовательностях, состоящих только из конечного числа различных элементов.

Покажем, что обратные импликации в общем случае не имеют места.

Будем говорить, что класс U прогнозируем без ошибок по какому-то из указанных определений, если в этом определении $F^{\lambda}(\Omega,f)<\infty$ для класса U можно заменить на $F^{\lambda}(\Omega,f)=0$.

ПРЕДЛОЖЕНИЕ I. Существует класс U о.р. функций, прогнозируемый на Ω без ошибок как в смысле GN, так и в смысле NV, но непрогнозируемый на всех выборках ни в смысле GN, ни в смысле NV.

ДОКАЗАТЕЛЬСТВО. Пусть $\{l_1, l_2, \ldots\}$ - множество всех геделевских номеров всех о.р. функций. Определим U как совокупность всех функций такого вида:

$$g(x) = \begin{cases} l_j & , \text{если } x = 0 \\ \varphi_{l_j}(x-1) & , \text{если } x > 0 . \end{cases}$$

Если прогнозирование производится на Ω_0, то, имея значение функции на нуле, т.е. l_j, нетрудно вычислить как некоторый геделевский номер функции $g(x)$, так и все требуемые значения этой функции. Если же прогнозирование производится на $\Omega = \{1,2,3,\ldots\}$, то оно эквивалентно прогнозированию класса всех о.р. функций на Ω, а еще Э.М.Голд [2] доказал непрогнозируемость этого класса. Предложение доказано.

Далее будем доказывать, что из прогнозируемости на всех выборках еще не следует равномерная прогнозируемость даже на всех бесконечных выборках.

Через $c(x,y)$, $l(x)$, $r(x)$ будем обозначать канторовские нумерационные функции, устанавливающие взаимнооднозначное соответствие между $N \times N$ и N.

ЛЕММА I. Каковы бы ни были о.р. функция $f(x)$ и натуральное число n, существует такое натуральное число i, что ч.р. функция $\varphi_i(x)$ с геделевским номером i для $x > n$ равна $f(x)$, а для $x \leq n$ равна $c(i, f(x))$.

ДОКАЗАТЕЛЬСТВО. Сопоставим каждому натуральному j геделевский номер $g(j)$ следующей функции

$$\varphi_{g(j)}(x) = \begin{cases} c(j, f(x)), & \text{если } x \leq n \\ f(x), & \text{если } x > n . \end{cases}$$

Из теоремы о неподвижной точке [3] следует существование такого i, что $\varphi_i \equiv \varphi_{g(i)}$.

ЛЕММА 2. (Следствие более общей теоремы Я.М.Барздиня [I]). Пусть U - произвольный класс о.р. функций. Если существует ч.р. стратегия F, прогнозирующая U в смысле NV на всех последовательностях типа $\{k, k+1, k+2, k+3, \ldots\}$, то существует ч.р. стратегия F', прогнозирующая U на этих же последовательностях в смысле GN.

ДОКАЗАТЕЛЬСТВО. $F'(<k>,<f(k)>)$ равна геделевскому номеру следующей функции

$$g(x)=\begin{cases} f(k) & \text{если } x=k \\ F(<k,k+1,\ldots,x>,<g(k),g(k+1),\ldots,g(x-1)>) & \text{если } x>k. \end{cases}$$

Заметим, что $g(x)$ определена для всех $x \geq k$.
$F'(<k,k+1>,<f(k),f(k+1)>), F'(<k,k+1,k+2>,<f(k),f(k+1),f(k+2)>)$ и дальнейшие значения определяются так. Пока $f(k+1)=g(k+1)$, $f(k+2)=g(k+2), \ldots,$ значение функции F' не меняется. Если обнаружилось такое s, что $f(s) \neq g(s)$, то $F'(<k,k+1,\ldots,s>,<f(k),f(k+1),\ldots,f(s)>)$ равно геделевскому номеру функции

$$g'(x)=\begin{cases} f(x), & \text{если } k \leq x \leq s \\ F(<k,k+1,\ldots,x>,<g'(k),g'(k+1),\ldots,g'(x-1)>) & \text{если } x>s. \end{cases}$$

ТЕОРЕМА I. Существует класс U о.р. функций, прогнозируемый на всех выборках без ошибок как в смысле GN, так и в смысле NV, но не прогнозируемый равномерно даже на всех бесконечных выборках ни в смысле GN, ни в смысле NV.

ДОКАЗАТЕЛЬСТВО. Обозначим через $g^{(f,n)}$ функцию с минимальным геделевским номером, соответствующую в смысле леммы I данной о.р. функции f и числу n. Класс U будет составлен из ряда таких функций $g^{(f,n)}$ (но не всевозможных), с таким расчетом, чтобы

1) $g^{(f_1,n_1)} \in U \ \& \ g^{(f_2,n_2)} \in U \ \& \ n_1 = n_2 \rightarrow f_1 \equiv f_2$,

2) $g^{(f_1,n_1)} \in U \ \& \ g^{(f_2,n_2)} \in U \ \& \ \exists x_1 \exists x_2 \, g^{(f_1,n_1)}(x_1) = g^{(f_2,n_2)}(x_2) \rightarrow$
$\rightarrow n_1 = n_2 \ \& \ f_1 \equiv f_2$.

Заметим, что из 1) и 2) уже следует прогнозируемость класса U без ошибок на всех выборках как в смысле GN, так и в смысле NV. Действительно, пусть дана последовательность Ω. Обозначим первое число в этой последовательности через m. Из 1) следует, что все, кроме конечного числа, функции класса U можно представить как $g^{(f,n)}$, где $n > m$. Следовательно, прогнозирование без ошибок в смысле GN на Ω может осуществляться следующим алгоритмом Σ. Алгоритм Σ имеет в памяти (конечный) список геделевских номеров всех таких функций $g^{(f,n)} \in U$, что $n \leq m$. Из 2) вытекает, что, имея значение некоторой функции $h \in U$ в точке m, можно определить, принадлежит ли функция h указанному списку. Если функция принадлежит списку, то ее номер (из-за свойства 2)) определяется однозначно. Если функция не принадлежит списку, то ее геделевским номером является $l(h(m))$. Из прогнозируемости без ошибок в смысле GN легко вытекает и прогнозируемость без ошибок в смысле NV.

Далее проведем построение класса U с таким расчетом, чтобы U не был прогнозируем равномерно на всех бесконечных выборках и в то же время выполнялись бы свойства 1) и 2). Выше было показано, что тогда, сколь бы неэффективно было построение класса U, все нужные свойства имеют место.

Пусть $\Phi_2 = \{\varphi_0^{(2)}, \varphi_1^{(2)}, \varphi_2^{(2)}, \dots\}$ - совокупность всех двуместных ч.р. функций. Для каждой $\varphi \in \Phi_2$ класс U будет содержать о.р. функцию, которую стратегия φ на подходящей Ω прогнозирует в смысле GN неправильно.

По индукции допустим, что для $\varphi_0^{(2)}, \varphi_1^{(2)}, \dots, \varphi_{k-1}^{(2)}$

такие функции уже построены и определено очередное "большое" простое число p_k. (При $k=0$ полагаем $p_k=2$). Тогда для $\varphi_k^{(2)}$ построение соответствующей функции и числа p_{k+1} происходит следующим образом.

При работе стратегии $\varphi_k^{(2)}$ на $\Omega=\{k+1,k+2,k+3,\ldots\}$ априорно возможны 3 ситуации, которые описаны ниже. В каждой из этих ситуаций будет определена о.р. функция f. Тогда в класс U включается функция $g^{(f,k)}$, а в качестве p_{k+1} берется наименьшее простое число, которое больше чем $max\{p_k, g^{(f,k)}(0)\}$.

Первая ситуация характерна тем, что существует такой кортеж натуральных чисел $\{y_{k+1},y_{k+2},\ldots,y_{k+r}\}$, состоящий из целых положительных степеней числа p_k, что $\varphi_k^{(2)}(<k+1,\ldots,k+r>,<y_{k+1},\ldots,y_{k+r}>)$ не определена. В этом случае

$$f(x)=\begin{cases} p_k, \text{ если } x\leq k \\ y_x, \text{ если } k+1\leq x\leq k+r \\ p_k, \text{ если } x\geq k+r+1 \end{cases}$$

Вторая ситуация определяется тем, что первая ситуация не имеет места, но для любого кортежа $\{y_{k+1},\ldots,y_{k+r}\}$ целых положительных степеней числа p_k существует такой кортеж $\{y_{k+r+1},\ldots,y_{k+s}\}$ целых положительных степеней числа p_k, что $\varphi_k^{(2)}(<k+1,\ldots,k+s>,<y_{k+1},\ldots,y_{k+s}>)\neq$ $\neq\varphi_k^{(2)}(<k+1,\ldots,k+r>,<y_{k+1},\ldots,y_{k+r}>)$. В этом случае определяем $f(x)=p_k$ для всех $x\leq k+1$ и дальнейшие значения функции f определяем следующей рекурсивной процедурой.

Вычисляем $\varphi_k^{(2)}(<k+1>,<p_k>)$ и перебором в естественном порядке всех кортежей целых положительных степеней числа p_k находим первый такой кортеж $\{y_{k+2},\ldots,y_{k+q}\}$ что $\varphi_k^{(2)}(<k+1,k+2,\ldots,k+q>,<p_k,y_{k+2},\ldots,y_{k+q}>)\neq\varphi_k^{(2)}(<k+1>,<p_k>)$. Тогда определяем $f(k+2)=y_{k+2},\ldots,f(k+q)=y_{k+q}$, приступаем к поиску такого кортежа $\{y_{k+q+1},\ldots,y_{k+q'}\}$ целых положительных степеней числа p_k, что $\varphi_k^{(2)}(<k+1,\ldots,k+q'>,<p_k,y_{k+2},\ldots,y_{k+q'}>)\neq$ $\neq\varphi_k^{(2)}(<k+1,\ldots,k+q>,<p_k,y_{k+2},\ldots,y_{k+q}>)$ и т.д.

Определение второй ситуации гарантирует, что f окажется всюду определенной, следовательно, общерекурсивной.

Третья ситуация определяется тем, что первая и вторая ситуации не имеют места. Следовательно, существует такой кортеж $\{y_{k+1},\dots,y_{k+t}\}$ целых положительных степеней числа p_k, что для любого кортежа $\{y_{k+t+1},\dots,y_s\}$ целых положительных степеней числа p_k $\varphi_k^{(2)}(<k+1,\dots,k+s>,<p_k,y_{k+2},\dots,y_{k+s}>)=\varphi_k^{(2)}(<k+1,\dots,k+t>,<p_k,y_{k+2},\dots,y_{k+t}>)$. В этом случае хотя бы одна из функций

$$f_1(x)=\begin{cases}p_k, \text{если } x\le k\\ y_x, \text{если } k+1\le x\le k+t\\ p_k, \text{если } x>k+t\end{cases}\qquad f_2(x)=\begin{cases}p_k, \text{если } x\le k\\ y_x, \text{если } k+1\le x\le k+t\\ p_k^2, \text{если } x>k+t\end{cases}$$

прогнозируется стратегией $\varphi_k^{(2)}$ в смысле GN на $\Omega=\{k+1,k+2,\dots\}$ неправильно. Ее и обозначаем через $f(x)$.

Проверка свойств 1) и 2) не представляет труда.

Итак, для любой $\varphi_k^{(2)}\in\Phi_2$ существует последовательность $\Omega=\{k+1,k+2,\dots\}$ и существует функция $g\in U$, что $\varphi_k^{(2)}$ на Ω прогнозирует g в смысле GN неправильно. Но тогда по лемме 2 класс U не прогнозируем равномерно на всех бесконечных выборках и в смысле NV. Теорема доказана.

Переходим к выяснению отношения между прогнозируемостью, равномерной на всех бесконечных выборках и прогнозируемостью, равномерной на всех выборках.

ПРЕДЛОЖЕНИЕ 2. Класс о.р. функций прогнозируем в смысле NV равномерно на всех бесконечных выборках тогда и только тогда, когда он прогнозируем в смысле NV равномерно на всех выборках.

ДОКАЗАТЕЛЬСТВО. ДОСТАТОЧНОСТЬ очевидна.

НЕОБХОДИМОСТЬ. Пусть класс U прогнозируется в смысле NV равномерно на всех бесконечных выборках стратегией F. Тогда для прогнозирования, равномерного на всех выборках, подходит следующая стратегия F'. Для выдачи значения $F'(<x_1,\ldots,x_t,x_{t+1}>,<f(x_1),\ldots,f(x_t)>)$ сначала проверяется, не совпадает ли x_{t+1} с каким-то из $x_1,\ldots,x_t$. Если совпадает, то значение $f(x_{t+1})$ извлекается из значения второго аргумента F'. Если не совпадает, то значение функции F' совпадает со значением функции F при тех же значениях аргументов.

ЛЕММА 3. Каковы бы ни были о.р. функция $f(x)$ и кортеж натуральных чисел $\{y_0,y_1,\ldots,y_n\}$, существует такое натуральное число i, что ч.р. функция $\varphi_i(x)$ с геделевским номером i для $x>n$ равна $c(i,f(x))$, а для $x\leqslant n$ равна y_x.

ДОКАЗАТЕЛЬСТВО. Сопоставим каждому натуральному j геделевский номер $g(j)$ следующей функции

$$\varphi_{g(j)}(x)=\begin{cases} y_x, & \text{если } x\leqslant n \\ c(j,f(x)), & \text{если } x>n. \end{cases}$$

Из теоремы о неподвижной точке [3] следует существование такого i, что $\varphi_i\equiv\varphi_{g(i)}$.

ЛЕММА 4.*) Пусть класс U о.р. функций таков, что для любого кортежа натуральных чисел $\{y_0,y_1,\ldots,y_n\}$ класс U содержит функцию f, принимающую значения $f(0)=y_0, f(1)=y_1,\ldots,f(n)=y_n$. Если U прогнозируем в смысле NV на Ω, то U содержится в эффективно перечислимом классе о.р. функций (т.е. в классе, имеющем о.р. универсальную функцию).

---

*) Это утверждение было сформулировано Я.М.Барздинем [I] в более слабом виде. М.И.Аугустон впервые заметил, что из доказательства Я.М.Барздиня на самом деле вытекает утверждение леммы 4.

ДОКАЗАТЕЛЬСТВО. Пусть стратегия F прогнозирует класс U в смысле NV на Ω_0. Тогда U содержится в классе U', имеющем универсальную функцию

$$g(l,x)=\begin{cases} y_x, & \text{если } x\leq n,\\ F(<0,1,\ldots,x-1,x>,<g(l,0),g(l,1),\ldots,g(l,x-1)>), & \text{если } x>n \end{cases}$$

где $\{y_0,y_1,\ldots,y_n\}$ - кортеж с канторовым номером l.

ТЕОРЕМА 2. Существует класс U о.р. функций, прогнозируемый равномерно на всех бесконечных выборках в смысле GN, но не прогнозируемый равномерно на всех выборках ни в смысле GN, ни в смысле NV.

ДОКАЗАТЕЛЬСТВО. Класс U состоит из всевозможных функций $\varphi_l(x)$, которые соответствуют в смысле леммы 3 какой-либо о.р. функции f и какому-либо кортежу натуральных чисел $\{y_0,y_1,\ldots,y_n\}$. Таким образом, по лемме 3 для любого кортежа $\{y_0,y_1,\ldots,y_n\}$ класс U содержит функцию ψ, принимающую значения $\psi(0)=y_0,\psi(1)=y_1,\ldots,\psi(n)=y_n$.

1. Класс U прогнозируем равномерно на всех бесконечных выборках в смысле GN следующей стратегией F_1:

$$F_1(<x_1,\ldots,x_t>,<f(x_1),\ldots,f(x_t)>)=l(f(\max\{x_1,\ldots,x_t\})).$$

2. Допустим от противного, что класс U прогнозируем в смысле NV на Ω_0. Тогда по лемме 4 U содержится в некотором эффективно перечислимом классе U' о.р. функций. Рассмотрим класс $U''=\{r(f)\mid f\in U'\}$. С одной стороны, U'' также эффективно перечислимый класс о.р. функций. С другой стороны, U'' - класс всех о.р. функций. Противоречие.

Тем более U не является прогнозируемым в смысле NV равномерно на всех выборках.

3. Допустим от противного, что класс U прогнозируем равномерно на всех выборках в смысле GN стратегией F.

Опишем о.р. функцию $h(l)$, сопоставляющую каждому натуральному i геделевский номер некоторой о.р. функции

$g(x)$, принимающей только два значения $c(i,0)$ и $c(i,1)$.

Полагаем $g(0)=c(i,0)$. Заметим, что из-за свойства, отмеченного при определении класса U, $F(<0>,<c(i,0)>)$ должна быть определена. Должны быть определены также все другие рассматриваемые ниже значения функции F. Для определения $g(1)$ попеременно рассматриваются последовательности значений стратегии F на $\Omega=\{0,1,1,1,1,\ldots\}$ для двух функций, принимающих, соответственно, значения $f_0(0)=g(0)$, $f_0(1)=c(i,0)$ и $f_1(0)=g(0)$, $f_1(1)=c(i,1)$.

$F(<0,1>,<g(0),c(i,0)>)$	$F(<0,1>,<g(0),c(i,1)>)$
$F(<0,1,1>,<g(0),c(i,0),c(i,0)>)$	$F(<0,1,1>,<g(0),c(i,1),c(i,1)>)$
$F(<0,1,1,1>,<g(0),c(i,0),c(i,0),c(i,0)>)$	$F(<0,1,1,1>,<g(0),c(i,1),c(i,1),c(i,1)>)$

Так как U содержит такие функции f_1 и f_2, то указанные последовательности значений функции F должны стабилизироваться на р а з н ы х числах - геделевских номерах функций, применяющих значения $f_0(0)=g(0)$, $f_0(1)=c(i,0)$ и, соответственно, $f_1(0)=g(0)$, $f_1(1)=c(i,1)$. Поэтому эффективно можно найти то из двух значений $c(i,0)$, $c(i,1)$, для которого в соответствующей последовательности первое значение, отличное от $F(<0>,<g(0)>)$, появляется раньше. Значение $g(1)$ полагаем равным найденному таким образом значению, а наименьшее t, при котором $F(<0,\underbrace{1,1,\ldots,1}_{t}>,<g(0),\underbrace{g(1),g(1),\ldots,g(1)}_{t}>)\neq F(<0>,<g(0)>)$, обозначим через t_1.

Для определения $g(2)$ попеременно рассматриваются последовательности значений стратегии F на $\Omega=\{0,\underline{1,1,\ldots,1},2,2,2,\ldots\}$ для двух функций, принимающих, соответственно, значения $f_0(0)=g(0)$, $f_0(1)=g(1)$, $f_0(2)=c(i,0)$; $f_1(0)=g(0)$, $f_1(1)=g(1)$, $f_1(2)=c(i,1)$. $g(2)$ мы определяем как то значение, при котором последовательность значений стратегии быстрее меняет значение, и t_2 - как наименьшее число повторений значения 2 в рассматриваемой последовательности Ω, необходимое, чтобы это изменение наступило. Потом аналогично определяются $g(3), t_3, g(4), t_4$ и т.д.

Так как стратегия F прогнозирует U равномерно на всех выборках, то все значения функции $g(x)$ будут определены. Следовательно, при любом i $h(i)$ является номером о.р. функции. Поэтому и $h(i_0)$, где i_0 - неподвижная точка функции h, тоже номер о.р. функции. Функция $\varphi_{h(i_0)}$ принимает только два различных значения $c(i_0,0)$ и $c(i_0,1)$. Поэтому $\varphi_{h(i_0)} \in$ $\in U$. Однако стратегии при прогнозировании $\varphi_{h(i_0)}$ на $\Omega = \{\underbrace{0,1,1,1,\dots,1}_{t_1},\underbrace{2,2,2,\dots,2}_{t_2},\dots\}$ бесконечно много раз меняет значение! Противоречие. Теорема доказана.

Заметим, что теорему 2 нельзя усилить (подобно предложению I и теореме I) таким образом, чтобы равномерная прогнозируемость на всех бесконечных выборках имела место без ошибок. Дело в том, что справедливо

ПРЕДЛОЖЕНИЕ 3. Если класс U о.р. функций прогнозируем в смысле GN равномерно на всех бесконечных выборках без ошибок, то U прогнозируем равномерно на всех выборках как в смысле GN, так и в смысле NV.

ДОКАЗАТЕЛЬСТВО очевидно.

ЗАМЕЧАНИЕ. Если некоторый класс U прогнозируем в смысле GN равномерно на всех бесконечных выборках, то этот класс прогнозируем равномерно на всех выборках в смысле GN^{∞}. (Определение GN^{∞} см. [4]). Таким образом, класс функций, построенный в теореме 2, дает также пример класса, который в смысле GN^{∞} прогнозируем равномерно на всех выборках, но в смысле GN не прогнозируем равномерно на всех выборках. Следует, однако, добавить, что первый пример класса с этим свойством был построен Я.М.Барздинем даже еще раньше построения опубликованного им в [4] примера класса, прогнозируемого в смысле GN^{∞} на Ω и не прогнозируемого в смысле GN на Ω_0.

В заключение автор желает выразить искренною благодарность К.М.Подниексу за полезную идею и Я.М.Барздиню за ценное обсуждение одного построения, в результате которого выяснилась естественность понятия равномерности на всех бесконечных выборках.

ЛИТЕРАТУРА

I. Барздинь Я.М., Фрейвалд Р.В. О прогнозировании общерекурсивных функций - "ДАН СССР", 1972, 206, № 3.
2. Gold E.M. Language identification in the limit - "Information and Control", 1967, 10.
3. Роджерс Х. Теория рекурсивных функций и эффективная вычислимость, М., 1972.
4. Барздинь Я.М. Две теоремы о предельном синтезе.- Настоящий сборник,стр.82-88.

ПРОГНОЗИРОВАНИЕ И ПРЕДЕЛЬНЫЙ СИНТЕЗ ЭФФЕКТИВНО ПЕРЕЧИСЛИМЫХ КЛАССОВ ФУНКЦИЙ *)

Я.М.Барздинь, Р.В.Фрейвалд

Сначала приведем некоторые обозначения, позаимствованные из [3]. φ - фиксированная геделевская нумерация всех I-местных частично рекурсивных функций (ч.р.ф.). R - класс всех I-местных общерекурсивных функций (о.р.ф.). U всегда обозначает некоторый класс о.р.ф. (т.е. $U \subseteq R$). В дальнейшем рассматриваются только э ф ф е к т и в н о п е р е ч и с л и м ы е к л а с с ы U (т.е. обладающие вычислимой нумерацией τ : $U = \{\tau_1, \tau_2, \dots, \tau_n, \dots\}$, $\tau_n(x)$ - о.р.ф. от n, x). В дальнейшем, когда говорится "н у м е р о в а н н ы й к л а с с (U,τ) ", имеется в виду эффективно перечислимый класс U вместе с некоторой его вычислимой нумерацией τ .

С т р а т е г и е й называется любая всюду определенная функция типа $N \to N$. В дальнейшем рассматриваются только о.р. стратегии.

Через $\Omega = \{x_1, x_2, \dots, x_m, \dots\}$ мы будем обозначать произвольную последовательность натуральных чисел. $\Omega_0 = \{1, 2, \dots, m, \dots\}$ - естественная последовательность.

Функции мы будем представлять как последовательности значений $f(0), f(1), \dots, f(x), \dots$. Например, $\tilde{\alpha}\, 1^2 0^\infty$ (где $\tilde{\alpha}$ - кортеж чисел $y_0, y_1, \dots, y_s$) - это следующая функция: $y_0, y_1, \dots, y_s, 1, 1, 0, 0, 0, \dots$.

$\langle x_1, \dots, x_n \rangle$ - эффективная нумерация всех конечных кортежей натуральных чисел.

П р о г н о з и р о в а н и е. Если F - стратегия, φ - всюду определенная функция, $\Omega = \{x_1, x_2, \dots\}$ - последовательность натуральных чисел, то значения

---

*) Основные результаты этой статьи были сформулированы без доказательств в [I] и [2].

$$F(<x_1, \varphi(x_1), x_2, \varphi(x_2), \ldots, x_m, \varphi(x_m), x_{m+1}>)\ ^{*)}$$

называются п р о г н о з а м и. Прогноз считается верным, если он равен $\varphi(x_{m+1})$, в противном случае прогноз считается ошибочным.

Для характеристики числа ошибок вводится функционал $F^{NV}(\Omega, \varphi)$, равный числу ошибок, которые стратегия F допускает при прогнозировании φ на последовательности Ω ($F^{NV}(\Omega,\varphi) = |\{m \mid F(<x_1, \varphi(x_1), \ldots, x_m, \varphi(x_m), x_{m+1}>) \neq \varphi(x_{m+1})\}|$).

Для изучения прогнозирования функций нумерованного класса (U,τ) вводится функция ошибок:

$$F^{NV}_{U,\tau}(\Omega, n) = \max_{1 \leq i \leq n} F^{NV}(\Omega, \tau_i).$$

Для каждого (U,τ) можно построить о.р. стратегию F, которая перебирает все функции τ_i по порядку, и, таким образом, для любого n имеем $F^{NV}_{U,\tau}(\Omega, n) \leq n-1$. Следующие теоремы (I и 2) показывают, что асимптотически точной оценкой функции ошибок является $\log_2 n$.

ТЕОРЕМА I. Для каждого нумерованного класса (U,τ) можно построить о.р. стратегию F такую, что для всех последовательностей Ω и всех n

$$F^{NV}_{U,\tau}(\Omega, n) \leq \log_2 n + \log_2 \log_2 n + o(\log_2 \log_2 n).$$

ДОКАЗАТЕЛЬСТВО. Основная идея состоит в том, что каждому τ-номеру i присваивается некоторая вероятность p_i $(\sum p_i = 1)$, а затем для определения прогноза

$$y_{m+1} = F(<x_1, y_1, x_2, y_2, \ldots, x_m, y_m, x_{m+1}>)$$

вычисляется для каждого s вероятность того, что произвольно выбранная функция φ из $\{\tau_1, \tau_2, \ldots, \tau_n, \ldots\}$ обладает свойством

$$\varphi(x_1) = y_1 \,\&\, \varphi(x_2) = y_2 \,\&\, \ldots \,\&\, \varphi(x_m) = y_m \,\&\, \varphi(x_{m+1}) = s.$$

---

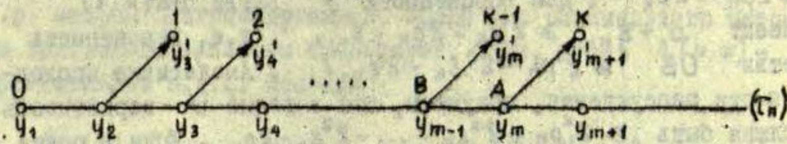
*) Предполагается, что $m = 0, 1, 2, \ldots$; при $m = 0$ имеем $F(<x_1>)$.

В качестве y'_{m+1} берется число λ, для которого эта вероятность наибольшая.

Если при таком прогнозировании мы допустим на функции τ_n (при $\Omega=\{x_1,\dots,x_m,\dots\}$, $y_m=\tau_n(x_m)$) всего к ошибок, то это даст сразу неравенство

$$2^k p_n \leq 1. \tag{I}$$

В самом деле, отобразим ситуацию на следующем "дереве":



Бесконечная ветвь состоит из всех тех функций класса U, которые на Ω совпадают с τ_n. Стрелки в сторону обозначают неверные прогнозы (т.е. $y'_m \neq y_m$).

Вероятность, приписанная всей бесконечной ветви (τ_n), не меньше p_n. k -ая ошибка совершается в "сторону большей вероятности", следовательно, конечной ветви OA должна быть приписана вероятность $\geq p_n+p_n=2p_n$. Аналогично для ветви OB имеем вероятность $\geq 2p_n+2p_n=2^2p_n$. Продолжая эти рассуждения, мы получаем, что в точке O вероятность должна быть $\geq 2^k p_n$. Но в то же время эта вероятность ≤ 1. Следовательно, $2^k p_n \leq 1$.

Отсюда получаем, что рассматриваемая стратегия F допускает на функции τ_n не более $\log_2 \frac{1}{p_n}$ ошибок. Если теперь в определении F взять $p_n=\frac{c}{n(\log_2 n)(\log_2\log_2 n)^2}$ (где c подобрано для обеспечения $\sum p_n=1$), то получим оценку

$$F^{NV}_{U,\tau}(\Omega,n)\leq \log_2 n+\log_2\log_2 n+o(\log_2\log_2 n). \tag{2}$$

Здесь, однако, имеется некорректность. Мы не можем гарантировать рекурсивность такой стратегии F, так как в процессе своей работы она требует абсолютно точного вычисления соответствующих вероятностей. Поэтому для "конструктивизации" стратегии F все нужные вероятности для определения j -го прогноза мы будем вычислять лишь с точностью до некоторого ε_j. (Это возможно, если последовательность $\{p_n\}$ рекурсивна и ряд $\sum p_n$ конструктивно сходится к I). А именно, возвращаясь опять к "дереву", мы видим, что вероятность p ветви OA_K должна удовлетворять неравенству $p+\varepsilon_K \geq p_n - \varepsilon_K$, т.е. вероятность ветви OA $\geq 2p_n - 2\varepsilon_K$. Для вероятности p' ветви $OB(K-1)$ имеем: $p'+\varepsilon_{K-1} \geq 2p_n - 2\varepsilon_K - \varepsilon_{K-1}$, т.е. вероятность ветви OB $\geq 2^2 p_n - 2^2\varepsilon_K - 2\varepsilon_{K-1}$. Аналогично продолжая эти рассуждения, получаем, что в точке O вероятность должна быть $\geq 2^K p_n - 2^K \varepsilon_K - \ldots - 2^2\varepsilon_2 - 2\varepsilon_1$. Если в определении стратегии F взять $\varepsilon_j = 2^{-2j}$, то вместо (I) мы получим неравенство $2^K p_n \leq 2$. Но это не влияет на неравенство (2).

Теорема доказана.

Можно рассматривать также и прогнозирование нерекурсивных функций φ. В этом случае, очевидно, $F^{NV}(\Omega_0, \varphi) = \infty$. Поэтому мы будем рассматривать только конечные начальные куски функции:

$$\varphi^n = \{\varphi(1), \ldots, \varphi(n)\},\ \Omega\varphi^n = \{\varphi(x_1), \ldots, \varphi(x_n)\}$$
$$(\Omega_0 \varphi^n = \varphi^n)$$

Под $F^{NV}(\Omega\varphi^n)$ мы будем понимать число ошибок, которые допускает стратегия F при прогнозировании первых n значений: $\varphi(x_1), \varphi(x_2), \ldots, \varphi(x_n)$.

С другой стороны, А.Н.Колмогоровым [4] было введено фундаментальное понятие сложности конструктивных объектов. Согласно этой идее под сложностью функции f относительно метода программирования $B(p,x)$ следует понимать величину

$$K_B(f) = l(k_B(f)),$$

где $l(k)$ - длина двоичной записи числа k и

$$k_B(f) = \min_{p:\lambda x B(p,x)=f(x)} p.$$

Далее можно показать, что существует такой ч.р. метод программирования $B^{\circ}(p,x)$, называемый оптимальным, при котором величина $K_{B_o}(f)$, с точностью до аддитивной константы, является минимальной.

Мы, однако, в дальнейшем будем рассматривать только о.р. методы программирования. Среди них оптимального метода программирования не существует. Итак, пусть $A(p,x)$ – произвольная о.р.ф. Положим

$$k_A(\Omega\varphi^n) = \min_{\substack{p:A(p,x_i)=\varphi(x_i)\\ i=1,\dots,n}} p \quad \text{и} \quad K_A(\Omega\varphi^n) = l(k_A(\Omega\varphi^n)).$$

Содержательно $K_A(\Omega\varphi^n)$ – это сложность n -начального куска $\{\varphi(x_1),\dots,\varphi(x_n)\}$ функции φ относительно метода программирования $A(p,x)$. Очевидно, если φ -нерекурсивная функция, то $K_A(\Omega_o\varphi^n) \longrightarrow \infty$ при $n \longrightarrow \infty$. Наша цель – установить связь между величинами

$$F^{NV}(\Omega\varphi^n) \quad \text{и} \quad K_A(\Omega\varphi^n).$$

ТЕОРЕМА I'. Для любой о.р.ф. $A(p,x)$ существует о.р. стратегия F такая, что какую бы (нерекурсивную) функцию φ и последовательность Ω мы ни брали, для любого n

$$F^{NV}(\Omega\varphi^n) \leqslant K_A(\Omega\varphi^n) + \log_2 K_A(\Omega\varphi^n) + o(\log_2 K_A(\Omega\varphi^n)).$$

ДОКАЗАТЕЛЬСТВО. Пусть $A(p,x)$ – о.р.ф., $A_p = \lambda x A(p,x)$ и $\mathcal{A} = \{A_1, A_2, \dots, A_p, \dots\}$. Для нумерованного класса $(\mathcal{A}, A)$ рассмотрим $F^{NV}_{\mathcal{A},A}(p)$. Пусть $\eta(p)$ – некоторая функция такая, что для любого p $F^{NV}_{\mathcal{A},A}(\Omega, p) \leqslant \eta(p)$, следовательно, для любого p также $F^{NV}(A_p) \leqslant \eta(p)$. Пусть $p_n = k_A(\Omega\varphi^n)$. Согласно определению p_n для $i \leqslant n$ имеем $A(p_n, x_i) = \varphi(x_i)$. Отсюда $F^{NV}(\Omega\varphi^n) \leqslant F^{NV}(\Omega, p_n) \leqslant$ $\leqslant \eta(p_n) = \eta(k_A(\Omega\varphi^n))$. Таким образом, мы доказали следующее утверждение:

А. Пусть $A(p,x)$ - произвольная о.р.ф. и пусть $\eta(p)$ - некоторая функция такая, что для любого p $F^{NV}_{A,A}(p) \le \eta(p)$. Тогда для любой (нерекурсивной) функции φ , любой последовательности Ω и любого n

$$F^{NV}(\Omega\varphi^n) \le \eta(k_A(\Omega\varphi^n)).$$

Из утверждения А и теоремы I непосредственно вытекает теорема I'.

Возвращаемся опять к прогнозированию о.р.ф.

ТЕОРЕМА 2. Существует нумерованный класс (U,τ) такой, что для любой стратегии F :

I) $\forall n\,(F^{NV}_{U,\tau}(\Omega_0,n) > \log_2 n - 3)$,

2) $\overset{\infty}{\exists} n\,(F^{NV}_{U,\tau}(\Omega_0,n) > \log_2 n + \log_2\log_2 n)$.

ДОКАЗАТЕЛЬСТВО. Построить нумерованный класс для утверждения I) легко: $U' = \{\bar{\alpha}0^\infty \mid \bar{\alpha}$ - двоичное слово$\}$ с лексикографической нумерацией τ' по $\bar{\alpha}$. Легко проверить, что $\forall n\,(F^{NV}_{U',\tau'}(\Omega_0,n) \ge \log_2 n - 2)$. Остается построить нумерованный класс (U'',τ'') для утверждения 2). Тогда класс $U = U' \cup U''$ с нумерацией

$$\tau_n = \begin{cases} \tau'_k, \text{если } n = 2k-1, \\ \tau''_k, \text{если } n = 2k \end{cases}$$

будет удовлетворять требованиям теоремы.

Для построения класса (U'',τ'') мы будем использовать следующую теорему Р.Мартин-Лёфа [5] (см.также [6]): Пусть $f(n)$ - произвольная о.р.ф. такая, что ряд $\sum 2^{-f(n)}$ расходится; тогда для любой двоичной (т.е. принимающей только значения 0 и I) функции φ

$$\overset{\infty}{\exists} n\,(K_{B_0}(\varphi^n) \le n - f(n)).$$

В качестве f можно брать, например, функцию $f(n) = \log_2 n + a(n)$, где $a(n)$ - достаточно медленно растущая о.р.ф. ($a(n) \to \infty$, когда $n \to \infty$).

В теореме Мартин-Лёфа $B^{\circ}(p,x)$ - ч.р.ф. (оптимальный метод программирования). Но доказательство этой теоремы основано на том, что указывается некоторый эффективный способ кодировки начальных кусков последовательностей, при котором и получается указанная оценка. Так как способ кодировки эффективный, то отсюда легко следует, что существует о.р.ф. $A^{\circ}(p,x)$ такая, что для любой двоичной функции

$$\exists^{\infty} n \left(K_{A^{\circ}}(\varphi^{n}) \leq n - f(n) \right).$$

В качестве искомого класса $(\mathcal{U}'', \Gamma'')$ возьмем теперь класс $(\mathcal{A}^{\circ}, A^{\circ})$. Допустим от противного, что существует стратегия F такая, что

$$\forall^{\infty} n \left(F^{NV}_{\mathcal{A}^{\circ}, A^{\circ}}(\Omega_{0}, n) \leq \log_{2} n + \log_{2}\log_{2} n \right)$$

и, следовательно,

$$\forall n \left(F^{NV}_{\mathcal{A}^{\circ}, A^{\circ}}(\Omega_{0}, n) \leq \log_{2} n + \log_{2}\log_{2} n + C \right).$$

Величину $\log_{2} n + \log_{2}\log_{2} n + C$ обозначим через $\eta(n)$ и применим утверждение А из доказательства теоремы $1'$. Получим для любой функции φ

$$\forall n \left(F^{NV}(\varphi^{n}) \leq \log_{2} k_{A^{\circ}}(\varphi^{n}) + \log_{2}\log_{2} k_{A^{\circ}}(\varphi^{n}) + C \leq K_{A^{\circ}}(\varphi^{n}) + \log_{2} K_{A^{\circ}}(\varphi^{n}) + C' \right).$$

Определим теперь двоичную функцию φ_{0} так, чтобы стратегия F на ней выдавала только ошибочные прогнозы, т.е. $\forall n \, F^{NV}(\varphi_{0}^{n}) = n$. Подставляя в предыдущем неравенстве φ_{0} вместо φ , получаем

$$\forall n \left(n \leq K_{A^{\circ}}(\varphi_{0}^{n}) + \log_{2} K_{A^{\circ}}(\varphi_{0}^{n}) + C' \right).$$

Но в то же время, согласно сказанному выше,

$$\exists^{\infty} n \left(K_{A^{\circ}}(\varphi_{0}^{n}) \leq n - \log_{2} n - a(n) \right), \; a(n) \to \infty \text{ при } n \to \infty.$$

В результате получаем, что

$$\exists^{\infty} n \left(n \leq n - \log_{2} n - a(n) + \log_{2}(n - \log_{2} n - a(n)) + C' \right).$$

Но это - противоречие. Теорема доказана.

Предельный синтез. Здесь, если H - стратегия, φ - всюду определенная функция, Ω - последовательность натуральных чисел, то значения

$$H(<x_1, \varphi(x_1), \ldots, x_m, \varphi(x_m)>)$$

называются гипотезами. Гипотеза i считается верной, если $\varphi_i(x_j) = \varphi(x_j)$ для всех j. Говорят, что стратегия H предельно синтезирует функцию φ на последовательности Ω, если соответствующая последовательность гипотез стабилизируется на верной гипотезе, т.е. на геделевом номере такой функции, которая на Ω совпадает с φ. Для характеристики "быстроты" синтеза вводится функционал:

$$H^{GN}(\Omega,\varphi) = \begin{cases} \text{число изменений гипотез, если } H \text{ предельно} \\ \text{синтезирует } \varphi \text{ на } \Omega, \\ \infty \text{ в противном случае.} \end{cases}$$

Для изучения синтеза функций нумерованного класса (U,τ) вводится функция

$$H^{GN}_{U,\tau}(n) = \max_{1 \le i \le n} H^{GN}(\Omega, \tau_i).$$

Для каждого (U,τ) легко построить о.р. стратегию со свойством $H^{GN}_{U,\tau}(\Omega,n) \le n-1$. Однако, теоремы 3 и 4 показывают, что асимптотически точной оценкой величины H^{GN} является $\log_2 n$.

ТЕОРЕМА 3. Для каждого нумерованного класса (U,τ) можно построить о.р. стратегию H такую, что для всех последовательностей Ω и всех n

$$H^{GN}_{U,\tau}(\Omega,n) \le \log_2 n + \log_2\log_2 n + o(\log_2\log_2 n).$$

ДОКАЗАТЕЛЬСТВО. Так же, как в доказательстве теоремы I, мы приписываем каждому τ-номеру i вероятность $p_i = c/i(\log_2 i)(\log_2\log_2 i)^2$. Для каждого кортежа $<x_1, y_1, \ldots, x_m, y_m>$ через $j(x_1, y_1, \ldots, x_m, y_m)$ обозначим геделев номер следующей функции η. Для определения $\eta(z)$ вычисляем для всех s с точностью до 2^{-2m} вероятность того, что произвольно выбранная функция φ из $\{\tau_1, \tau_2, \ldots, \tau_n, \ldots\}$ обладает свойством

$$\varphi(x_1) = y_1 \,\&\, \ldots \,\&\, \varphi(x_m) = y_m \,\&\, \varphi(z) = s .$$

Затем, если $z \bar{\in} \{x_1, \ldots, x_m\}$, берем s, для которого эта вероятность наибольшая (с точностью до 2^{-2m}), и полагаем $\eta(z) = s$. Если же $z = x_i$, то $\eta(x_i) = y_i$.

Теперь искомая стратегия H определяется следующим образом: $H(<x_1, y_1, \ldots, x_m, y_m>) = j(x_1, y_1, \ldots, x_k, y_k)$, где k — наименьшее среди чисел k', не превосходящих n таких, что

$$\varphi_{j(x_1, y_1, \ldots, x_{k'}, y_{k'})}(x_1) = y_1 \,\&\, \ldots \,\&\, \varphi_{j(x_1, y_1, \ldots, x_{k'}, y_{k'})}(x_m) = y_m .$$

Легко видеть, что $\varphi_{H(<x_1, y_1, \ldots, x_m, y_m)}(x_{m+1})$ является, по существу, тем прогнозом, который выдает стратегия F из доказательства теоремы I на кортеже $<x_1, y_1, \ldots, x_m, y_m, x_{m+1}>$. Неравенство

$$H(<x_1, y_1, \ldots x_m, y_m>) \neq H(<x_1, y_1, \ldots, x_m, y_m, x_{m+1}, y_{m+1}>)$$

возможно лишь в случае, когда $F(<x_1, y_1, \ldots, x_m, y_m, x_{m+1}>)$ — неверный прогноз. Но на функции τ_n F допускает не более $\log_2 n + \log_2 \log_2 n + o(\log_2 \log_2 n)$ ошибок. Следовательно, $H^{BN}(\tau_n) \leq \log_2 n + \log_2 \log_2 n + o(\log_2 \log_2 n)$.

Теорема доказана.

ТЕОРЕМА 4. Существует нумерованный класс (U, τ) такой, что для любой о.р. стратегии H найдется константа C такая, что для всех n

$$H^{BN}_{U,\tau}(\Omega_0, n) > \log_2 n - C .$$

ДОКАЗАТЕЛЬСТВО. Возьмем класс $U = \{\alpha 0^\infty \mid \alpha$ — двоичное число$\}$. Чтобы получить нумерацию τ с требуемым свойством, мы сначала осуществим одну диагональную конструкцию. Для каждой пары $<i, j>$ мы построим 2^j о.р.ф. класса U таких, что в случае, когда φ_i — о.р. стратегия, предельно синтезирующая класс U, то среди этих 2^j о.р.ф. найдется одна, на которой φ_i не менее j раз будет менять гипотезу.

Сначала мы вычисляем параллельно для всех k значения $\varphi_i(<1,0,\ldots,k,0>)$ (здесь удобно опускать значения аргумента и писать просто $\varphi_i(<0^k>)$). Если в момент времени t ни одного значения не получено, мы полагаем все 2^j функций равными 0 при $x=t$. (Если значения так и не появятся, то все 2^j функций будут совпадать с 0^∞). Пусть теперь в момент t для некоторого k мы получаем $\varphi_i(<0^k>)=1$ (можно считать, что $k\leq t$). Тогда дальнейшие значения наших 2^j функций определяем так: При $x=t$ половина функций получает значение 0, другая половина – I.

Дальше вычисляем параллельно для всех l,m значения $\varphi_i(<0^{t-1}00^l>)$, $\varphi_i(<0^{t-1}10^m>)$. Опять, пока идет процесс вычисления, дальнейшие значения всех 2^j функций полагаем равными 0. Пусть в некоторый момент t при некотором m мы обнаруживаем, что $\varphi_i(<0^{t-1}10^m>)$ определено и не равно $\varphi_i(<0^k>)$. Тогда те 2^{j-1} функций, которые имеют начальный кусок $0^{t-1}1$, делим на две части: первая половина получает очередное значение 0, вторая – I. Остальные 2^{j-1} функций полагаем далее тождественно равными 0.

Этот процесс продолжаем до тех пор, пока на одной из функций наберется j изменений значений (гипотез относительно φ_i). В этом случае все дальнейшие значения всех 2^j функций полагаем равными 0. (В случае, когда j изменений не наберется ни на одной функции, они также будут равны 0, начиная с некоторого места).

Если φ_i – о.р. стратегия, предельно синтезирующая класс U, то легко видеть, что на одной из наших 2^j функций, соответствующих паре $<i,j>$, обязательно наберется j изменений. Введем следующее кодирование троек (i,j,k); $i\geq 1, 1\leq k\leq 2^j$. Номером тройки (i,j,k) будет натуральное число с двоичным разложением

$$1\underbrace{\alpha_1\ldots\alpha_j}_{\overline{k-1}}\underbrace{10\ldots0}_{i}, \qquad (3)$$

где $\overline{k-1}$ – двоичное разложение числа $k-1$, содержащее вначале столько нулей, чтобы общая длина была равна j.

Определим теперь искомую нумерацию τ. Если n имеет вид (3) для некоторых (i,j,k), то τ_n полагаем равной k-й функции из тех 2^i, которые мы построили для пары $\langle i,j\rangle$. Если же n нельзя представить в виде (3), т.е. n имеет вид 10^s, то τ_n полагаем равной s-й функции какой-нибудь заранее фиксированной вычислимой нумерации класса U.

Итак, нами определен нумерованный класс (U,τ). Пусть о.р. стратегия $H=\varphi_l$ предельно синтезирует все функции класса U. Тогда для любого $n \geqslant 2^{l+2}$ наибольшее j такое, что число с двоичным разложением $1^j1^l0^l$ не превосходит n, обладает свойством $j > \log_2 n - l - 2$. Поэтому среди функций с τ-номером $\leqslant n$ найдется такая функция, на которой стратегия H более, чем $\log_2 n - l - 2$ раз будет менять гипотезу. Следовательно, $H^{GN}_{U,\tau}(\Omega_0, n) > \log_2 n - C$. Теорема доказана.

ЛИТЕРАТУРА

1. Барздинь Я.М., Фрейвалд Р.В. Прогнозирование общерекурсивных функций. - "ДАН СССР", 1972, 206, № I.
2. Барздинь Я.М., Подниекс К.М. К теории индуктивного вывода. - Труды симпозиума "Mathematical foundations of computer science", High Tatras, Czechoslovakia, 1973.
3. Подниекс К.М. Сравнение различных типов предельного синтеза и прогнозирования функций. - Настоящий сборник, стр.68-8I.
4. Колмогоров А.Н. Три подхода к определению понятий количество информации. - "Проблемы передачи информации".
5. Мартин-Лёф П. О понятии случайной последовательности. - "Теория вероятности и ее применения", 1966, 2, № I.
6. Звонкин А.К., Левин Л.А. Сложность конечных объектов и обоснование понятия информации и случайности с помощью теории алгоритмов. - "Успехи математических наук", 1970, 25, № 6.

ПРЕДЕЛЬНЫЙ СИНТЕЗ τ-НОМЕРОВ

Я.М.Барздинь

Будем пользоваться терминами и обозначениями, введенными в [I]. Так же, как и в [I], будем рассматривать предельный синтез нумерованных классов (U,τ), где U - некоторый класс общерекурсивных функций (о.р.ф.) и τ - вычислимая нумерация этого класса. Единственная принципиальная разница с [I] будет состоять в том, что в данной статье мы будем рассматривать синтез не геделевских номеров, а τ-номеров, и поэтому гипотезу

$$F(<x_1,\varphi(x_1),\ldots,x_t,\varphi(x_t)>)$$

будем считать верной только тогда, когда она является τ-номером некоторой функции, которая на последовательности $\Omega=\{x_1,x_2,\ldots,x_t,\ldots\}$ неотличима от φ. Ясно, что по τ-номеру всегда можно эффективно найти геделевский номер, однако обратно-не всегда. Поэтому рассматриваемая здесь задача труднее задачи синтеза геделевских номеров. Цель настоящей статьи - оценить эту степень трудности на языке функции изменения числа гипотез. Соответствующую функцию числа изменений гипотез в данном случае обозначим через $F^{\#}_{U,\tau}(\Omega,n)$:

$$F^{\#}_{U,\tau}(\Omega,n)=\max_{1\leq i\leq n}F^{\#}(\Omega,\tau_i).$$

В дальнейшем мы будем рассматривать синтез τ-номеров только на естественной последовательности $\Omega_0=\{0,1,2,\ldots\}$.

Поэтому при вычислении гипотез будем опускать значения аргументов и писать просто

$$F(<\varphi(0),\varphi(1),\ldots,\varphi(t)>).$$

Аналогично вместо $F^{\#}(\Omega_0,\varphi)$ и $F^{\#}_{U,\tau}(\Omega_0,n)$ мы будем писать $F^{\#}(\varphi)$ и $F^{\#}_{U,\tau}(n)$.

Стратегия F, которая перебирает все функции по порядку, дает верхнюю оценку $F^{\#}_{U,\tau}(n)\leq n-1$. Цель настоящей статьи - показать, что порядок этой оценки в общем случае не может быть понижен. Сопоставляя этот результат с теоремой I из [I], получаем, что переход от синтеза геделевских номеров к синтезу τ -номеров может вызвать экспоненциальный рост числа изменений гипотез.

Отметим также, что рассматриваемый в статье [2] предельный синтез конечных автоматов фактически является синтезом τ -номеров, где под нумерацией τ понимается естественная нумерация конечных автоматов. Однако там, как показывает теорема 3 из [2], имеет место логарифмическая верхняя оценка, аналогичная той, которая имеет место при синтезе геделевских номеров. Поэтому полученная ниже оценка существенно зависит от выбранного класса функций и его нумераций τ.

ТЕОРЕМА I. Существует нумерованный класс $(\mathcal{U},\tau)$ такой, что для любой о.р. стратегии F найдется константа $c>0$ со свойством: для почти всех n

$$F^{*}_{\mathcal{U},\tau}(n) > \frac{n}{c}.$$

ДОКАЗАТЕЛЬСТВО. Построение искомого нумерованного класса $\mathcal{U}=\{\tau_1,\tau_2,\ldots,\tau_n,\ldots\}$ будет основано на некоторой диагональной конструкции. Сначала все τ -номера (т.е. все натуральные числа) мы разобьем на последовательности

$$\{l_1^{(i)}, l_2^{(i)}, \ldots, l_m^{(i)}, \ldots\},$$

где $l_m^{(1)} = 2m-1$, $l_m^{(i)} = 2^{i-1}(2m-1)$, $i=1,2,\ldots$. τ - номера, принадлежащие $\{l_m^{(i)}\}$, мы будем использовать для "опровержения" стратегии $F=\varphi_i$.

Множество функций $\{\tau_{l_m^{(i)}} \mid m=1,2,\ldots\}$ разобьем на части $\{\tau_{l_m^{(i)}} \mid 2^k \leq m < 2^{k+1}\}$, $k=0,1,2,\ldots$ и каждую часть будем определять параллельно и независимо от остальных частей.

Определение функций $\tau_{l_m^{(i)}}$, $2^k \leq m < 2^{k+1}$ будет состоять из нескольких этапов:

Э т а п 0. (Этот этап специфический). Первые $i+k+3$ значений всех 2^k функций полагаем равными соответветственно $1^i01^{k+1}0$ (мы, как и в [I], функцию φ ассоциирует с последовательностью ее значений $\varphi(0)\varphi(1)\varphi(2)\ldots$).

Таким образом, в первых $i+k+3$ значениях любой из этих 2^k функций закодированы числа i, k. Начальный кусок $1^i01^{k+1}0$ обозначим через $\bar{x}_0$.

Этап I. Для определения дальнейших значений этих функций начинаем параллельно вычислять гипотезы $\varphi_i(<\bar{x}_0 0^u>)$ для $u=1,2,3,\ldots$. Пока идет процесс поиска первой гипотезы, дальнейшие значения всех 2^k функций полагаем равными 0 (с таким расчетом, что если в какой-то момент в указанном параллельном процессе вычисления гипотез мы дошли до $u=u'$, то к этому моменту обязательно все 2^k функций должны быть определены до $\bar{x}_0 0^{u'}$). Если φ_i неопределено на рассматриваемых значениях аргумента, то все 2^k функций будут равны $\bar{x}_0 0^\infty$. Допустим теперь, что в какой-то момент (пусть к этому моменту наши функции определены до $\bar{x}_0 0^{s_1}$) мы получаем значение $\varphi_i(<\bar{x}_0 0^{u_1}>)=z_1$ $(u_1 \leq s_1)$.

Если z_1 имеет вид $l_r^{(j)}$, где $j \neq i$, или вид $l_r^{(i)}$, где $r<2^k$ или $r \geq 2^{k+1}$, то гипотеза z_1 заведомо неверна (из-за начального куска $\bar{x}_0 = 1^i01^{k+1}0$), и мы переходим к этапу 2 с начальным куском рассматриваемых функций $\bar{x}_1 = \bar{x}_0 0^{s_1}$.

Если z_1 имеет вид $l_r^{(i)}$, где $2^k \leq r < 2^{k+1}$, то функцию $\tau_{l_1^{(i)}}$ мы определяем равной $\bar{x}_0 0^{s_1} 0^\infty$, а в качестве начальных кусков оставшихся 2^k-1 функций берем $\bar{x}_1 = \bar{x}_0 0^{s_1} 1$ (чтобы z_1 на этих 2^k-1 функциях была неверной гипотезой). Далее переходим к этапу 2.

Этап 2 определяется в точности так же, как этап I, только вместо начального куска $\bar{x}_0$ берется начальный кусок $\bar{x}_1$. Единственное отличие состоит в том, что ищется не просто какая-нибудь гипотеза $\varphi(<\bar{x}_1 0^{u_2}>)$, а гипотеза $z_2 = \varphi(<\bar{x}_1 0^{u_2}>)$, которая отлична от предыдущей гипотезы z_1 (пусть к этому моменту наши функции уже определены до $\bar{x}_1 0^{s_2}$, $s_2 \geq u_2$). Гипотеза z_2 анализируется в точности так же, как гипотеза z_1, и в результате мы получаем новый начальный кусок $\bar{x}_2$ для оставшихся функций.

Далее аналогично выполняются этапы 3,4,5,... и этот процесс продолжается до тех пор, пока все 2^k функций определены до конца.

Допустим теперь, что о.р. стратегия $F=\varphi_i$ синтезирует τ-номер каждой функции класса $U=\{\tau_n\}$. Возьмем произвольное $k\geqslant 0$ и рассмотрим, как ведет себя стратегия φ_i при синтезе функций $\tau_{l_m^{(i)}}$, где $2^k\leqslant m<2^{k+1}$. Легко видеть, что в этом случае описанный процесс построения этих функций продолжается по крайней мере 2^k этапов (так как на каждом этапе из этих 2^k функций отпадает не более одной функции) и на последней из них стратегия φ_i будет выдавать гипотезы $z_1\neq z_2\neq\dots\neq z_{2^k}$.

Таким образом, для любого $k\geqslant 0$ найдется m такое, что $2^k\leqslant m<2^{k+1}$ и $F^{\#}(\tau_{l_m^{(i)}})\geqslant 2^k-1$. Но $l_m^{(i)}=2^{i-1}(2m-1)$. Отсюда следует, что начиная с достаточно больших n

$$F^{\#}(n)=\max_{1\leqslant j\leqslant n}F^{\#}(\tau_j)>\frac{n}{2^{i+2}}-1.$$

Теорема доказана.

ЗАМЕЧАНИЕ I. В теореме I константа c зависит от стратегии F. Оказывается, что это по существу. А именно, нетрудно доказать следующее утверждение: для любого нумерованного класса (U,τ) и любой константы $c>0$ существует о.р. стратегия F такая, что для бесконечно многих n $F^{\#}_{U,\tau}(n)\leqslant\frac{n}{c}$. Справедливость этого утверждения вытекает из того, что для любого нумерованного класса (U,τ) существует действительное число $\rho=\overline{\lim}\frac{S_n}{n}$, где S_n – число различных функций среди первых n функций $\tau_1,\dots,\tau_n$. В зависимости от того, с какой точностью ε мы задаем число ρ при определении стратегии F, мы получаем различные константы c, при которых для бесконечно многих n $F^{\#}_{U,\tau}(n)\leqslant\frac{n}{c}$ (эти бесконечно многие n выбираются из тех n, для которых $\rho-\varepsilon\leqslant\frac{S_n}{n}<\rho+\varepsilon$). Дальнейшие вопросы, связанные с эффективностью построения соответствующей стратегии F по константе c, примыкает к статье [3] и более подробно здесь не исследуются.

ЗАМЕЧАНИЕ 2. Для предельного синтеза τ-номеров имеет место точный аналог теоремы I из [4]. Доказательство этого факта ничем не отличается от доказательства упомянутой теоремы из [4].

ЛИТЕРАТУРА

I. Барздинь Я.М., Фрейвалд Р.В. Прогнозирование и предельный синтез эффективно перечислимых классов функций.- Настоящий сборник, стр. IOI-III.
2. Барздинь Я.М. Прогнозирование и предельный синтез конечных автоматов.- Настоящий сборник, стр. I29-I44.
3. Барздинь Я.М. О частотном решении проблемы вхождения в рекурсивно перечислимое множество.- Труды Мат.ин-та им. В.А.Стеклова АН СССР, I973, I33.
4. Барздинь Я.М., Кинбер Е.Б., Подниеко К.М. Об ускорении предельного синтеза и прогнозирования функций.- Настоящий сборник, стр. II7-I28.

ОБ УСКОРЕНИИ СИНТЕЗА И ПРОГНОЗИРОВАНИЯ ФУНКЦИЙ

Я.М.Барздинь, Е.Б.Кинбер, К.М.Подниекс

Определения и обозначения см. в [I]. Единственное отличие состоит в том, что мы рассматриваем прогнозирование и синтез только на естественной последовательности $\Omega_0 = \{1, 2, 3, \ldots\}$. Поэтому можно считать, что стратегия F применяется непосредственно к начальным кускам $<\varphi(1) \ldots \varphi(n)>$ функции φ. Прогноз $F(<\varphi(1)\ldots\varphi(n)>)$ считается верным, если он равен $\varphi(n+1)$. Гипотеза $F(<\varphi(1)\ldots\varphi(n)>)$ считается верной, если она представляет собой некоторый геделев номер функции φ.

Так как у нас всегда $\Omega = \Omega_0$, то вместо $F^{GN}(\Omega,\varphi)$ и $F^{NV}(\Omega,\varphi)$ мы будем писать просто $F^{GN}(\varphi)$, $F^{NV}(\varphi)$. Для нумерованного класса (U,τ) и стратегии F вводятся величины:

$$F^{GN}_{U,\tau}(n) = \max_{1\le i\le n} F^{GN}(\tau_i), \quad F^{NV}_{U,\tau}(n) = \max_{1\le i\le n} F^{NV}(\tau_i).$$

Их можно рассматривать как своего рода сигнализирующие функции, характеризующие соответственно "скорость" предельного синтеза и "скорость" прогнозирования.

Нас будут интересовать асимптотики роста (монотонных) функций $F^{GN}_{U,\tau}(n)$, $F^{NV}_{U,\tau}(n)$. Теоремы I, 2 работы [I] показывают, что для некоторых нумерованных классов существуют о.р. стратегии, асимптотически оптимальные в смысле величины F^{NV} (этот оптимум в конкретном случае теоремы I имеет вид $\log_2 n$). Теоремы 3, 4 из [I] показывают, что аналогичный факт имеет место для F^{GN}

Возникает вопрос: имеет ли место аналогичная ситуация в случае л ю б о г о нумерованного класса (U,τ)? Оказывается, что это не так: для некоторых (U,τ) можно доказать своего рода аналог теоремы ускорения М.Блюма [2].

Доказательство этого факта соответственно для предельного синтеза и прогнозирования составляет содержание настоящей статьи.

Будем говорить, что для синтеза нумерованного класса $(\mathcal{U},\tau)$ имеет место абсолютная теорема ускорения (АТУ), если для любой о.р.ф. $r(x)$ и любой о.р. стратегии F, предельно синтезирующей класс $\mathcal{U}$, можно построить такую о.р. стратегию G, также предельно синтезирующую $\mathcal{U}$, что

$$\forall_n^\infty \; r\left(G_{\mathcal{U},\tau}^{GN}(n)\right) \leq F_{\mathcal{U},\tau}^{GN}(n). \tag{1}$$

Аналогично будем говорить, что для прогнозирования нумерованного класса $(\mathcal{U},\tau)$ выполняется АТУ, если для любой о.р.ф. $r(x)$ и любой о.р. стратегии F можно построить о.р. стратегию G такую, что

$$\forall_n^\infty \; r\left(G_{\mathcal{U},\tau}^{NV}(n)\right) \leq F_{\mathcal{U},\tau}^{NV}(n). \tag{2}$$

Теорема об ускорении называется "абсолютной" потому, что для одного и того же нумерованного класса здесь достигается ускорение любого порядка $r(x)$. (В теореме Блюма для каждой $r(x,y)$ строилась своя функция).

Сначала докажем простую лемму, которой будем неоднократно пользоваться в дальнейшем.

ЛЕММА. Для синтеза (прогнозирования) нумерованного класса $(\mathcal{U},\tau)$ выполняется абсолютная теорема ускорения, если и только если выполняются условия:

а) невозможна о.р. стратегия F такая, что $F_{\mathcal{U},\tau}^{GN}(n) = O(1)$ (соответственно $F_{\mathcal{U},\tau}^{NV}(n) = O(1)$);

б) для любой монотонной неограниченной о.р.ф. $f(n)$ найдется о.р. стратегия G, предельно синтезирующая (прогнозирующая) класс $\mathcal{U}$, такая, что $\forall_n^\infty$ $G_{\mathcal{U},\tau}^{GN}(n) \leq f(n)$ (соответственно $G_{\mathcal{U},\tau}^{NV}(n) \leq f(n)$).

ДОКАЗАТЕЛЬСТВО мы проведем для предельного синтеза;

в случае прогнозирования рассуждения аналогичны.

НЕОБХОДИМОСТЬ. Пусть для синтеза нумерованного класса (U,τ) имеет место АТУ. Тогда выполнение условия а) очевидно.

Для доказательства б) возьмем монотонную неограниченную о.р.ф. $f(n)$. Легко построить о.р. стратегию F со свойством

$$\forall n \; F^{GN}_{U,\tau}(n) \leq n \tag{3}$$

(F работает, перебирая последовательно все функции τ_1, $\tau_2, \ldots$). Затем найдем о.р.ф. $z(x)$ со свойством

$$\forall x \, \forall n \left(z(x) \leq n \rightarrow x \leq f(n) \right). \tag{4}$$

Тогда, по предположению, существует о.р. стратегия G со свойством:

$$\overset{\infty}{\forall} n \; z\left(G^{GN}_{U,\tau}(n) \right) \leq F^{GN}_{U,\tau}(n). \tag{5}$$

Теперь (3), (4), (5) дают вместе:

$$\overset{\infty}{\forall} n \; G^{GN}_{U,\tau}(n) \leq f(n),$$

что и требовалось.

ДОСТАТОЧНОСТЬ. Пусть для нумерованного класса (U,τ) выполняются условия а),б). Возьмем некоторую о.р. стратегию F, которая предельно синтезирует класс $\mathcal{U}$, и произвольную о.р.ф. $z(x)$. Сразу же перейдем к монотонной функции $z(x) = \max\limits_{y \leq x} z(y)$.

Если применять стратегию F параллельно ко всем функциям τ_i, подсчитывая для каждой число изменений гипотез, мы сможем легко построить монотонную неограниченную о.р.ф. $f(n)$ такую, что

$$\forall n \; F^{GN}_{U,\tau}(n) \geq f(n) \tag{6}$$

(ведь по условию а) $F^{GN}_{U,\tau}(n) \rightarrow \infty$).

Затем можно найти также монотонную неограниченную функцию $g(n)$ со свойством:

$$\forall_n^\infty \, r'g(n) \leq f(n). \qquad (7)$$

По условию б) найдется о.р. стратегия G, предельно синтезирующая класс $\mathcal{U}$ такая, что:

$$\forall_n^\infty \, G_{\mathcal{U},\tau}^{GN}(n) \leq g(n).$$

Отсюда, учитывая, что $r \leq r'$, а r' - монотонна, получаем:

$$\forall_n^\infty \, r\left(G_{\mathcal{U},\tau}^{GN}(n)\right) \leq r'g(n),$$

что вместе с (6), (7) дает

$$\forall_n^\infty \, r\left(G_{\mathcal{U},\tau}^{GN}(n)\right) \leq F_{\mathcal{U},\tau}^{GN}(n).$$

Лемма доказана.

Ускорение предельного синтеза.

ТЕОРЕМА I. Если нумерованный класс $(\mathcal{U},\tau)$ обладает свойствами:

а) предикат $P(i,j) \equiv (\tau_i = \tau_j)$ рекурсивен,

б) невозможна о.р. стратегия F такая, что $F_{\mathcal{U},\tau}^{GN}(n) = O(1)$

то для синтеза $(\mathcal{U},\tau)$ имеет место абсолютная теорема ускорения.

ДОКАЗАТЕЛЬСТВО. Воспользуемся леммой. Чтобы обеспечить условия ее применимости, возьмем произвольную монотонную неограниченную функцию $f(n)$ и будем строить для нее следующую о.р. стратегию F.

Используя возможность нумерации τ и рекурсивность предиката $\tau_i = \tau_j$ для каждого n можно эффективно найти число $z(n)$ такое, что две функции $\tau_i(x), \tau_j(x)$ $(i,j \leq n)$ либо полностью совпадают, либо отличаются уже при некотором $x \leq z(n)$. (можно считать, что $z(n) < z(n+1)$).

Пусть теперь стратегия F получает последовательность значений функции φ. Сначала F находит число n, такое, что $f(n_1) > 1$ и затем полагает для всех $x \le z(n_1) - 1$

$$F(<\varphi(1)\ldots\varphi(x)>) = 0.$$

Далее, для вычисления следующей гипотезы F сравнивает функции $\varphi(x), \tau_1(x), \ldots, \tau_{n_1}(x)$ при $x \le z(n_1)$. Ясно, что для этих x φ может совпадать не более чем с одной из функций $\tau_1, \ldots, \tau_{n_1}$. Если ни с одной из них φ не совпадает, полагаем $F(<\varphi(1)\ldots\varphi z(n_1)>)$ = предыдущей гипотезе. Если l - наименьший номер совпадения, полагаем $F(<\varphi(1)\ldots\varphi z(n_1)>) = d(l)$, где $d(x)$ - о.р.ф., сводящая нумерацию τ к геделевской нумерации. Далее F "итерирует" последнюю гипотезу до $x = z(n_2) - 1$, где n_2 - такое число, что $n_2 > n_1 \,\&\, f(n_2) > 2$. И опять, чтобы определить $F(<\varphi(1)\ldots\varphi z(n_2)>)$, стратегия сравнивает функции $\varphi(x), \tau_1(x), \ldots, \tau_{n_2}(x)$ при $x \le z(n_2)$ и т.д.

Стратегия F определена. Нетрудно убедиться, что F предельно синтезирует класс U.

Если $n \ge n_1$ и оказалось, что $\varphi = \tau_i \,\&\, i \le n$, то F на φ будет менять гипотезу возможно лишь в таких точках $x = z(n_1), \ldots, z(n_k)$, где $n_k \le n < n_{k+1}$, после чего наступит стабилизация на геделевском номере функции φ. В таком случае для числа изменений гипотезы должно выполняться неравенство

$$F^{GN}_{U,\tau}(n) \le k < f(n)$$

(так как $f(n_k) > k$ и f монотонна). Итак, условие б) леммы выполнено, поэтому из нее следует утверждение теоремы I.

Согласно одной теореме Фридберга [3] (см.также [4]), каждый бесконечный эффективно перечислимый класс о.р.ф. имеет однозначную вычислимую нумерацию. Но для такой нумерации предикат $\tau_i = \tau_j$ рекурсивен. Поэтому из теоремы I вытекает

СЛЕДСТВИЕ. Для каждого нетривиального (условие б) теоремы I) эффективно перечислимого класса U о.р.

функций существует вычислимая нумерация τ такая, что для синтеза $(\mathcal{U},\tau)$ выполняется абсолютная теорема ускорения.

Отсюда следует также, что справедливость нижней оценки $\log_2 n - O(1)$ в теореме 4 из [I] существенно опирается на специальный выбор нумерации τ. Для однозначной нумерации класса U этой теоремы имеет место АТУ.

П р о б л е м а   у с к о р е н и я   п р о г н о з и р о в а н и я.

Для прогнозирования абсолютная теорема ускорения также может выполняться, но только в очень редких случаях. Область поиска таких случаев ограничивается теоремой 2.

ТЕОРЕМА 2. Если для прогнозирования нумерованного класса (U,τ) выполняется абсолютная теорема ускорения, то существует (нерекурсивная) стратегия H такая, что $H^{NV}_{U,\tau}(n) = O(1)$.

ДОКАЗАТЕЛЬСТВО. $p\times q$ - таблицей класса (U,τ) назовем таблицу чисел $\tau_i(x)$, где $i \leqslant p$, $x \leqslant q$. Всевозможные стратегии H, действуя в пределах $p\times q$ - таблицы, дают только конечное число вариантов распределения ошибок по функциям $\tau_1,\dots,\tau_p$ до $x = q$. Все эти варианты можно эффективно перебрать и найти таким образом число $s(p,q)$ такое, что:

а) любая стратегия H делает в некоторой строке $p\times q$ таблицы (U,τ) не менее $s(p,q)$ ошибок;

б) найдется стратегия H_0, которая во всех строках этой таблицы делает не более $s(p,q)$ ошибок.

Ясно, что функция $s(p,q)$ общерекурсивна и монотонна по p и q. Очевидно также, что для любой стратегии H:

$$\forall p \forall q \; H^{NV}_{U,\tau}(p) \geqslant s(p,q). \qquad (8)$$

Поскольку существует о.р. стратегия F со свойством $F^{NV}_{U,\tau}(p) \leqslant p$, то при фиксированном p функция $s(p,q)$ будет ограниченной. Поэтому, если эта функция в целом была бы все же неограниченной, то можно было бы найти о.р.ф. $t(p)$ такую, что $s(p, t(p)) \to \infty$ монотонно. Но это, в силу (6), означает, что нарушено условие б) леммы.

Таким образом, если для прогнозирования (U,τ) имеет место АТУ, то функция $s(p,q)$ ограничена: $\forall p \forall q\ s(p,q) \leq C$.

Теперь мы можем доказать существование требуемой стратегии H (она будет обязательно нерекурсивной в силу условия а) леммы).

В силу $s(q,q) \leq C$ для каждого q непусто множество $\widetilde{H}_q$ тех стратегий, которые в строках $q \times q$ -таблицы (U,τ) делают не более C ошибок. $\widetilde{H}_q$ разбивается на конечную систему классов, где в один класс попадают стратегии, неотличимые с точки зрения их действия на $q \times q$ -таблице. Обозначим эту систему следующим образом: $\{\widetilde{H}_q^1, \ldots$ $\ldots, \widetilde{H}_q^{k_q}\}$. Легко видеть, что

$$\forall k \leq k_{q+1}\ \exists l \leq k_q\ \widetilde{H}_{q+1}^k \subseteq \widetilde{H}_q^l .$$

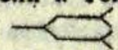
Отсюда по теореме компактности для конечно-ветвящихся деревьев получаем, что существует стратегия H со свойством:

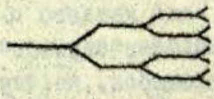
$$\forall q\ \exists k \leq k_q\ H \in \widetilde{H}_q^k$$

или просто $H \in \widetilde{H}_q$ для всех q. По определению $\widetilde{H}_q$ стратегия H делает на любой $q \times q$ -таблице (U,τ) не более C ошибок, поэтому $H_{U,\tau}^{NV}(q) = O(1)$, что и требовалось доказать.

Теорема 2 указывает, таким образом, в какой примерно области можно искать нумерованные классы, для прогнозирования которых может выполняться АТУ. Простейшим типом классов этой области будут классы со следующим деревом:

Назовем такие классы 1 -классами ("ствол" дерева - функция g может входить в класс, но не обязательно). Таким образом, если U - 1 -класс со "стволом" g , то из $\varphi, \psi \in U \& (\forall x < n\ \varphi(x) = \psi(x) = \varphi(x)) \& (\varphi(n) = \psi(n) \neq g(n))$ следует, что $\varphi = \psi$.

ЗАМЕЧАНИЕ I. Легко видеть, что некоторое дерево функций будет деревом 1 -класса, если и только если это дерево не содержит поддеревьев вида  . Если дерево функций класса U не содержит поддеревьев вида



этот класс называется 2 -классом, и так далее. Нетрудно доказать, что для всякого t -класса найдется стратегия H, прогнозирующая все функции этого класса с числом ошибок $\leq t$. (Однако даже в случае эффективно перечислимого 1 - класса эта стратегия может и не быть рекурсивной, см. теорему 4). Верно и обратное: если существует H со свойством $H^{NV}(\varphi) \leq C$ для всех $\varphi \in U$, то U есть t - класс для некоторого t .

Теорема 3 показывает, что семейство всех эффективно перечислимых 1 -классов распадается на два подсемейства: "тривиальные" классы и классы, для прогнозирования которых имеет место АТУ. (Теорема 4 покажет, что второе подсемейство непусто).

ТЕОРЕМА 3. Если нумерованный класс (U, τ) обладает свойствами:

а) U является 1 -классом,

б) невозможна о.р. стратегия F такая, что $F^{NV}_{U,\tau}(n) = O(1)$,

то для прогнозирования (U, τ) имеет место абсолютная теорема ускорения.

ДОКАЗАТЕЛЬСТВО. Воспользуемся леммой. Возьмем в качестве $f(n)$ произвольную монотонную неограниченную о.р.ф. и будем строить требуемую стратегию F.

Поскольку $\mathcal{U}$ - бесконечный 1 -класс (условие б), то по нумерации τ можно эффективно найти "ствол" дерева класса $\mathcal{U}$ - некоторую о.р.ф. g. В самом деле: если $\tau_i(x)=\tau_j(x)$ при $x\leqslant n$ и $\tau_i(n+1)\neq\tau_j(n+1)$, то $g(x)=\tau_i(x)$ для $x\leqslant n$. Все пары (i,j) такие, что $\tau_i\neq\tau_j$ можно эффективно перечислить, и, следовательно, так как $\mathcal{U}$ бесконечен, функцию $g(x)$ можно вычислить на всех x.

Теперь, имея функцию g, определяем действие стратегии F на функции φ следующим образом: если при $x\leqslant n$ $\varphi(x)=g(x)$, то полагаем $F(<\varphi(1)\ldots\varphi(n)>)=g(n+1)$.

При первой же ошибке (показывающей, что $\varphi\neq g$) берем данную о.р.ф. $f(n)$ и находим возрастающую о.р.ф. $h(x)$ со свойством

$$\forall n\ h(f(n)\dot{-}1)\geqslant n. \tag{9}$$

Чтобы теперь определить прогноз $F(<\varphi(1)\ldots\varphi(m)>)$, ищем наименьшее $i\leqslant h(m)$ такое, что $\tau_i(x)=\varphi(x)$ при $x\leqslant m$. Если нашли, то полагаем

$$F(<\varphi(1)\ldots\varphi(m)>)=\tau_i(m+1), \tag{10}$$

если такого i не существует, пусть прогноз будет каким угодно, например, 0. Стратегия F определена полностью.

Подсчитаем число ошибок, которые F допустит при прогнозировании функции τ_n. Если $\tau_n=g$, то ошибок, разумеется, не будет. Если же $\tau_n\neq g$, то первая ошибка будет допущена в прогнозе первого значения $\tau_n(m_0)$, не равного $g(m_0)$. С этого момента наступает "период сомнительных прогнозов", который кончается как только m достаточно велико, а именно: $h(m)\geqslant n$. В самом деле, если $m\geqslant m_0\ \&\ h(m)\geqslant n$, то прогноз (10) не может дать ошибку: ведь $\mathcal{U}$ есть 1 -класс, $\tau_n(m_0)\neq g(m_0)$, и поэтому никакое i со свойством $\tau_i\neq\tau_n$ не попадает в (10).

Итак: $F^{NV}(\tau_n) \le 1 + \min\{m \mid h(m) \ge n\}$.
Но из (9) следует, что $f(n) \dot- 1 \in \{m \mid h(m) \ge n\}$, поэтому
$$F^{NV}(\tau_n) \le 1 + (f(n) \dot- 1),$$
и, наконец, $\forall\overset{\infty}{n}\, F^{NV}_{\mathcal{U},\tau}(n) \le f(n)$.

Для любой $f(n)$ можно построить такую F. Теперь, используя лемму, получаем утверждение теоремы 3.

ТЕОРЕМА 4. Существует нумерованный класс $(\mathcal{U}', \tau')$ такой, что:

а) $\mathcal{U}'$ является 1-классом

б) невозможна о.р. стратегия F со свойством
$$F^{NV}_{\mathcal{U}',\tau'}(n) = O(1).$$

ДОКАЗАТЕЛЬСТВО. Возьмем диагональную функцию $\varphi_x(x)$ и некоторое эффективное перечисление ее графика: $<x_1, y_1>$, $<x_2, y_2>, \dots$. Тогда τ'_n определяется как последовательность нулей и единиц:
$$\tau'_n = 0^{x_n} \alpha_1 \dots \alpha_k 0^{\infty},$$
где $\alpha_1 \dots \alpha_k$ суть двоичная запись числа y_n.

Таким образом, определен класс $\mathcal{U}'$, его вычислимая нумерация τ'. Очевидно, $\mathcal{U}'$ является 1-классом, его дерево имеет "ствол" $g = 0^{\infty}$.

Допустим теперь от противного, что существует о.р. стратегия F и число t такое, что
$$\forall n\, F^{NV}(\tau'_n) \le t. \tag{11}$$

Для каждого x построим следующее двоичное слово $\beta_1 \dots \beta_{t+1}$:
$$\begin{aligned} &F(0^x 1) = 1 - \beta_1, \\ &F(0^x 1 \beta_1) = 1 - \beta_2, \\ &\dots\dots\dots\dots\dots \\ &F(0^x 1 \beta_1 \dots \beta_t) = 1 - \beta_{t+1}. \end{aligned} \tag{12}$$

Число с двоичным разложением $1\beta_1\ldots\beta_{t+1}$ обозначим через $h(x)$. Очевидно, $h(x)$ - о.р.ф. от x (в силу того, что F - о.р. стратегия). Но тогда оказывается, что

$$\forall x\ \varphi_x(x) \neq h(x). \tag{I3}$$

В самом деле, если $\varphi_x(x) = h(x)$, то функция $0^x 1\beta_1\ldots\beta_{t+1}0^\infty \in \mathcal{U}'$, однако F делает (в силу (I2)) на этой функции $\geqslant t+1$ ошибок, что противоречит (II).

Но (I3) также представляет собой противоречие: для некоторого a имеем $h = \varphi_a$ и, далее, $\varphi_a(a) \neq h(a)$ или $\varphi_a(a) \neq \varphi_a(a)$.

Поэтому невозможна о.р. стратегия F со свойством $F^{NV}_{\mathcal{U}',\tau'}(n) = O(1)$. Теорема доказана.

Теоремы 3, 4 вместе дают

СЛЕДСТВИЕ. Существует эффективно перечислимый класс (даже 1 -класс) $\mathcal{U}'$ такой, что при любой вычислимой его нумерации τ для прогнозирования нумерованного класса $(\mathcal{U}', \tau)$ имеет место абсолютная теорема ускорения.

ЗАМЕЧАНИЕ 2. Согласно теореме 3, всякий эффективно перечислимый 1 -класс либо тривиален, либо для его прогнозирования выполняется АТУ (независимо от выбора нумерации). Однако в случае 2 -классов (см.замечание I) это уже не так. Подходящая модификация доказательства теоремы 4 из [I] позволяет построить нумерованный класс $(\mathcal{U}, \tau)$ такой, что:

I) $\mathcal{U}$ является 2 -классом,

2) Для всякой о.р. стратегии F найдется константа C такая, что $\forall n\ F^{NV}_{\mathcal{U},\tau}(n) > \log_2 n - C$.

Класс $\mathcal{U}$ нетривиален, однако для прогнозирования $(\mathcal{U},\tau)$ АТУ не выполняется (условие б) леммы нарушено, например, при $f(n) = \frac{1}{2}\log_2 n$). С другой стороны, анализ доказательства теорем 2, 3 показывает, что если нумерованный класс $(\mathcal{U},\tau)$ обладает свойствами

I) $\mathcal{U}$ является t -классом для некоторого $t \geqslant 2$,

2) предикат $\tau_i \equiv \tau_j$ рекурсивен,

3) невозможна о.р. стратегия F такая, что $F^{NV}_{U,\tau}(n) = O(1)$,

то для прогнозирования (U,τ) выполняется АТУ. Таким образом, для каждого нетривиального t -класса U найдется нумерация τ (в качестве τ можно взять однозначную нумерацию U) такая, что для прогнозирования (U,τ) имеет место АТУ.

Основные результаты этой статьи сформулированы без доказательств в [5]. Постановки задач и теорема I принадлежат Я.М.Барздиню, теоремы 2, 3, 4 доказаны К.П.Подниексом и Е.Б.Кинбером.

ЛИТЕРАТУРА

I. Барздинь Я.М., Фрейвалд Р.В. Прогнозирование и синтез эффективно перечислимых классов функций.- Настоящий сборник, стр. IOI-III.
2. Блюм М. Машинно-независимая теория сложности рекурсивных функций. - Сб. "Проблемы математической логики", М., I970.
3. Friedberg R. Three theorems on recursive enumeration. - "Journal of Symbolic Logic", 1958, 23, No.2.
4. Ершов Ю.Л. Теория нумераций I, Новосибирск, 1969.
5. Барздинь Я.М., Подниекс К.М. К теории индуктивного вывода. - Труды симпозиума "Mathematical foundations of computer science", High Tatras, Czechoslovakia, 1973.

ПРОГНОЗИРОВАНИЕ И ПРЕДЕЛЬНЫЙ СИНТЕЗ КОНЕЧНЫХ АВТОМАТОВ *)

Я.М.Барздинь

Рассматриваются инициальные конечные автоматы (Мили) с фиксированным входным алфавитом $X=\{1,2,\ldots,a\}$. Выходной алфавит может быть произвольным с тем лишь неосущественным ограничением, что он выбирается из множества $\{1,2,\ldots,n\}$. Класс всех таких автоматов обозначим через U_a. Подкласс, который получается фиксацией выходного алфавита $Y=\{1,2,\ldots,b\}$, обозначим через $U_{a,b}$.

Автоматы рассматриваются как "черные ящики", о которых известно лишь то, что они принадлежат классу U_a. Пусть на вход некоторого такого автомата A подается входная последовательность

$$\omega=\{x(1),x(2),\ldots,x(t),\ldots\}$$

и пусть

$$\{y(1),y(2),\ldots,y(t),\ldots\}$$

– соответствующая выходная последовательность. Требуется для любого t по $\{x(1),\ldots,x(t)\},\{y(1),\ldots,y(t)\},x(t+1)$ прогнозировать $y(t+1)$. При этом разрешается пользоваться любыми эффективными правилами (называемыми стратегиями).

Ясно, что безошибочное решение такой задачи невозможно. Поэтому возникает вопрос о числе ошибок, возникающих в процессе прогнозирования. Основным результатом настоящей работы является доказательство того, что данная задача прогнозирования может быть решена с относительно небольшим числом ошибок: для автоматов с k состояниями – порядка $(a-1)k\log_2 k$ и что эта оценка в общем случае не может быть асимптотически понижена. Заметим, что простой перебор всех автоматов по порядку дает верхнюю оценку только вида k^k.

Переходим к точным определениям. Под стратегией мы будем понимать любую одноместную общерекурсивную функцию.

---

*) В несколько более слабой форме и без доказательств результаты настоящей статьи были опубликованы в [1] и [2].

Будем говорить, что стратегия Σ, примененная к входной последовательности ω и автомату A, в момент t совершает ошибку, если

$$\Sigma(\langle x(1),\ldots,x(t),y(1),\ldots,y(t),x(t+1)\rangle)\neq y(t+1)$$

($\langle\ldots\rangle$ - эффективная нумерация кортежей натуральных чисел; в дальнейшем вместо $\Sigma(\langle x(1),\ldots,x(t),y(1),\ldots,y(t),x(t+1)\rangle)$ обычно будем писать $\Sigma(x(1),\ldots,x(t);y(1),\ldots,y(t);x(t+1))$.

Число различных t, в которых Σ совершает ошибку, обозначим через $\Sigma^*(\omega,A)$. Пусть U - некоторый класс автоматов. Положим

$$\Sigma^*(\omega,U,k)=\max\Sigma^*(\omega,A),$$

где $\max$ берется по всем $A\in U$, которые имеют не более k состояний.

ТЕОРЕМА I. Пусть $a\geq 2$. Существует стратегия Σ такая, что для любой входной последовательности ω

$$\Sigma^*(\omega,U_a,k)\leq(a-1)k\log_2 k.$$

ДОКАЗАТЕЛЬСТВО. Вместо автоматов класса U_a рассмотрим автоматные графы [3] с входным алфавитом $X=\{1,\ldots,a\}$. Из каждого класса изоморфных графов (как графов с фиксированной начальной вершиной) выберем по представителю и упорядочим эти представители в порядке возрастания числа вершин, попутно выбрасывая те графы, у которых часть, достижимая из начальной вершины, совпадает с каким-нибудь из предыдущих графов (так как такие графы не порождают новые автоматные операторы). Графы с одинаковым числом вершин могут быть расположены в любом порядке. Получим последовательность графов

$$\mathcal{G}=\{G_1,G_2,\ldots,G_i,\ldots\}.$$

Очевидно, если число вершин $|G_i|$ графа G_i не превышает k, то

$$i\leq I(a,k),\tag{I}$$

где $I(a,k)$ - число всех попарно неизоморфных инициальных автоматных графов с k вершинами и a-буквенным входным алфавитом. Из [4] следует, что

$$I(a,k) \sim \frac{k^{ak}}{k!} \cdot k, \text{ если } a \geqslant 3, \text{ и } I(a,k) \sim e^{\frac{1}{2e}} \frac{k^{ak}}{k!} \cdot k, \text{ если } a = 2 \qquad (2)$$

(множитель k , по сравнению с [4], возникает за счет того, что мы рассматриваем инициальные автоматные графы).

Далее, следуя идее, использованной при доказательстве теоремы I из [5], графам G_i сопоставим веса

$$p(G_i) = \frac{c_0}{i(\log_2(i+1))^2}, \qquad (3)$$

где положительное действительное число c_0 подобрано так, чтобы $\sum_{i=1}^{\infty} p(G_i) = s_0 < 1$. Нетрудно убедиться, что ряд $\sum p(G_i)$ сходится конструктивно.

Сначала мы построим неэффективную "стратегию" $\bar{\Sigma}$, которая даст требуемую оценку. Эта стратегия в процессе своей работы будет просматривать всю бесконечную последовательность $\mathcal{G}$. Потом, используя конструктивную сходимость ряда $\sum p(G_i)$, мы "обрежем" эту "стратегию" и сделаем ее эффективной.

"Стратегию" $\bar{\Sigma}$ мы определим как некоторый последовательный процесс прогнозирования, который дугам автоматных графов из $\mathcal{G}$ последовательно будет приписывать выходные буквы (постепенно превращая их в автоматы), и будет вычеркивать те графы, для которых обнаружится несовместимость с входом $x(1), \dots, x(t)$ и выходом $y(1), \dots, y(t)$. Пусть путь $x(1), \dots, x(t)$ в графе G - это путь, начинающийся с начальной вершины и соответствующий входному слову $x(1) \dots x(t)$.

Прогнозирование мы начнем с шага $t = 1$, когда по $x(1), y(1), x(2)$ требуется определить $y(2)$. В качестве исходной последовательности автоматных графов возьмем последовательность $\mathcal{G}^0 = \{G_1^0, \dots, G_i^0, \dots\}$, совпадающую с первоначальной последовательностью $\mathcal{G}$, за исключением того, что в каждом графе G_i дуге $x(1)$, исходящей из начальной вершины, приписана выходная буква $y(1)$; веса автоматных графов оставлены прежние. В результате работы шага $t=1$ мы получим новую последовательность $\mathcal{G}^1 = \{G_1^1, \ldots, G_i^1, \ldots\}$ с новыми весами.

Опишем сразу общий шаг t, соответствующий

$$x(1), \ldots, x(t);\ y(1), \ldots, y(t);\ x(t+1).$$

В качестве исходной последовательности автоматных графов возьмем последовательность $\mathcal{G}^{t-1} = \{G_1^{t-1}, \ldots, G_i^{t-1}, \ldots\}$, полученную на предыдущем шаге. Эта последовательность характеризуется тем, что в каждом G_i^{t-1} дугам, соответствующим пути $x(1)x(2)\ldots x(t)$, приписаны соответственно выходные буквы $y(1), y(2), \ldots, y(t)$, притом каждой дуге разрешается приписать не более одной выходной буквы. Очевидно, $\mathcal{G}^{t-1}$ будет содержать уже не все автоматные графы, так как не для любого автоматного графа последнее требование выполнимо. Другими словами, если G_i^{t-1} рассматривать как автомат, то при подаче $x(1)\ldots x(t)$ он должен выдать $y(1)\ldots y(t)$ и перейти в состояние $q_{\nu_t} = q_1 x(1)\ldots x(t)$. Скажем, что G_i^{t-1} при подаче $x(t+1)$ выдает выходную букву y, если дуга $x(t+1)$ вершины q_{ν_t} принадлежит пути $x(1)\ldots x(t)$ и ей приписана выходная буква y. Если G_i^{t-1} при подаче $x(t+1)$ выдает какую-нибудь выходную букву, т.е. дуга $x(t+1)$ вершины q_{ν_t} принадлежит пути $x(1)\ldots x(t)$, то будем говорить, что на данном шаге G_i^{t-1} участвует в прогнозировании.

Кроме того, в результате работы предыдущего шага элементам последовательности $\mathcal{G}^{t-1}$ должны быть приписаны определенные веса $p(G_i^{t-1})$ и сумма этих весов $\sum_{i=1}^{\infty} p(G_i^{t-1}) = \Delta_0 < 1$. Под весом выходной буквы y в G_i^{t-1} будем понимать вес самого G_i^{t-1}, если G_i^{t-1} при подаче $x(t+1)$ выдает y; в противном случае вес буквы y в G_i^{t-1} будем считать равным 0.

"Стратегия" $\overline{\overline{\Sigma}}$, примененная к $x(1), \ldots, x(t), y(1), \ldots, y(t), x(t+1)$, в качестве результата выдает ту выходную букву, которая в сумме по всем элементам последовательности $\mathcal{G}^{t-1}$, участвующим на данном шаге в прогнозировании, имеет наибольший вес (если таких букв несколько, то выбирается одна из них, например, первая по порядку).

Для полного описания "стратегии" Σ остается только определить новую последовательность $\mathcal{G}^t$. Сначала из последовательности $\mathcal{G}^{t-1}$ вычеркиваем все те члены, которые при подаче $x(t+1)$ выдают какую-нибудь выходную букву y (т.е. участвуют в прогнозировании), но $y \neq y(t+1)$ (это как раз те автоматные графы, для которых обнаруживается несовместимость с входом $x(1),\dots,x(t),x(t+1)$ и выходом $y(1),\dots,y(t),y(t+1)$. Далее, в оставшихся автоматных графах дуге $x(t+1)$ вершины q_t, если ей еще не приписана выходная буква, приписываем $y(t+1)$. Получаем последовательность $\mathcal{G}_t^t = \{G_1^t,\dots,G_i^t,\dots\}$. Определим теперь веса $p(G_i^t)$. Для членов, которые на данном шаге не участвовали в прогнозировании, веса оставляем такими же, какими они были в $\mathcal{G}^{t-1}$. Для членов, участвовавших в прогнозировании, веса доопределяем следующим образом. Пусть s_t - сумма весов всех членов последовательности $\mathcal{G}^{t-1}$, участвовавших в прогнозировании, и r_t - сумма весов той части из этих членов, которые мы оставили в последовательности $\mathcal{G}^t$. Тогда новые веса этих членов мы определяем, умножая прежние веса на s_t/r_t. Очевидно, общая сумма новых весов от этого не изменится, т.е. $\sum_{i=1}^{\infty} p(G_i^t) = s_0$.

Отметим теперь следующее важное свойство "стратегии" Σ, вытекающее непосредственно из ее определения: если Σ на шаге t ошибается, то

$$\frac{s_t}{r_t} \geqslant 2 \qquad (4)$$

и, следовательно, веса членов, участвовавших в прогнозировании и оставленных в $\mathcal{G}^t$, возрастают по крайней мере в 2 раза.

Пусть теперь G_α - первый автоматный граф последовательности $\mathcal{G}$, который совместим с входом $x(1),\dots,x(t)$ и выходом $y(1),\dots,y(t)$, т.е. его дугам можно так приписать выходные буквы, что получится автомат, который

$x(1)\ldots x(t)$ перерабатывает в $y(1)\ldots y(t)$. Очевидно, $|G_\alpha| \leqslant \kappa$, где κ - число состояний "черного ящика". Очевидно, этот автоматный граф мы никогда не будем вычеркивать, и каждый раз, когда $\bar{\Sigma}$ совершает ошибку, с графом G_α произойдет одно из двух: либо какому-нибудь ребру, которому ранее не была приписана выходная буква, мы припишем выходную букву (это в случае, когда на данном шаге G_α не участвует в прогнозировании), либо его вес мы увеличим, по крайней мере, в два раза (это в том случае, когда G_α участвует в прогнозировании). Так как число дуг графа G_α, которым мы можем приписать выходные буквы, не превосходит $a\kappa$ и вес любого графа должен быть меньше $\delta_0 < 1$, то отсюда получаем, что максимальное число ошибок должно быть меньше числа z такого, что $2^{z-a\kappa} \cdot p(G_\alpha) = 1$. Отсюда, используя (I), (2), (3), получаем

$$\bar{\Sigma}^*(\omega, U_a, \kappa) < z \lesssim (a-1)\kappa \log_2 \kappa. \qquad (5)$$

Остается убедиться, что "стратегию" $\bar{\Sigma}$ можно заменить эффективной стратегией Σ, которая с достаточно точным приближением делает то же самое, что $\bar{\Sigma}$. Для этого используем то, что сумма весов конструктивно сходится. Поэтому, рассматривая достаточно длинный, но конечный начальный кусок последовательности $\mathcal{G}^t = \{G_1^t, \ldots, G_i^t, \ldots\}$, мы можем эффективно добиться того, что если мы выдаем в качестве значения $\Sigma(x(1),\ldots,x(t); y(1),\ldots,y(t); x(t+1))$ букву y, то для любой другой буквы y' сумма весов $p(y')$ должна удовлетворять, например, неравенству

$$p(y') \leqslant p(y) + \frac{2}{j_t+1} p(y), \ j_t - \text{число ошибок до момента } t \qquad (6)$$

(вместо точного неравенства $p(y') \leqslant p(y)$ в случае $\bar{\Sigma}$). Отсюда вместо неравенства (4) мы получим неравенство

$$\frac{\delta_t}{\tau_t} \geqslant 2 - \frac{2}{j_t+1}.$$

Но это, как легко убедиться, не влияет на асимптотику (5).

Теорема доказана.

ТЕОРЕМА 2. Пусть $a \geqslant 2$. Существует входная последовательность ω_0 такая, что для любой стратегии Σ и любого $b \geqslant 2$

$$\Sigma^*(\omega_0, U_{ab}, \kappa) \gtrsim (a-1)\kappa \log_2 \kappa,$$

(следовательно, $\Sigma^*(\omega_0, U_a, \kappa) \gtrsim (a-1)\kappa \log_2 \kappa$).

ДОКАЗАТЕЛЬСТВО. Пусть входной алфавит $X=\{x_1,\dots,x_a\}$ и выходной алфавит $Y=\{0,1\}$. Для каждого натурального $\kappa \geqslant 64$ определим следующий класс R_κ автоматов. Автоматы этого класса имеют вид, изображенный на рис.I (на этом рисунке изображены только те дуги, которые существенны для дальнейших рассуждений). Автоматы класса R_κ, как видно из рисунка, имеют

$$s + \mu\alpha = 2[\log_2 \kappa] + 6 + 2^{[\log_2\kappa - \log_2\log_2\kappa]} \cdot [\log_2\kappa - \log_2\log_2\kappa] = \kappa - o(\kappa)$$

состояний. Первые $s-1$ состояний образуют один подавтомат, который назовем κ-шифратором (он у всех автоматов класса R_κ один и тот же), следующие κ состояний образуют другой подавтомат, который назовем основным подавтоматом.

Сначала опишем κ-шифратор. Возьмем двоичную запись числа κ, подставим везде вместо буквы 1 слово x_2x_1, вместо буквы 0 – слово x_1x_2, а в конце полученного слова припишем $x_1x_1x_1$. Полученное таким образом слово обозначим через $\bar{\kappa}$. Очевидно, $\bar{\kappa}$ имеет $s=2[\log_2\kappa]+6$ букв, которые обозначим соответственно через $\nu_1, \nu_2, \dots, \nu_s$. Слово $\bar{\kappa}$ нигде не содержит подряд три раза букву x_1. κ-шифратор характеризуется тем, что он "пропускает" к основному подавтомату только такие слова, которые содержат кусок, равный $\bar{\kappa}$, притом перед этим $\bar{\kappa}$-шифратор должен быть переведен в начальное состояние q_1. Из изображения κ-шифратора (рис.I) и определения слова $\bar{\kappa}$ следует, что κ-шифратор всегда перейдет в состояние q_1, если до подачи $\bar{\kappa}$ будет подана три раза буква x_1 и будет оставаться в этом состоянии все время, пока будет подаваться буква x_1.

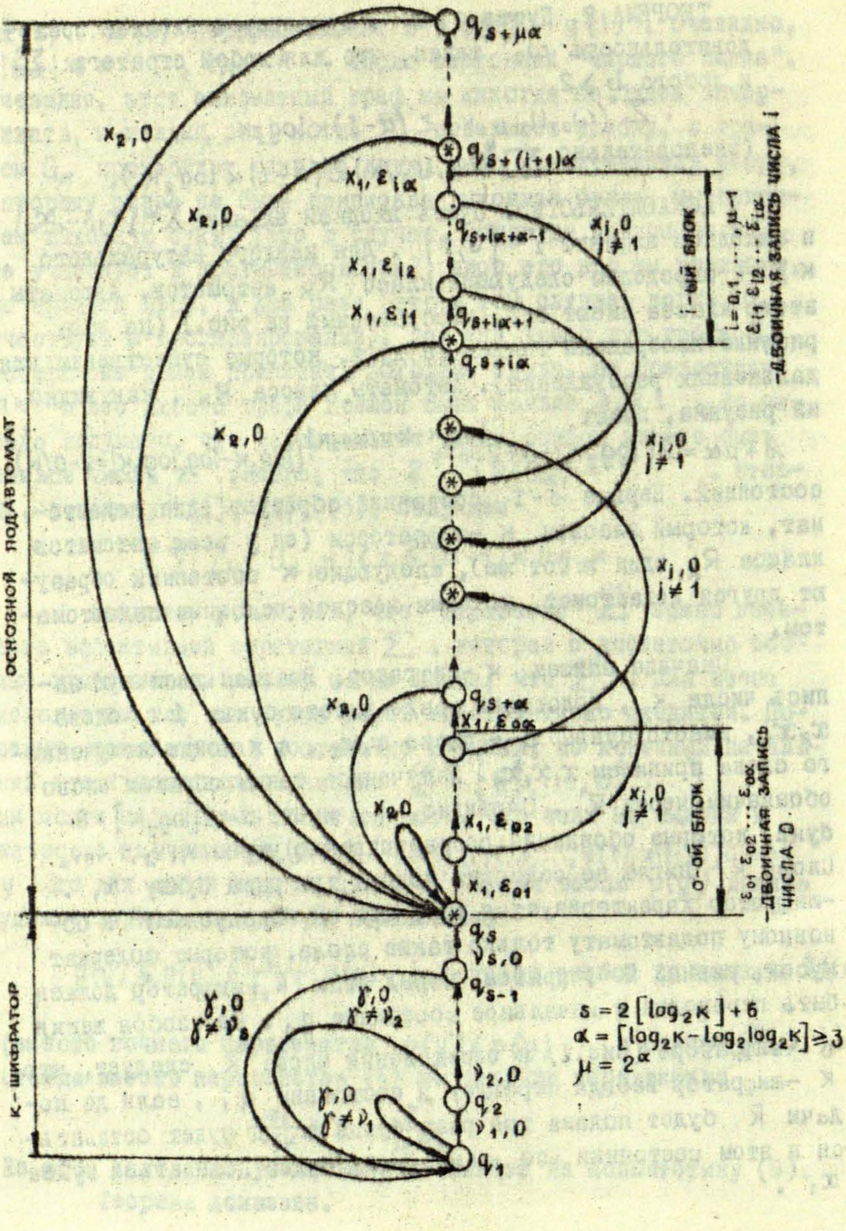

ОСНОВНОЙ ПОДАВТОМАТ
К-ШИФРАТОР
$q_{s+\mu\alpha}$
$q_{s+(i+1)\alpha}$
$q_{s+i\alpha+\alpha-1}$
$q_{s+i\alpha+1}$
$q_{s+i\alpha}$
$q_{s+\alpha}$
q_s
q_{s-1}
q_2
q_1
$x_2, 0$
$x_1, \varepsilon_{i\alpha}$
x_1, ε_{i2}
x_1, ε_{i1}
$x_1, \varepsilon_{0\alpha}$
x_1, ε_{02}
x_1, ε_{01}
$x_j, 0$
$j \neq 1$
$\nu_s, 0$
$\nu_2, 0$
$\nu_1, 0$
$\gamma, 0$
$\gamma \neq \nu_s$
$\gamma \neq \nu_2$
$\gamma \neq \nu_1$
i-ЫЙ БЛОК
$i = 0, 1, \ldots, \mu-1$
$\varepsilon_{i1}\ \varepsilon_{i2} \ldots \varepsilon_{i\alpha}$
-ДВОИЧНАЯ ЗАПИСЬ ЧИСЛА i
0-ОЙ БЛОК
$\varepsilon_{01}\ \varepsilon_{02} \ldots \varepsilon_{0\alpha}$
-ДВОИЧНАЯ ЗАПИСЬ ЧИСЛА 0.
$s = 2\left[\log_2 K\right] + 6$
$\alpha = \left[\log_2 K - \log_2 \log_2 K\right] \geq 3$
$\mu = 2^\alpha$

Рис. 1.

Опишем теперь основной подавтомат. Он состоит из отдельных блоков, i - й блок начинается с состояния $q_{s+i\alpha}$ (на рис.I начальные состояния помечены *), длина каждого блока (т.е. число состояний) $\alpha=[\log_2 k-\log_2\log_2 k]\geqslant 3$, общее число блоков $\mu=2^{\alpha}\leqslant\frac{k}{\log_2 k}$. Выходные пометки на дугах, исходящих из состояний i -го блока и помеченных буквой x_1, образуют двоичное слово $\varepsilon_{i1}\varepsilon_{i2}\ldots\varepsilon_{i\alpha}$, являющееся двоичной записью числа i (вначале столько нулей , чтобы общая длина была α). Слово $\varepsilon_{i1}\varepsilon_{i2}\ldots\varepsilon_{i\alpha}$ будем называть характеристической последовательностью блока i. Таким образом, каждому блоку соответствует своя характеристическая последовательность. Еще заметим, что число двоичных слов длины α в точности совпадает с числом блоков; следовательно, любое двоичное слово длины α является характеристической последовательностью некоторого блока. Дуги, исходящие из начальных состояний блоков (т.е. из состояний $q_{s+i\alpha}, i=0,1,\ldots,\mu$) и помеченные входной буквой x_2, подсоединены к состоянию q_s. Дуги, исходящие из состояний, лежащих внутри блоков, и помеченные входными буквами, отличными от x_1 (такие дуги назовем переменными), подсоединены к произвольным начальным состояниям блоков (таких состояний всего 2^{α}). Именно этим одни автоматы класса R_k отличаются от других автоматов этого же класса.

Рассмотрим теперь переменные дуги. Их всего $u=(a-1)\mu(\alpha-1)$. Упорядочим эти дуги: $d_1, d_2, \ldots, d_u$. С каждым основным подавтоматом ассоциируется последовательность $\delta_{11},\ldots,\delta_{1\alpha},\ldots\ldots,\delta_{j1},\ldots,\delta_{j\alpha},\ldots\ldots,\delta_{u1},\ldots,\delta_{u\alpha}$ из 0 и 1 длины αu, которая получается следующим образом: $\delta_{j1},\ldots,\delta_{j\alpha}$ - характеристическая последовательность того блока, к начальному состоянию которого подсоединена дуга d_j. Такую последовательность будем называть характеристической последовательностью данного основного подавтомата. Легко видеть (учитывая значения α и μ), что любая последовательность из 0 и 1 длины αu является характеристической последовательностью некоторого основного подавтомата.

Определим теперь одну специальную входную последовательность. Через d'_j обозначим выходную букву, которая приписана дуге d_j, а через V_j - последовательность, которая состояние q_{V_1} переводит в то состояние, из которого исходит дуга d_j. Положим

$$D_K=\{V_1,d'_1,\underbrace{x_1,\ldots,x_1}_{\alpha},x_2,\ldots\ldots,V_j,d'_j,\underbrace{x_1,\ldots,x_1}_{\alpha},x_2,\ldots\ldots,V_u,d'_u,\underbrace{x_1,\ldots,x_1}_{\alpha}\}.$$

Рассмотрим основной подавтомат как самостоятельный автомат с начальным состоянием q_{V_1}. Подадим на его вход последовательность D_K. Тогда он выдаст выходную последовательность

$$E=\{W_1,0,\delta_{11},\ldots,\delta_{1\alpha},0,\ldots\ldots,W_j,0,\delta_{j1},\ldots,\delta_{j\alpha},0,\ldots\ldots,W_u,0,\delta_{u1},\ldots,\delta_{u\alpha}\},$$

где W_j - выходная последовательность, соответствующая куску V_j, а $\delta_{j1},\ldots,\delta_{j\alpha}$ - выходная последовательность, соответствующая куску $\underbrace{x_1,\ldots,x_1}_{\alpha}$, следующему непосредственно после V_j. Очевидно, подпоследовательность

$$\delta_{11},\ldots,\delta_{1\alpha},\ldots\ldots,\delta_{u1},\ldots,\delta_{u\alpha}$$

последовательности E есть характеристическая последовательность данного основного подавтомата.

Пусть Σ - произвольная стратегия. Теперь нетрудно показать, что существует основной подавтомат A_Σ такой, что если его рассматривать как самостоятельный автомат с начальным состоянием q_{V_1}, то стратегия Σ, примененная к автомату A_Σ и входной последовательности D_K, сделает ошибки во всех тех местах, которые соответствуют кускам $\underbrace{x_1,\ldots,x_1}_{\alpha}$ последовательности D_K, т.е. для любых j и l, $1\leqslant l\leqslant\alpha$, $1\leqslant j\leqslant u$, выполнится неравенство

$$\Sigma(V_1,d'_1,\underbrace{x_1,\ldots,x_1}_{\alpha},x_2,\ldots\ldots,V_j,d'_j,\underbrace{x_1,\ldots,x_1}_{l-1};$$
$$W_1,0,\delta_{11},\ldots,\delta_{1\alpha},0,\ldots\ldots,W_j,0,\delta_{j1},\ldots,\delta_{j(l-1)};x_1)\neq\delta_{jl}.$$

Это неравенство показывает, как надо определить характеристическую последовательность искомого автомата A_Σ. Далее, по характеристической последовательности однозначно можно восстановить сам автомат A_Σ. Таким образом,

$$\sum^{*}(D_{\kappa}, A_{\Sigma}) \geqslant u\alpha = (a-1)\mu(\alpha-1)\alpha .$$

Определим теперь искомую входную последовательность следующим образом:

$$\omega_0 = \left\{ \bar{K}_0, D_{K_0}, (\overline{K_0+1}), D_{K_0+1}, \ldots\ldots, \bar{K}, D_K, \ldots\ldots \right\}, K_0 = 64.$$

Пусть A - произвольный автомат из класса R_K. Из определения K-шифратора следует, что с помощью последовательности ω_0 мы впервые достигнем состояния q_s автомата A после подачи начального куска

$$\omega_0(K) = \left\{ \bar{K}_0, D_{K_0}, \ldots\ldots, D_{K-1}, \bar{K} \right\}.$$

До этого мы все время будем находиться в K-шифраторе (последовательность, которую при этом выдаст K-шифратор, обозначим через $\Omega_0(K)$). Последовательность ω_0 построена так, что далее следует кусок D_K. Таким образом, после подачи $\omega_0(K)$ мы будем находиться в точности в такой же ситуации, как если бы мы основной подавтомат рассматривали как самостоятельный автомат с начальным состоянием q_s, подавали входную последовательность D_K и прогнозировали согласно следующей стратегии:

$$\sum{}'(\gamma; \beta; x_l) = \sum(\omega_0(K), \gamma; \Omega_0(K), \beta; x_l).$$

Пусть теперь для автомата A в качестве основного подавтомата выбран автомат $A_{\Sigma'}$. Тогда, очевидно,

$$\sum^{*}(\omega_0, A) \geqslant \sum{}'^{*}(D_K, A_{\Sigma'}) \geqslant (a-1)\mu(\alpha-1)\alpha .$$

Поэтому $\sum^{*}(\omega_0, U_{a,2}, s+\mu\alpha) \geqslant (a-1)\mu(\alpha-1)\alpha$.
Отсюда, учитывая значения s, α, μ, получаем

$$\sum^{*}(\omega_0, U_{a,2}, K) \gtrsim (a-1) K \log_2 K .$$

Теорема доказана.

Вместо прогнозирования поведения автоматов можно рассматривать также и предельный синтез автоматов. В этом случае по паре $\{x(1), \ldots, x(t)\}$, $\{y(1), \ldots, y(t)\}$ требуется построить автомат A', который на последовательности $\omega = \{x(1), \ldots, x(t), \ldots\}$ неотличим от "черного ящика" A . Совокупность всех таких A' обозначим через $\{A_\omega\}$. В этом случае под стратегией Σ понимается общерекурсивная функция, которая каждому кортежу $\langle x(1),\dots,x(t),y(1),\dots,y(t)\rangle$ сопоставляет некоторый автомат из U_a (или более точно: номеру кортежа сопоставляет номер автомата из U_a). $A_t = \Sigma(x(1),\dots,x(t);y(1),\dots,y(t))$ будем называть гипотезой, порождаемой в момент t. Пусть:

а) для любого t автомат A_t входное слово $x(1)\dots x(t)$ прорабатывает в $y(1)\dots y(t)$, т.е. A_t не является "явно" неверной гипотезой;

б) существует такое τ, что

$$A_\tau = A_{\tau+1} = \dots = A' \quad \text{и} \quad A' \in \{A_\omega\}.$$

В таком случае будем говорить, что стратегия Σ синтезирует в пределе автомат A на последовательности ω . В этом случае через $\Sigma^{\#}(\omega,A)$ будем обозначать число изменений гипотез, т.е. число различных t , при которых $A_t \neq A_{t+1}$. В остальных случаях $\Sigma^{\#}(\omega,A) = \infty$. Величина $\Sigma^{\#}(\omega, U_a, \kappa)$ определяется аналогично $\Sigma(\omega,U_a,\kappa)$, только вместо $\Sigma^{*}(\omega,A)$ берется $\Sigma^{\#}(\omega,A)$.

ТЕОРЕМА 3. Пусть $a \geqslant 2$. Существует стратегия Σ такая, что для любой входной последовательности ω

$$\Sigma^{\#}(\omega,U_a,\kappa) \lesssim (a-1)\,\kappa \log_2 \kappa .$$

ДОКАЗАТЕЛЬСТВО. Рассмотрим вместо стратегии $\bar{\Sigma}$ из доказательства теоремы I стратегию $\bar{\Sigma}'$, которая делает все то же самое, что $\bar{\Sigma}$, за исключением того, что последовательность $\mathcal{Y}^t$ она меняет только в те моменты t , в которые происходят ошибки, т.е. $y(t+1) \neq \bar{\Sigma}'(x(1),\dots,x(t);y(1),\dots,y(t);x(t+1))$. Легко проверить, что асимптотика (5) сохраняет силу и в этом случае, так как при ее выводе мы фактически используем только те моменты t , в которые стратегия ошибается (т.е. выполняется неравенство (4)). Моменты, в которые происходят ошибки, обозначим соответственно через $t_1,t_2,\dots,t_n$, $n = \bar{\Sigma}'^{*}(\omega,A)$. Таким образом, наша стратегия Σ' будет использовать только последовательности $\mathcal{Y}, \mathcal{Y}^{t_1}, \mathcal{Y}^{t_2}, \ldots, \mathcal{Y}^{t_n}$.

Рассмотрим теперь построение эффективной стратегии Σ' по стратегии Σ'. Букве y, которую стратегия Σ' выдает в момент t, должна удовлетворять неравенству (6). Пусть $t \in (t_i, t_{i+1}]$. Тогда неравенство (6) принимает вид

$$p(y') \leq p(y) + \frac{2}{i} p(y). \tag{7}$$

Для каждого $t \in (t_i, t_{i+1}]$ букву y можно определить, используя только конечный начальный кусок последовательности $\mathcal{Y}^{t_i}$. Этот начальный кусок назовем существенным для данного момента t. Учитывая конструктивную сходимость ряда $\sum_j p(\mathcal{Y}_j)$, нетрудно убедиться, что он эффективно определяется по паре $\{x(1), \ldots, x(t)\}, \{y(1), \ldots, y(t)\}$. Заметим, что если существенный начальный кусок для момента t выбран достаточно длинным, то он может оказаться существенным также и для следующего момента и т.д. Пусть теперь $t_{i_1}, t_{i_2}, \ldots, t_{i_m}$ - моменты из промежутка $(t_i, t_{i+1}]$, в которые мы вынуждены менять (удлинять) выбранный ранее существенный начальный кусок (чтобы можно было проверить неравенство (7)). Далее заметим, что если мы уже дошли до существенного начального куска, который содержит искомый граф G_α, и этот кусок за G_α содержит еще достаточно длинный "хвост", то по крайней мере по причине неравенства (7) его менять не потребуется (если он будет меняться, то только за счет ошибки, когда меняется само $\mathcal{Y}^t$). Отсюда следует, что если каждый следующий существенный начальный кусок мы будем выбирать достаточно длинным по сравнению с предыдущим (например, длины 2^n, где n - длина предыдущего куска), то мы можем добиться того, что общее число изменений существенных начальных кусков за счет неравенства (7) не превзойдет $O(|G_\alpha| \log_2 |G_\alpha|)$. А число изменений существенных начальных кусков за счет изменения самой последовательности $\mathcal{Y}^t$ равно числу изменений последовательностей $\mathcal{G}^t$, т.е. равно числу ошибок, совершенных стратегией Σ', а это число, как следует из доказательства теоремы I, не превосходит $(a-1)k\log_2 k + o(k\log_2 k)$. Отсюда мы получаем, что описанная стратегия Σ' существенные начальные куски меняет не более чем $(a-1)k\log_2 k + o(k\log_2 k) + o(|G_\kappa|\log_2|G_\kappa|) = (a-1)k\log_2 k + o(k\log_2 k)$ раз. В промежуток времени, когда существенный начальный кусок не меняется, стратегия Σ' прогнозирует следующее значение, используя только этот начальный кусок и текущую вершину каждого графа из этого куска. Более точно это означает следующее. Текущая вершина графа G_j в момент t – это та вершина графа G_j, в которую слово $x(1)\dots x(t)$ переводит начальную вершину. Таким образом, если для графа G_j мы знаем текущую вершину, то мы можем определить букву y, которую G_j (как автомат) выдает после подачи $x(t+1)$, если до этого было подано $x(1)\dots x(t)$; информацию о самом слове $x(1)\dots x(t)$ мы можем не хранить.

Отсюда следует, что прогнозирование, выполняемое стратегией Σ' в промежуток времени, когда существенный начальный кусок не меняется, может выполнять конечный автомат. Состояниями этого автомата являются всевозможные расстановки текущих вершин в выбранном начальном куске (т.е. каждое состояние – это выбранный начальный кусок, где в каждом графе из этого куска отмечена одна вершина, называемая текущей; одни состояния от других отличаются выбором текущих вершин). Переход из одного состояния в другое осуществляется в соответствии с переводом текущих вершин в каждом графе при подаче буквы x. В качестве выхода берется та буква, которую должна выдать стратегия Σ' в данной ситуации.

Этот автомат и является той гипотезой, которую в рассматриваемый промежуток времени выдает искомая стратегия Σ. Очевидно, число изменений гипотез равно числу изменений существенных начальных кусков. Следовательно,

$\Sigma^*(\omega, A) \leqslant (a-1)k\log_2 k$. Теорема доказана.

Ясно, что нижняя оценка, доставляемая теоремой 2, сохраняет силу и в данном случае.

Рассмотренные выше случаи имели аналогию с простым экспериментом. Рассмотрим теперь случай, который имеет аналогию с кратным экспериментом. Пусть на вход "черного ящика" A подается последовательность входных слов

$$\Omega = \{\varphi_1, \varphi_2, \dots, \varphi_t, \dots\}$$

и пусть $\{\eta_1, \eta_2, \dots, \eta_t, \dots\}$ - соответствующая последовательность выходных слов (подача каждого слова начинается с начального состояния автомата A).

При прогнозировании по тройке $\{\varphi_1, \dots, \varphi_t\}, \{\eta_1, \dots, \eta_t\}, \varphi_{t+1}$ требуется предсказать η_{t+1}. В данном случае $\Sigma^*(\Omega, A)$ означает число различных t, при которых

$$\Sigma(\varphi_1, \dots, \varphi_t; \eta_1, \dots, \eta_t; \varphi_{t+1}) \neq \eta_{t+1}.$$

При предельном синтезе по паре $\{\varphi_1, \dots, \varphi_t\}, \{\eta_1, \dots, \eta_t\}$ требуется определить автомат A', который на входных словах из Ω неотличим от A. Величина $\Sigma^*(\Omega, A)$ определяется аналогично $\Sigma^*(\omega, A)$, только вместо букв $x(t)$, $y(t)$ берутся соответственно слова φ_t, η_t и гипотеза A_t определяется как $A_t = \Sigma(\varphi_1, \dots, \varphi_t; \eta_1, \dots, \eta_t)$.

Повторяя те же самые рассуждения, которыми мы пользовались при доказательстве теорем I и 3, можно доказать следующие несколько более общие утверждения.

ТЕОРЕМА I'. Пусть $a \geqslant 2$. Существует стратегия Σ такая, что для любой последовательности входных слов Ω

$$\Sigma^*(\Omega, U_a, k) \leqslant (a-1)k\log_2 k.$$

ТЕОРЕМА 3'. Пусть $a \geqslant 2$. Существует стратегия Σ такая, что для любой последовательности входных слов Ω

$$\Sigma^*(\Omega, U_a, k) \leqslant (a-1)k\log_2 k.$$

Теорема 3' играет важную роль при исследовании синтеза программ по историям их работы (см. [6]).

ЛИТЕРАТУРА

1. Барздинь Я.М. О расшифровке автоматов при отсутствии верхней оценки числа состояний.- "ДАН СССР", 1970, 190, № 5.
2. Barzdin J.M. Prognostication of automata and functions, - IFIP Congress 71, Ljubljana, 1971.
3. Трахтенброт Б.А., Барздинь Я.М. Конечные автоматы (поведение и синтез, М., 1970.
4. Коршунов А.Д. Об асимптотических оценках числа конечных автоматов.- "Кибернетика", 1967, № 2.
5. Барздинь Я.М., Фрейвалд Р.В. Прогнозирование и предельный синтез эффективно перечислимых классов функций.- Настоящий сборник, стр. 101-111.
6. Барздинь Я.М. Замечание о синтезе программ по историям их работы. - Настоящий сборник, стр. 145-151.

ЗАМЕЧАНИЕ О СИНТЕЗЕ ПРОГРАММ ПО ИСТОРИЯМ ИХ РАБОТЫ *)

Я.М.Барздинь

Синтез программ по историям их работы, по-видимому, является одной из наиболее важных проблем теории обучаемости. Даже обучение таким алгоритмам, как сложение и умножение, обычно происходит следующим образом: учитель разъясняет их работу на отдельных примерах, т.е. сообщает истории их работы на этих примерах, а затем ученики сами на основании этой информации синтезируют общие алгоритмы (программы). Недавно Бирманом [2, 3] были предложены эвристические алгоритмы синтеза программ по историям их работы, и эти алгоритмы были проверены с помощью электронной вычислительной машины. Однако математические основы такого процесса синтеза почти не исследованы. Именно ввиду важности этой проблемы мы решили выделить полученные нами результаты в виде отдельной статьи, хотя с математической точки зрения они являются простыми следствиями работы [4].

В качестве модели мы будем рассматривать машины Поста. Все результаты, которые здесь будут получены, легко переносятся и на более общие языки программирования, только в оценках может добавиться константный множитель.

Будем рассматривать однолентoчные машины Поста с внешним алфавитом $\{0,1\}$. Они задаются командами вида: $\leftarrow$ - сдвиг головки на одну ячейку влево, $\rightarrow$ - сдвиг головки на одну ячейку вправо, V - печатание I в обозреваемой ячейке, 0 - печатание 0 в обозреваемой ячейке, ? - условная команда: переход по I при наличии I в обозреваемой ячейке, переход по 0 в остальных случаях, ! - команда "СТОП". На рис.I изображен пример программы. Эта программа

---

*) Результаты настоящей статьи были опубликованы без доказательств в [I].

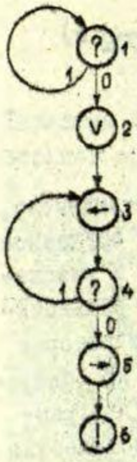


Рис.I.

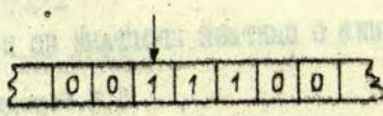


Рис.2.

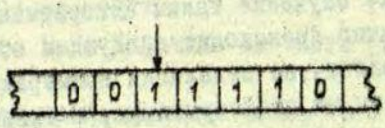


Рис.3.

$x = III$ (рис.2) перерабатывает в $y = 1111$ (рис.3). При этом программа выполняет следующую последовательность команд:

$$? \rightarrow ? \rightarrow ? \rightarrow ?\ V \leftarrow\ ? \leftarrow ? \leftarrow ? \leftarrow ? \rightarrow\ !$$

Она составлена из всех команд, которые последовательно проходит данная программа при переработке x . Такую последовательность мы будем называть операционно-логической историей данной программы на x (понятие введено в [5]).

Сформулируем теперь исследуемую проблему. Пусть P - произвольная программа машины Поста и

$$\Omega = \{x_1, x_2, \ldots, x_t, \ldots\}$$

- бесконечная последовательность натуральных чисел. Будем считать, что на всех этих числах x_t программа P когда-нибудь останавливается и выдает результат $P(x_t)$(такие Ω будем называть допустимыми для P). Пусть h_t - операционно-логическая история программы P на x_t. Пусть дано

$$\{(x_1, h_1), \ldots, (x_t, h_t)\}.$$

Используя эту информацию, требуется определить программу

P' , которая на Ω совпадает с P , т.е. $P'(x)=P(x)$ для $x \in \Omega$. Совокупность всех таких P' обозначим через $\{P_\Omega\}$. Под стратегиями будем понимать общерекурсивные функции Π, которые кортежам вида $< x_1, h_1, \dots, x_t, h_t >$ сопоставляют программы машин Поста. Программу $P_t = \Pi(x_1, h_1, \dots, x_t, h_t)$ будем называть гипотезой, порождаемой в момент t . Пусть:

а) для любого t программа P по крайней мере на $x_1, x_2, \dots, x_t$ совпадает с P ,

б) существует такое τ , что

$$P_\tau = P_{\tau+1} = \dots = P' \quad и \quad P' \in \{P_\Omega\}.$$

В таком случае будем говорить, что стратегия Π синтезирует в пределе программу P на последовательности Ω по операционно-логическим историям. В этом случае через $\Pi^*(\Omega,P)$ будем обозначать число изменений гипотез, т.е. число различных t , при которых $P_t \neq P_{t+1}$. В остальных случаях $\Pi^*(\Omega, P) = \infty$. Наша цель - оценить величину $\Pi^*(\Omega, P)$. Через $\|P\|$ обозначим число условных команд, входящих в программу P .

ТЕОРЕМА I. Существует стратегия Π такая, что для любой программы P и любой последовательности Ω

$$\Pi^*(\Omega,P) \leq \|P\| \log_2 \|P\|.$$

При использовании достаточно развитых алгоритмических языков для реальных программ величина $\|P\|$ обычно относительно небольшая. Так, например, для программы умножения двух матриц $\|P\| = 3$. Таким образом, теорема I показывает, что существует стратегия, которая в процессе синтеза делает весьма небольшое число ошибок (почти соизмеримое с тем, что делает средний программист при составлении аналогичных программ).

Для доказательства теоремы I каждой программе P мы сопоставим следующий автомат $P_{авт.}$ с входным алфавитом $\{0,1\}$. Пусть программа P начинается с условной команды (это обстоятельство не ограничивает общности). Программу P представим себе как граф. Оставим в этом графе те вершины, которые соответствуют командам вида "?" и "!". Пути, составленные из остальных вершин, заменим дугами. Точнее, если путь имеет вид, изображенный на рис.4а, то его заменим дугой с входной пометкой ε и выходной пометкой $(\gamma_1,\gamma_2,\dots,\gamma_s,\delta)$ (рис.4б). В результате мы получим диаграмму некоторого автомата, который обозначим через $P_{авт.}$. Для программы, изображенной на рис.I, соответствующий автомат $P_{авт.}$ имеет вид, изображенный на рис.5; его выходной алфавит: $\{(\to,?),(\vee,\leftarrow,?),(\leftarrow,?),(\to,!)\}$.

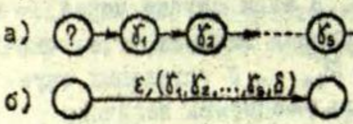


Рис.4.

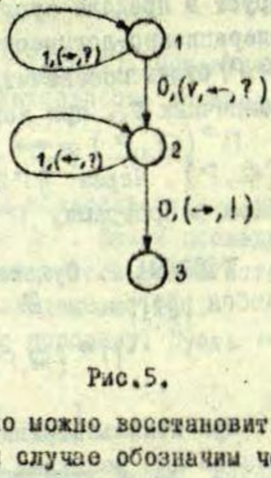


Рис.5.

Очевидно, по $P_{авт.}$ однозначно можно восстановить саму программу P (которую в данном случае обозначим через $(P_{авт.})_{прогр.}$).

Теперь с элементом x_t последовательности Ω и историей $h_t = ?u_1?u_2?\dots?u_{l_t}$, где u_i – последовательность команд (безусловных), лежащих между двумя условными командами, ассоциируется входное слово $\varphi_t=\varepsilon_{t1}\varepsilon_{t2}\dots\varepsilon_{tl_t}$ ($\varepsilon_{tj}\in\{0,1\}$) следующим образом: $\varepsilon_{t1},\varepsilon_{t2},\dots,\varepsilon_{tl_t}$ – это последовательность значений, которые принимают условные команды при переработке x_t согласно истории h_t (мы считаем, что условная команда "?" принимает значения I, если в обозреваемой ячейке записано I, и 0 в остальных случаях). Другими словами, слово φ_t в диаграмме автомата $P_{авт.}$, задает тот же самый путь, что и слово x_t с историей h_t в программе P. Заменяя теперь в последовательности $\Omega=\{x_1,\dots,x_t,\dots\}$ элементы x_t словами φ_t, мы получим последовательность входных слов

$$\Omega'=\{\varphi_1,\dots,\varphi_t,\dots\}.$$

Теперь нетрудно убедиться в справедливости следующего основного утверждения, устанавливающего связь между синтезом автоматов и программ:

А. Если A - некоторый автомат, который неотличим от $P_{авт.}$ на последовательности входных слов Ω' (подача каждого слова начинается с начального состояния 1), то программа $(A)_{прогр.}$, получаемая из автомата A, неотличима от программы P на последовательности Ω (т.е. имеет такие же истории и выдает такие же результаты).

Применим теперь к последовательности Ω и автомату $P_{авт.}$ стратегию Σ из теоремы 3' работы [4]. Получим

$$\Sigma^*(\Omega', P_{авт.})\leq |P_{авт.}|\log_2|P_{авт.}|-(\|P\|+1)\log_2(\|P\|+1). \quad (I)$$

Стратегия Σ для вычисления $(t+1)$- й гипотезы использует кортеж $K_t=\langle\varphi_1,\dots,\varphi_t,P_{авт.}(\varphi_1),\dots,P_{авт.}(\varphi_t)\rangle$. Однако искомая стратегия Π для вычисления $(t+1)$- й гипотезы может использовать только кортеж $N_t=\langle x_1,h_1,\dots,x_t,h_t\rangle$. Но по этому кортежу, как нетрудно убедиться, эффективно можно построить кортеж K_t. Поэтому стратегия Π работает следующим образом: по кортежу N_t находит кортеж K_t, затем к кортежу K_t применяет стратегию Σ и находит гипотезу $A_t=\Sigma(K_t)$, далее, автомат A_t преобразует в программу $(A_t)_{прогр.}$ и эту программу выдает в качестве результата на кортеже N_t. Из утверждения А и соотношения (I) следует, что

$$\Pi^{*}(\Omega, P) \leq \|P\| . \log_2 \|P\| .$$

Теорема доказана.

ЗАМЕЧАНИЕ. Фактически мы доказали несколько более сильное утверждение: построенная стратегия Π синтезирует в пределе программу P', которая на последовательности Ω не только выдает такие же значения, как P, но, кроме того, такие же операционно-логические истории.

Рассмотрим теперь вместо операционно-логических историй так называемые операционные истории [5]. Обычно они являются той минимальной информацией, которая сообщается ученику в процессе обучения того или иного алгоритма. Они получаются из операционно-логических историй, если вычеркнуть все условные команды. Так, например, операционная история, соответствующая рассмотренному выше примеру, имеет вид $\rightarrow \rightarrow \rightarrow \vee \leftarrow \leftarrow \leftarrow \leftarrow \rightarrow !$. В этом случае величину, задающую число изменений гипотез, обозначим через $\Pi^{\#}(\Omega, P)$. Через $|P|$ обозначим число всех команд, входящих в программу P. Имеет место

ТЕОРЕМА 2. Существует стратегия Π такая, что для любой программы P и любой последовательности Ω

$$\Pi^{\#}(\Omega, P) \leq |P| \log_2 |P| .$$

Справедливость теоремы 2 легко вытекает из теоремы I. Для этого достаточно заметить, что любую программу P можно преобразовать в эквивалентную программу P', вставляя между любыми двумя безусловными командами $\bigcirc \longrightarrow \bigcirc$ условную команду "?": $\bigcirc \longrightarrow ? \rightrightarrows \bigcirc$. Очевидно, $\|P'\| \leq |P|$. При этом операционные истории програм P' и P совпадают. Но для программы P' по операционной истории $k_1 k_2 \ldots k_s !$ однозначно можно восстановить операционно-логическую историю: $? k_1 ? k_2 ? \ldots ? k_s !$. Следовательно, для программы P' можно применить стратегию Π из теоремы I. Получаем оценку $\Pi^{\#}(\Omega, P') \leq \|P'\| \log_2 \|P'\| \leq |P| \log_2 |P|$. Теорема доказана.

Вопрос о том, имеет ли место в случае операционных историй точный аналог теоремы I, остается открытым.

Также представляет интерес исследование предельного синтеза программ с небольшим параметром $\| P \|$; приведенные асимптотики для этого случая мало пригодны.

ЛИТЕРАТУРА

I. Барздинь Я.М. О синтезе программ по отдельным примерам. - Теория программирования. Труды симпозиума, Новосибирск, 1972.
2. Biermann A.W. On the inference of Turing Machines from sample computations. Artificial intelligence, 1972.
3. Biermann A.W., Baum R., Krishnaswamy R., Petry F.E. Automatic program synthesis. - Technical report, The Ohio state university, 1973.
4. Барздинь Я.М. Прогнозирование и предельный синтез конечных автоматов. - Настоящий сборник, стр.129-144.
5. Ershov A.P., Theory of program achemata, IFIP Congress 71, Ljubljana, 1971, 1.

ПОСТРОЕНИЕ ПОЛНОЙ СИСТЕМЫ ПРИМЕРОВ ДЛЯ ПРОВЕРКИ КОРРЕКТНОСТИ ПРОГРАММ

Я.М.Барздинь, Я.Я.Бичевский, А.А.Калниньш

1° Проверка корректности (отладка) программ является наиболее трудоемкой частью на пути автоматизации программирования. Один из подходов для решения этой проблемы был намечен в работах Дж.Маккарти [1] и Д.Скотта [2] и затем развит в более практическом плане в работах [3], [4] и др. В этих работах речь идет об аксиоматизации процесса вычислений. Однако практическая реализация этого подхода пока еще связана со значительными трудностями. Поэтому в настоящий момент и, наверное, в ближайшем будущем программисты будут пользоваться старым и проверенным способом отладки программ, состоящим в том, что каждый раз строится некоторая система примеров, и программа проверяется на этой системе примеров. Если отлаживаемая программа сработает правильно на этой системе примеров, то программист обычно считает, что программа составлена правильно. Подбор хорошей системы примеров является наиболее тонкой частью такого процесса отладки. Обычно программист старается найти такую систему примеров, чтобы на ней можно было пройти все ветви программы. И если на такой системе примеров, называемой в дальнейшем полной, программа сработает правильно, то у нас обычно есть большая уверенность в том, что программа будет правильно работать и на других примерах. Конечно, этот критерий не является абсолютным, но на практике им пользуются достаточно успешно.

Таким образом, основной проблемой для автоматизации описанного процесса отладки является задача автоматического построения полной системы примеров по заданной программе. Сразу ясно, что в общем случае проблема построения полной системы примеров алгоритмически неразрешима (см.также [5]). Цель настоящей статьи - доказать, что для весьма широкого круга задач типа обработки данных эта проблема разрешима. Для формализации существенной части (с точки зрения нашей проблемы) процесса обработки данных мы введем некоторую абстрактную машину (типа машины Тьюринга), работающую с массивами. Эта машина будет называться базовой машиной, а язык программирования, порожденный ею, - базовым языком. В конце статьи будет показано, каким образом, используя алгоритм построения полной системы примеров для программ на базовом языке, можно построить полную систему примеров для достаточно широкого класса программ, записанных на КОБОЛе.

Определим базовую машину (рис.I). Она состоит из управляющего блока, конечного числа односторонних входных лент $A, B, \ldots C$ и конечного числа односторонних выходных лент $U, V, \ldots Z$. Управляющий блок содержит конечное число ячеек $a, b, \ldots, v$, называемых внутренними ячейками.

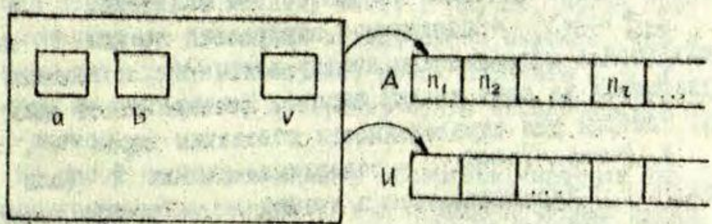


Рис.I.

В каждую внутреннюю ячейку можно записывать произвольное целое число. Входные и выходные ленты состоят также из ячеек; в каждую ячейку также может записываться произвольное целое число. Ленты будут использоваться для изображения массивов. Под массивами мы будем понимать конечные последовательности целых чисел. Будем говорить, что на ленте записан массив $(n_1, n_2, \ldots, n_k)$, если, начиная с первой ячейки, записаны числа $n_1, n_2, \ldots, n_k$, а остальные ячейки - пустые (см.рис.I). Числа $n_1, n_2, \ldots, n_k$, следуя традиции, будем называть записями массива. В дальнейшем, говоря о массиве X, будем понимать массив, записанный на ленте X.

Определим теперь команды, которые может выполнять машина. Эти команды в дальнейшем будем называть базовыми командами. Пусть X - обозначение произвольной входной ленты, Y - обозначение произвольной выходной ленты, t, v - обозначения произвольных внутренних ячеек, c - произвольное целое число (константа).

I. $X \Longrightarrow t$ - содержимое обозреваемой ячейки ленты X переписывается во внутреннюю ячейку t (т.е. ячейке t присваивается очередная запись массива X), и головка передвигается на одну ячейку вправо. Команда имеет два выхода: выход "+", когда обозреваемая ячейка содержит целое число, и выход "-", когда обозреваемая ячейка пустая. В последнем случае значение ячейки t не меняется.

2. $t \Longrightarrow Y$ - содержимое внутренней ячейки t переписывается в обозреваемую ячейку ленты Y, а головка передвигается на одну ячейку вправо. Команда имеет один выход, который для определенности обозначим через "+".

3. $t \Longrightarrow v$ $(c \Longrightarrow v)$ - содержимое ячейки t (или константа c) переписывается в ячейку v. Команда имеет один выход, который также обозначим через "+".

4. $t < v$ $(t < c)$ - команда имеет два выхода: если содержимое ячейки t меньше содержимого ячейки v (или константы c), то управление передается по выходу "+", в противном случае по выходу "-".

5. НОП - ничего не делается (нет операции); команда имеет один выход "+".

6. СТОП - машина останавливается. Команда не имеет выходов.

Команды I и 4, имеющие два выхода, мы будем называть условными командами, остальные – безусловными командами.

Содержательно под программой мы будем понимать программу, записанную в указанной системе команд. Формально программу мы определим как четверку $\{\mathcal{X}, \mathcal{Y}, \mathcal{Z}, \mathcal{T}\}$, где $\mathcal{X}$ – алфавит обозначений входных лент (например, $\mathcal{X} = \{A, B, \ldots, C\}$), $\mathcal{Y}$ – алфавит обозначений выходных лент (например $\mathcal{Y} = \{U, V, \ldots, Z\}$), $\mathcal{Z}$ – алфавит обозначений внутренних ячеек (например, $\mathcal{Z} = \{a, b, \ldots, v\}$), $\mathcal{T}$ – граф-схема, составленная из описанных выше команд. При построении граф-схемы допускается, что выходы некоторых команд остаются свободными, т.е. не ведут ни в одну команду программы (например, на рис.2 выход "–" первой команды). Кроме того, предполагается, что все команды (точнее, все вхождения команд в граф-схему) перенумерованы, и выполнение программы всегда начинается с первой команды. Будем считать, что в момент запуска программы все внутренние ячейки содержат нули. Программа останавливается, когда она попадает на команду СТОП. В случае, когда программа попадает на свободный выход, будем считать, что с ней происходит авария. В случае, когда алфавиты $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$ будут ясны из контекста, программу будем отождествлять с ее граф-схемой.

На рис.2 изображен пример программы, которая из двух упорядоченных массивов A и B (упорядоченным будем считать массив, в котором записи следуют в возрастающем порядке) образует новый упорядоченный массив U , содержащий все записи массивов A и B . Если массив A пустой, то происходит авария (выход "–" первой команды – свободный). Программа написана ошибочно: если в массивах A и B встречаются одинаковые записи, то программа зацикливается.

2°. Переходим к определению полной системы примеров. Под путем программы мы будем понимать любую последовательность вида $(a_1, \varepsilon_1, a_2, \varepsilon_2, \ldots, a_{l-1}, \varepsilon_{l-1}, a_l)$, где a_i , $i = 1, 2, \ldots, l$ – команды данной программы (точнее, их номера),

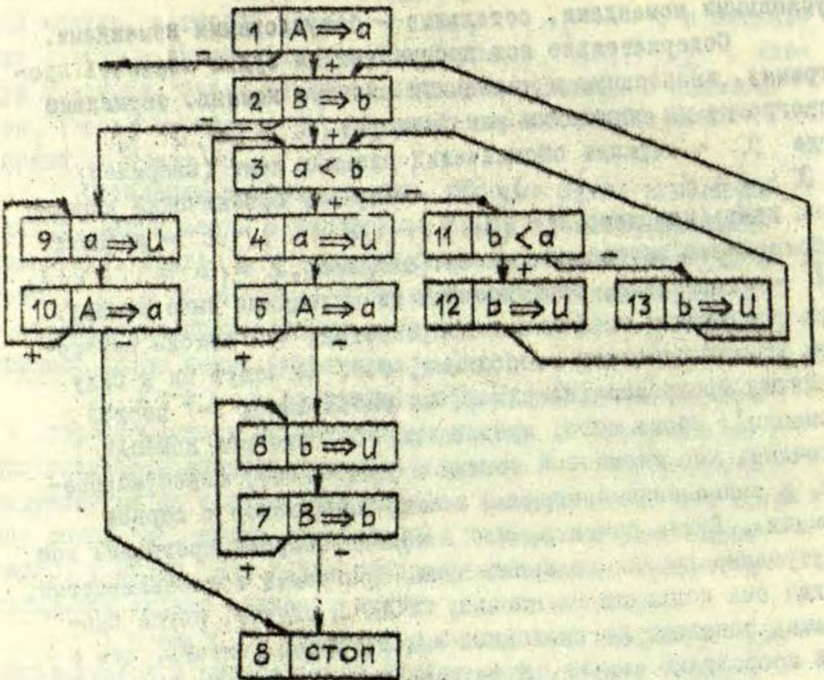


Рис.2.

и из команды a_l в команду a_{l+1} ведет выход $\varepsilon_l (\varepsilon \in \{+,-\})$
Для простоты будем считать, что из любой команды в любую другую команду ведет не более одного выхода. Поэтому в дальнейшем при обозначении пути будем писать просто $(a_1, a_2, \ldots, a_l)$. Путь $\delta = (a_1, a_2, \ldots, a_k)$ будем называть ветвью программы, если a_1 - либо первая команда программы, либо условная команда, a_k - либо условная команда, либо команда СТОП, а все a_l - безусловные команды. Таким образом ветви - это линейные куски программы. Программа, изображенная на рис.2, имеет следующие ветви: (1, 2), (2,3), (3,4,5), (2,9,10), (10,8), (11,13, 3) и т.д.

Под примером P для программы $T = \{\mathfrak{X}, \mathcal{Y}, \mathcal{Z}, \mathcal{T}\}$ будем понимать сопоставление, которое каждой входной ленте, т.е. каждой букве алфавита $\mathfrak{X}$ ставит в соответствие конкретный массив. Под применением программы T к примеру P будем понимать выполнение программы T при условии, что на входных лентах записаны сопоставленные им массивы. Будем говорить, что путь $\alpha = (a_1, a_2, \ldots, a_l)$ программы T реализуем на примере P, если программа T, примененная к P, проходит путь α, т.е. в своей работе, начиная с некоторого момента, выполняет последовательность команд $(a_1, a_2, \ldots, a_l)$. Пример P будем называть допустимым для программы T, если T, примененная к P, не имеет аварий и когда-нибудь останавливается. Далее систему примеров Σ для программы T будем называть полной, если:

1) Σ состоит из допустимых примеров для программы T;

2) любая ветвь программы T, которая реализуема на каком-нибудь допустимом примере, реализуема на некотором примере из Σ.

ТЕОРЕМА I. Существует алгоритм, который для любой программы T строит конечную полную систему примеров.

Для доказательства теоремы нам потребуется ряд вспомогательных утверждений. Пусть T - произвольная программа с алфавитом входных лент $\mathfrak{X} = \{A, B, \ldots, C\}$, с алфавитом выходных лент $\mathcal{Y} = \{U, V, \ldots, Z\}$, с алфавитом внутренних ячеек $\mathcal{Z} = \{a, b, \ldots, v\}$, и пусть $X \in \mathfrak{X}$, $t, v \in \mathcal{Z}$ и $Y \in \mathcal{Y}$. Введем обозначения: X^l - l-тая запись массива X, t^j - j-тое значение внутренней ячейки t, т.е. t^1 - первое значение, присваиваемое t командами вида $X \Rightarrow t$, $v \Rightarrow t$, $c \Rightarrow t$, t^2 - второе значение и т.д. ($t^0 = 0$), $\alpha = (a_1, a_2, \ldots, a_l)$ - произвольный путь программы T.

Введем граф реализуемости $G(\alpha)$ пути $\alpha = (a_1, a_2, \ldots, a_l)$ при условии, что a_1 - первая команда программы.

Вершины этого графа будут изображать записи массивов, введенных при выполнении пути α. К вершине, изображающей запись X^i, будем приписывать метку X^i, называемую основной. Вершины с основными метками вида X^i будем называть переменными вершинами. Этот граф будет содержать также и вершины, называемые константными, соответствующие константам, встречающимся в командах рассматриваемого пути (т.е. в командах вида $c \Rightarrow v, t < c$). Значения этих констант будут приписываться к соответствующим константным вершинам в качестве основных меток. Конец массива X будет изображаться специальной вершиной с основной меткой X^*. В дальнейшем вершину с основной меткой x будем называть просто вершиной x. Кроме этого, к вершинам будут приписываться вспомогательные метки вида t^j, содержательно означающие, что значение t^j равно значению, соответствующему основной метке, приписанной к той же вершине. Дуги между вершинами $G(\alpha)$ содержательно будут изображать те отношения между записями и константами, которые должны выполняться для того, чтобы α был реализуем.

ПРОЦЕДУРА ПОСТРОЕНИЯ $G(\alpha)$. $G(\alpha)$ строим индуктивно. В качестве базиса берем граф $G(\alpha_0)$, состоящий из одной константной вершины с основной меткой "0" и со вспомогательными метками $a^0, b^0, \ldots, v^0$ (содержательно этим изображается тот факт, что начальные значения всех внутренних ячеек равны нулю). Допустим, что уже изображена $p-1$ команда пути α и в результате этого получен граф $G(\alpha_{p-1})$. Пусть i, j, k – наибольшие индексы, встречающиеся в $G(\alpha_{p-1})$ соответственно в метках вида $X^{i'}, t^{j'}, v^{k'}$. Если метка вида $X^{i'}(t^{j'}, v^{k'})$ не использована в $G(\alpha_{p-1})$, то считаем $i=0 (j=0, v=0)$. Строим $G(\alpha_p)$:

1. Если команда a_p имеет вид $X \Rightarrow t$, и из a_p в a_{p+1} ведет выход "+", то к $G(\alpha_{p-1})$ добавляется вершина X^{i+1} со вспомогательной меткой t^{j+1}.

2. Если $X \Rightarrow t$ и выход "–", то добавляется спе-

циальная вершина X^* (если такая в $G(\alpha_{p-1})$ еще не имеется).

3. Если $t \Rightarrow Y$, то $G(\alpha_{p-1})$ не меняется.

4. Если $t \Rightarrow v$, то к вершине со вспомогательной меткой t^j приписывается еще одна вспомогательная метка v^{k+1}.

5. Если $c \Rightarrow v$, то строится константная вершина c (если такая в $G(\alpha_{p-1})$ еще не имеется) и к ней приписывается вспомогательная метка v^{k+1}.

6. Если $t<v\,(t<c)$ и выход "+", то из вершины с меткой v^k (из вершины c, которая добавляется к $G(\alpha_{p-1})$, если ранее ее не было) в вершину с меткой t^j проводится дуга, и к этой дуге приписывается метка "I", которую будем называть элементарным весом данной дуги. Если из вершины v^k в вершину t^j уже проведена дуга, то новая дуга не строится, но к уже существующей приписывается элементарный вес "I".

7. Если $t<v\,(t<c)$ и выход "-", то из вершины с меткой t^j в вершину с меткой v^k (в вершину c) проводится дуга с элементарным весом "0" (если такая еще не проведена). Если ранее уже была проведена дуга с элементарным весом "I", то новая дуга не строится.

8. Если НОП, СТОП, то $G(\alpha_{p-1})$ не меняется.

Определим теперь систему неравенств $N(\alpha)$, соответствующую пути α. По определению переменными системы $N(\alpha)$ являются все основные метки вида X^i, встречающиеся в $G(\alpha)$. (Подчеркнем, что если $G(\alpha)$ имеет метку A^i, то $G(\alpha)$ имеет и метку A^{i-1}). $N(\alpha)$ содержит столько отдельных неравенств, сколько дуг имеется в графе $G(\alpha)$. Пусть x и y - произвольные вершины $G(\alpha)$ (на самом деле это метки вида X^i или константы). Неравенство $x \geqslant y\,(x>y)$ принадлежит $N(\alpha)$ тогда и только тогда, когда в графе $G(\alpha)$ из вершины x в вершину y ведет дуга с элементарным весом 0 (соответственно I). Таким образом, решение системы $N(\alpha)$-это конкретные массивы

$A, B, \ldots, C$, т.е. пример для программы T .

На рис.3 изображен граф реализуемости $G(\alpha)$ при $\alpha = (1,2,3,4,5,3,11,12,2,3,4,5,6,7,8)$ для программы из рис.2.

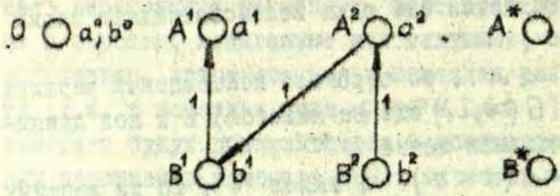


Рис.3.

Система неравенств $N(\alpha)$, соответствующая этому пути, имеет вид:

$$\begin{cases} A^1 < B^1 \\ A^2 > B^1 \\ A^2 < B^2 \end{cases}$$

Путь $\alpha = (a_1, a_2, \ldots, a_m, \ldots, a_n, \ldots a_l)$ будем называть противоречивым, если он содержит две команды вида $X \Rightarrow t$ и $X' \Rightarrow t'$, где $X = X'$ (пусть номера этих команд соответственно a_m и a_n, где $1 \leqslant m < n < l$) и из a_m в a_{m+1} ведет выход "−", а из a_n в a_{n+1} выход "+". Ясно, что противоречивый путь нереализуем, т.е. не существует пример, который его реализует.

ЛЕММА I. Пусть α - путь, начинающийся первой командой программы.

1. Путь α реализуем тогда и только тогда, когда он непротиворечив и $N(\alpha)$ имеет решение.

2. Если $A^1, A^2, \ldots, A^m, B^1, B^2, \ldots, B^n, \ldots\ldots, C^1, C^2, \ldots, C^2$ — решение системы $N(\alpha)$ и α - непротиворечивый путь, то пример $P = (A, B, \ldots, C)$, где

$$A = (A^1, A^2, \ldots, A^m)$$
$$B = (B^1, B^2, \ldots, B^n)$$
$$\ldots$$
$$C = (C^1, C^2, \ldots, C^z),$$

реализует путь α.

Справедливость этой леммы вытекает из построения $G(\alpha)$ и $N(\alpha)$.

Весом ориентированного пути графа $G(\alpha)$ будем называть сумму элементарных весов дуг, составляющих этот путь.

ЛЕММА 2. Система неравенств $N(\alpha)$ имеет решение тогда и только тогда, когда в $G(\alpha)$:

I. Не имеется циклических путей с весом больше нуля.

2. Вес любого пути, ведущего из любой константной вершины c_1 в любую другую константную вершину c_2, не больше разности $c_1 - c_2$.

Необходимость очевидна. Докажем достаточность. Решение системы будем строить следующим образом: исходя из $G(\alpha)$, мы будем постепенно приписывать переменным вершинам вида X^i некоторые значения. Эти значения будут служить решением системы $N(\alpha)$. Будем считать, что $G(\alpha)$ является связным графом; в противном случае описанную ниже процедуру будем проводить для каждой связной части $G(\alpha)$ отдельно. Будем говорить, что вершина x графа $G(\alpha)$ больше вершины y на $p (p > 0)$ единиц (или y меньше x на p единиц), если максимальный из весов путей, ведущих из x в y, равен p. Будем считать, что константным вершинам значения уже присвоены - это соответствующие константы. Если $G(\alpha)$ константных вершин не имеет, то выбираем какую-нибудь переменную вершину и присваиваем ей произвольное значение. Остальным вершинам значения будут приписываться индуктивно. Опишем шаг индукции. Выбираем произвольную вершину x, которая больше (или меньше) какой-нибудь вершины y, которой уже присвоено некоторое значение a. Из связности $G(\alpha)$ следует, что такая вершина обязательно найдется. Пусть вершина x больше (или меньше) вершины y на p единиц. Тогда вершине x мы присваиваем значение $a+p$ (соответственно $a-p$). Этот процесс продолжаем до момента, когда всем переменным вершинам будут присвоены значения. Из условий леммы следует, что этот процесс всегда выполним. В результате мы получим решение системы $N(\alpha)$.

Условия I и 2 леммы 2 в дальнейшем будем называть условиями реализуемости. Дальнейшее поведение программы после выполнения пути α, очевидно, зависит только от отношений между записями, содержащимися во внутренних ячейках. Для характеристики этой ситуации введем понятие состояния $S(\alpha)$ программы после выполнения пути α. $S(\alpha)$ определим, исходя из $G(\alpha)$, при помощи следующей процедуры:

I. Стираем в $G(\alpha)$ все основные метки вида X'.

2. Стираем все вспомогательные метки вершин, значения верхних индексов которых не являются максимальными; у вспомогательных меток, значения верхних индексов которых являются максимальными, стираем только эти индексы.

3. Исключаем поочередно вершины, которые после выполнения I и 2 не имеют меток, и между оставшимися вершинами проводим новые дуги, учитывая транзитивность отношений „$\geq$" и „$>$", т.е. если в исключаемую вершину входят дуги d_i с весами p_i и выходят дуги e_j с весами q_j, то любую пару дуг (d_i, e_j) заменяем на новую дугу f_k с весом $r_k = p_i + q_j$.

4. Заменяем веса дуг, которые превышают разность между максимальной и минимальной константами программы на эту разность (максимальной (соответственно, минимальной) константой программы называем наибольшую (соответственно, наименьшую) константу, входящую в команды вида $c \Rightarrow v$ и $t < c$; разность между максимальной и минимальной константами обозначаем через d). Если программа содержит

менее двух различных констант, то $d = 1$.

5. Стираем изолированные константные вершины, которые не имеют вспомогательных меток.

Таким образом, $S(\alpha)$ характеризует отношения между значениями внутренних ячеек после выполнения пути α. Подчеркнем, что признак конца массива (метка вида X^*) также остается в состоянии $S(\alpha)$.

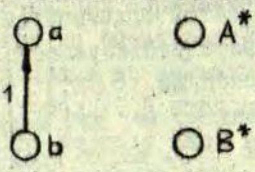


Рис.4.

На рис.4 представлено состояние программы из рис.2 после выполнения пути $\alpha = (1, 2,3,4,5,3,11,12,2,3,4,5,6,7, 8)$. (см.также рис.3).

Два состояния $S(\alpha)$ и $S(\beta)$ мы считаем одинаковыми $(S(\alpha) = S(\beta))$, если они как графы с помеченными вершинами и дугами совпадают. Очевидно, если $S(\alpha)=S(\beta)$, то $S(\alpha)$ и $S(\beta)$ одновременно удовлетворяют или не удовлетворяют условиям реализуемости. Отметим еще два свойства состояния $S(\alpha)$, вытекающие из его построения:

СВОЙСТВО А. Если в графе $G(\alpha)$ имеются вершины со вспомогательными метками t^j и v^k, где j, k - максимальные индексы в метках вида $t^{j'}, v^{k'}$, и, если из вершины с меткой t^j в вершину с меткой v^k ведет путь с весом p, где $p \leq d$, то в $S(\alpha)$ существуют вершины с метками t и v, и из вершины t в вершину v ведет также путь с весом p; если $p > d$, то в $S(\alpha)$ соответствующие вершины соединяет путь с весом d.

СВОЙСТВО В. Для любой программы число всевозможных различных состояний (когда мы берем всевозможные пути) конечно.

Путь $\alpha = (a_1, a_2, \dots, a_{n-1}, a_n, a_{n+1}, \dots, a_l)$, составленный из путей $\beta = (a_1, a_2, \dots, a_{n-1}, a_n)$ и $\gamma = (a_n, a_{n+1}, \dots, a_l)$, обозначим через $\alpha = \beta + \gamma$

ЛЕММА 3. (основная). Если пути α и β программы T, оканчивающиеся на одной и той же команде, реализуемы, и $S(\alpha)=S(\beta)$, то для любого пути γ из того, что реализуем путь $\alpha+\gamma$, следует, что реализуем путь $\beta+\gamma$.

Для доказательства леммы по аналогии с графом реализуемости $G(\gamma)$ введем граф реализуемости $G(S(\alpha),\gamma)$ пути γ из состояния $S(\alpha)$. Он получается применением описанной выше процедуры построения графа реализуемости, только в качестве исходного графа берется не $G(\alpha_0)$ (см. построение $G(\alpha)$), а граф $S(\alpha)$ (считается, что его меткам вида $t\in\mathcal{Z}$ приписаны верхние индексы "0"). Легко проверить, что для любого реализуемого пути α и любого пути γ графы $G(\alpha+\gamma)$ и $G(S(\alpha),\gamma)$ удовлетворяют условиям реализуемости одновременно. Для этого достаточно заметить, что $G(S(\alpha),\gamma)$ строится, исходя из $S(\alpha)$, а $G(\alpha+\gamma)$ – исходя из $G(\alpha)$. Но процедура построения графа реализуемости такова, что в каждый момент она использует только вершины с метками, имеющими наибольшие индексы. Согласно свойству А состояния $S(\alpha)$ отношения между этими вершинами в $S(\alpha)$ и в $G(\alpha)$ с точки зрения условий реализуемости одинаковы. Отсюда и вытекает требуемое соотношение между $G(S(\alpha),\gamma)$ и $G(\alpha+\gamma)$. Далее, если α и β оканчиваются на одной и той же команде и $S(\alpha)=S(\beta)$, то, очевидно, для любого γ $G(S(\alpha),\gamma)$ и $G(S(\beta),\gamma)$ удовлетворяют условиям реализуемости одновременно. Таким образом, мы получаем, что все четыре графа $G(\alpha+\gamma)$, $G(S(\alpha),\gamma)$ $G(S(\beta),\gamma)$, $G(\beta+\gamma)$ удовлетворяют условиям реализуемости одновременно. Отсюда и из реализуемости пути $\alpha+\gamma$ следует реализуемость пути $\beta+\gamma$. Лемма доказана.

Рассмотрим развертку программы T в виде дерева. К корню этого дерева (назовем его 0-м ярусом) припишем номер первой команды, т.е. единицу. Если q – произвольная вершина k-го яруса и к ней приписан номер команды $a(q)$, то из вершины q выходят столько же дуг и с теми же метками, что и из $a(q)$ -той команды в программе T . В концах этих дуг находятся вершины $k+1$ - го яруса, к которым приписаны номера соответствующих команд. Дуги, соответствующие свободным выходам команд, не строятся. Под ветвью дерева развертки программы будем понимать ветви, начинающиеся с корня дерева и идущие до обрыва (если ветвь когда-нибудь обрывается) или до бесконечности (если ветвь никогда не обрывается). Очевидно, ветвь обрывается, когда она доходит до команды СТОП или команды, все выходы которой свободны. В этом случае ветвь является конечной; последнюю вершину этой ветви будем называть концевой. Если концевой вершине приписан номер команды СТОП, то такую ветвь будем называть СТОП-ветвью, а остальные конечные ветви - тупиковыми ветвями. Ветвь будем задавать последовательностью вершин $(q_1, q_2, \dots, q_t, \dots)$, из которых она составлена.

Ясно, что между вершинами дерева развертки и путями программы, начинающимися первой командой, можно установить взаимно однозначное соответствие: вершине q , лежащей на ветви $(q_1, q_2, \dots, q, \dots)$, сопоставим путь $(a(q_1), a(q_2), \dots a(q))$. Этот путь обозначим через $\alpha(q)$. Далее, будем считать, что каждой вершине q дерева развертки кроме номера команды $a(q)$ приписано также состояние $S(\alpha(q))$, где $\alpha(q) = (a(q_1), a(q_2), \dots, a(q))$. Это состояние будем называть состоянием данной вершины.

В дальнейшем по определенному правилу будем обрезать ветви дерева. Мы начнем с ветви, определяемой меткой "+", т.е. с ветви, на которой из любой вершины в следующую вершину ведет выход "+". Пусть мы уже обрезали ветвь h в некоторой вершине q_{i_0} . (Более точно это означает, что в дереве развертки мы стерли дугу, исходящую из q_{i_0} и принадлежащую h , вместе со всеми ее продолжениями, принадлежащими как h , так и другим ветвям). Тогда ветвь h' , которую будем обрезать следующей, определим следующим образом: двигаясь по предыдущей ветви h обратно, начиная с вершины q_{i_0} в сторону начальной вершины отыскиваем вершину q', из которой выход "-" ведет в вершину q'', не принадлежащую ветви h. В таком случае начало ветви h' до вершины q' совпадает с началом ветви h, затем следует дуга (q', q''); а продолжение ветви h' опять определяется меткой "+".

Опишем теперь правило, по которому мы будем обрезать ветви дерева (ветви, оставшиеся после обрезания, будем называть обрезанными ветвями). Пусть $h=(q_1, q_2, \ldots, q_i, \ldots)$ - очередная ветвь. Просматривая эту ветвь, начиная с q_1, обрезаем ее на первой вершине q_{i_0}, на которой выполняется одно из следующих условий:

1. q_{i_0} является концевой вершиной ветви (тривиальное обрезание).

2. Если путь $\alpha(q_{i_0})=(a(q_1), a(q_2), \ldots, a(q_{i_0}))$ реализуем, но путь $\alpha(q_{i_0+1})=(a(q_1), a(q_2), \cdots, a(q_{i_0}), a(q_{i_0+1}))$, который получается добавлением к $\alpha(q_{i_0})$ дуги (q_{i_0}, q_{i_0+1}) - нереализуем.

3. Просматриваем все обрезанные ветви в порядке их обрезания и все их вершины в порядке их следования на ветвях, включая и вершины данной ветви, лежащие ниже q_{i_0}. Если среди этих вершин существует вершина p_j такая, что $a(p_j)=a(q_{i_0})$ и $S(\alpha(p_j))=S(\alpha(q_{i_0}))$, то ветвь h обрезаем в вершине q_{i_0} и полученную обрезанную ветвь называем ветвью, обрезанной по повторению. Первую такую вершину p_j при описанном выше порядке просмотра дерева будем называть обрезающей вершиной. Легко проверить, что обрезающая вершина не может быть концевой вершиной ни на одной ветви. Если обрезающая вершина находится на обрезаемой ветви, то эту ветвь будем называть обрезанной по самоповторению.

Из того, что число различных состояний для любой программы конечно, следует, что все ветви будут обрезаны и процесс закончится построением конечного дерева, состоящего из обрезанных ветвей. Легко убедиться, что описанный процесс обрезания является эффективным. В случаях I и 3 это очевидно, эффективность случая 2 вытекает из лемм I и 2. Учитывая еще, что дерево развертки достаточно строить только до точек обрезания, мы получаем, что существует алгоритм, который по любой программе T строит описанное выше обрезанное дерево развертки.

Возьмем теперь обрезанное дерево и будем приписывать к его ветвям, обрезанным по повторению (в том числе и по самоповторению), продолжения, заканчивающиеся командой СТОП. А именно, пусть для некоторой обрезанной ветви $h=(q_1, q_2, \dots, q_{i_0})$ обрезающая ее вершина p_j лежит на какой-нибудь СТОП ветви $(p_1, p_2, \dots, p_{j-1}, p_j, p_{j+1}, \dots, p_r)$ ($a(p_r)$ - номер команды СТОП). Тогда к ветви h приписываем отрезок СТОП-ветви $(p_j, p_{j+1}, \dots, p_r)$ и вновь полученную ветвь $(q_1, q_2, \dots q_{i_0}, p_{j+1}, \dots, p_r)$ называем СТОП-ветвью. Получаем новое дерево. К этому дереву опять применяем описанную процедуру. Такой процесс продолжаем до тех пор, пока количество СТОП-ветвей больше не увеличивается. Полученное в итоге дерево будем называть деревом реализуемости. Дерево реализуемости программы из рис.2 имеет вид, изображенный на рис.5. В рис.5 в концах обрезанных ветвей приписано: СП - для ветвей обрезанных по самоповторению, СТОП - для СТОП-ветвей, НР - для нереализуемых ветвей. Прерывистыми линиями отмечены продолжения СТОП-ветвей, приписанные к обрезанным ветвям. В концах ветвей указаны номера ветвей.

Заметим, что каждая ветвь $h=(q_1, q_2, \dots, q_r)$ дерева представляет определенный путь в программе, а именно: $(a(q_1), a(q_2), \dots, a(q_r))$. Из построения дерева реализуемости и леммы 3 непосредственно вытекает

ЛЕММА 4. Для любой ветви дерева реализуемости существует пример P , который реализует эту ветвь (т. е. реализует путь программы, задаваемый этой ветвью).

Если путь программы, задаваемый некоторой ветвью h

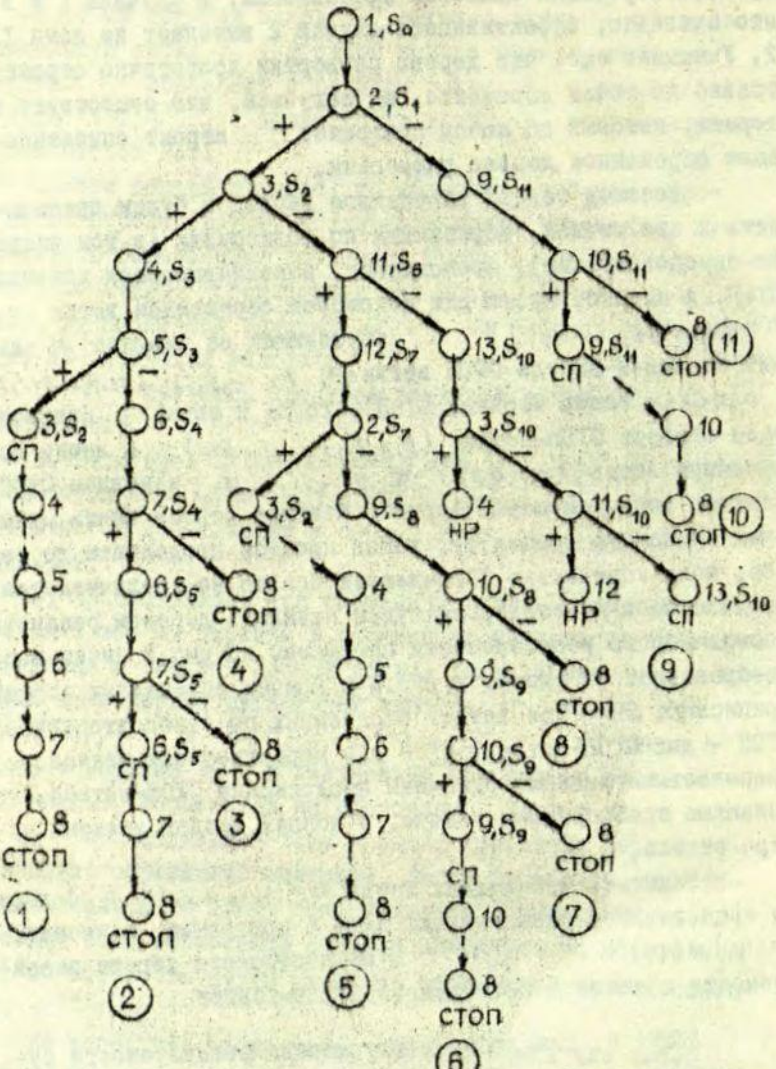

1, S_0
2, S_1
3, S_2
9, S_{11}
4, S_3
11, S_6
10, S_{11}
5, S_3
12, S_7
13, S_{10}
9, S_{11}
СП
8
СТОП
3, S_2
СП
6, S_4
2, S_7
3, S_{10}
10
4
7, S_4
3, S_2
СП
9, S_8
4
НР
11, S_{10}
8
СТОП
5
6, S_5
8
СТОП
4
10, S_8
12
НР
13, S_{10}
СП
6
7, S_5
5
9, S_9
8
СТОП
7
6, S_5
СП
8
СТОП
6
10, S_9
8
СТОП
7
7
9, S_9
СП
8
СТОП
8
СТОП
8
СТОП
10
8
СТОП
1
2
3
4
5
6
7
8
9
10
11
+
−

Рис. 5

дерева реализуемости, содержит ветвь δ программы, то будем говорить, что ветвь h дерева реализуемости содержит ветвь δ программы.

ЛЕММА 5. Ветвь программы $\delta=(a_1,a_2,\dots,a_k)$ реализуема на каком-нибудь допустимом примере тогда и только тогда, когда существует СТОП-ветвь дерева реализуемости, которая содержит ветвь δ.

Достаточность условий леммы вытекает из леммы 4. Докажем необходимость. Будем говорить, что первая дуга ветви $(a_0,a_1,\dots,a_k)$ программы принадлежит ветви $(q_1, q_2,\dots,q_r,\dots,q_s)$ дерева, если существует такое r, что $a(q_r)=a_0, a(q_{r+1})=a_1$. Из построения дерева реализуемости и определения ветви программы (как линейного куска программы) вытекает следующее свойство: если СТОП-ветвь дерева реализуемости содержит первую дугу какой-нибудь ветви программы, то эта ветвь дерева содержит целиком всю ветвь программы. Поэтому для доказательства того, что произвольная ветвь $\delta=(a_1,a_2,\dots,a_k)$ программы принадлежит какой-нибудь СТОП-ветви дерева реализуемости, нам достаточно доказать, что первая дуга этой ветви принадлежит некоторой СТОП-ветви дерева реализуемости. Фактически мы будем доказывать более сильное утверждение, но перед этим введем несколько понятий. Будем говорить, что дуга $\alpha=(a,b)$ программы T реализуема из состояния S, если для T существует пример, который реализует $\alpha=(a,b)$, и перед выполнением команды a программа находится в состоянии S. Будем говорить, что дуга $\alpha=(a,b)$ программы, начинающаяся состоянием S, принадлежит дереву реализуемости, если в дереве существует ветвь $h=(q_1,q_2,\dots,q_s)$ с вершинами q_r и q_{r+1} такими, что $a(q_r)=a, a(q_{r+1})=b$ и $S(q_r)=S$. Очевидно, лемма 5 будет доказана, если мы докажем следующее

УТВЕРЖДЕНИЕ А. Пусть $\alpha=(a_1,a_2,\dots,a_l,\dots)$- произвольный реализуемый путь, где a_1 - первая команда программы и пусть $S_1, S_2, \dots, S_l, \dots$ - состояния, в которых находится программа перед выполнением соответственно команд $a_1, a_2, \dots, a_l, \dots$. Тогда любая дуга этого пути, начинающаяся с любого состояния, из которого она реализуема, принадлежит дереву реализуемости. Притом, если $\alpha = (a_1, a_2, \dots, a_l)$ реализуем и a_l - команда СТОП, то любая дуга этого пути, начинающаяся с состояния, из которого она реализуема, принадлежит некоторой СТОП-ветви дерева реализуемости.

Утверждение будем доказывать индукцией по l, $l = 1, 2, \dots, l-1, \dots$. Из построения дерева реализуемости вытекает базис индукции: дуга (a_1, a_2), начинающаяся с начального состояния программы, принадлежит некоторой ветви $h_0 = (q_1^0, q_2^0, \dots, q_\tau^0)$ дерева реализуемости, где $a(q_1^0) = a_1$, $a(q_2^0) = a_2$ и $S(q_1^0)$ - начальное состояние программы. Пусть мы уже доказали, что дуга (a_{l-1}, a_l), начинающаяся из состояния S_{l-1}, из которого она реализуема, принадлежит некоторой ветви $h_{l-1} = (q_1, q_2, \dots, q_\tau, \dots)$ дерева реализуемости, где $a(q_{\tau-1}) = a_{l-1}$, $a(q_\tau) = a_l$ и $S(q_{\tau-1}) = S_{l-1}$. Докажем, что существует ветвь $h_l = (q_1', q_2', \dots, q_p', q_{p+1}', \dots, q_s')$ дерева реализуемости, где $a(q_p') = a_l$, $a(q_{p+1}') = a_{l+1}$ и $S(q_p') = S_l$. Сначала отметим, что если ветвь h_{l-1} в вершине q_τ не была обрезана, то $h_l = h_{l-1}$ и требуемое утверждение доказано. Поэтому рассмотрим случай, когда ветвь h_{l-1} обрезана в вершине q_τ. Пусть q_p' - обрезающая вершина $(S(q_\tau) = S(q_p'))$ и пусть она принадлежит ветви $h_l = (q_1', q_2', \dots, q_p', \dots)$. Ясно, что в дереве развертки из q_p' выходит продолжение ветви $(q_p', q_{p+1}', \dots, q_s', \dots)$ такое, что $a(q_p') = a_l$, $a(q_{p+1}') = a_{l+1}, \dots, a(q_s') = a_l, \dots$. Согласно лемме 3, путь $(a(q_1'), a(q_2'), \dots, a(q_p'), a(q_{p+1}'), \dots)$ реализуем. Ввиду того, что q_p' - обрезающая вершина для ветви h_{l-1}, ветвь h_l не может быть обрезана в вершине q_p', и, следовательно, q_{p+1}' принадлежит дереву реализуемости.

То, что к вершине q'_p приписано состояние S_l, из которого реализуема дуга (a_l, a_{l+1}), тривиально следует из построения дерева реализуемости. Первая половина утверждения А доказана. Остается убедиться, что если a_l - команда СТОП, то в описанной выше конструкции любая дуга пути принадлежит именно некоторой СТОП-ветви. Для этого рассмотрим описанный индуктивный процесс в обратном направлении. Очевидно, что ветвь h_{l-1}, которой принадлежит последняя дуга (a_{l-1}, a_l), является СТОП-ветвью. Вспоминая процесс дополнения обрезанных ветвей СТОП-ветвями при построении дерева реализуемости, мы получаем, что ветвь h_{l-2} также является СТОП-ветвью. Продолжая индуктивно это рассуждение, мы получаем, что СТОП-ветвями являются и остальные ветви $h_{l-3}, h_{l-4}, \ldots, h_l, \ldots, h_0$. Утверждение А и тем самым лемма 5 доказаны.

Доказательство теоремы I закончим кратким изложением алгоритма построения полной системы примеров:

I. Для данной программы T строим дерево реализуемости.

2. Составляем системы неравенств $N(\alpha_1), N(\alpha_2), \ldots, N(\alpha_s)$ путей программы, задаваемых всем СТОП-ветвям $h_1, h_2, \ldots, h_s$ дерева реализуемости.

3. Решаем все системы неравенств $N(\alpha_1), N(\alpha_2), \ldots, N(\alpha_s)$ ($N(\alpha_i)$, $i = 1, 2, \ldots, s$ имеет решение в силу лемм I и 4). Для каждой системы неравенств $N(\alpha_i)$ выбираем одно решение. Это решение задает пример P_i, реализующий ветвь h_i. Из леммы 5 вытекает, что система примеров $\Sigma = (P_1, P_2, \ldots, P_s)$ является искомой. Теорема доказана.

Полная система примеров для программы из рис.2, построенная описанным методом, имеет следующий вид:

Пример	Система неравенств	Входные массивы		Результат
P_i	$N(\alpha_i)$	A	B	И
P_1	$A^1 < B^1,\ A^2 < B^1$	0,I	2	0,I,2
P_2	$A^1 < B^1,\ B^2, B^3$	0	I,2,3	0,I,2,3
P_5	$A^1 > B^1,\ A^1 < B^2$	I	0,2	0,I,2
P_6	$A^1 > B^1,\ A^2, A^3$	I,2,3	0	0,I,2,3
P_{10}	$A^1,\ A^2,\ A^3$	0,I,2	-	0,I,2

В этой таблице примеры $P_1, P_2, P_5, P_6, P_{10}$ построены соответственно по СТОП-ветвям I,2,5,6,I0 (см.рис.5). Эти СТОП-ветви уже содержат все ветви программы, поэтому другие СТОП-ветви дерева реализуемости не рассматривались.

3°. Построенная выше полная система примеров предназначена для проверки работы программы в "нормальных" ситуациях, когда программа на любом примере когда-нибудь останавливается. Однако возникает вопрос о выявлении "ненормальных" ситуаций в программах, т.е. ситуаций, когда существует пример, на котором программа вообще не останавливается. В последнем случае будем говорить, что программа зацикливается.

ТЕОРЕМА 2. Существует алгоритм, который для любой программы T выясняет, зацикливается ли эта программа.

Сначала докажем несколько лемм.

ЛЕММА 6. Пусть $\alpha = (a_1, a_2, \ldots, a_r, \ldots, a_{r+s}, \ldots)$ - путь, реализуемый на примере P, и пусть $S_1, S_2, \ldots, S_r, \ldots, S_{r+s}, \ldots$ - состояния программы перед выполнением соответственно команд $a_1, a_2, \ldots, a_r, \ldots, a_{r+s}, \ldots$. Пусть $a_r = a_{r+s}$, $S_r = S_{r+s}$ и в пути $(a_r, a_{r+1}, \ldots, a_{r+s})$ нет команды вида $X \Rightarrow t$ с переходом по выходу "+".

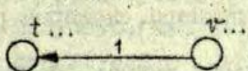
Тогда пример P зацикливает программу на последовательности $((S_t, a_t), (S_{t+1}, a_{t+1}), \dots, (S_{t+s}, a_{t+s}))$, т.е. $S_t = S_{t+ls}$, $a_t = a_{t+ls}$, $S_{t+1} = S_{t+ls+1}$, $a_{t+1} = a_{t+ls+1}$, ..., $S_{t+s-1} = S_{t+ls+s-1}$, $a_{t+s-1} = a_{t+ls+s-1}$, $l = 1,2,\dots$.

Справедливость леммы вытекает из следующих двух утверждений:

(I) Пусть $a_t = a_{t+s}$ - произвольная команда пути α и пусть a_t на рассматриваемом примере осуществляет переход по выходу ε $(\varepsilon \in \{+, -\})$; в таком случае, если выполняются условия леммы, то a_{t+s} также осуществляет переход по выходу ε. (Следовательно, $a_{t+1} = a_{t+s+1}$)

(2) $S_{t+1} = S_{t+s+1}$.

Так как $S_t = S_{t+s}$ и следующее состояние однозначно определяется предыдущим состоянием, текущей командой и ее выходом, то справедливость (2) вытекает из справедливости (I). Остается убедиться в справедливости (I). Утверждение (I) нетривиально, если команда a_t имеет вид $t < v$. Для определенности предположим, что в момент выполнения команды a_t условие $t < v$ выполняется, т.е. переход из a_t в a_{t+1} осуществляется по выходу "+". Это означает, что после выполнения команды a_t в $G((a_1, a_2, \dots, a_t))$ и, следовательно, в состоянии S_{t+1} должна быть дуга вида



(*)

Ясно, что если уже в состоянии $S_t = S_{t+s}$ имеется дуга (*), то команда a_t и, следовательно, команда a_{t+s} может осуществлять переход только по выходу "+". Таким образом, доказательство утверждения (I) сводится к доказательству того, что уже в состоянии S_t имеется дуга (*). Из того, что $S_t = S_{t+s}$ и среди команд $a_t, a_{t+1}, \dots, a_{t+s}$ нет команды вида $X \Rightarrow t$ с переходом по выходу "+", следует, что при построении графа реализуемости $G(\alpha)$ в промежутке $(a_r, a_{r+1}, \dots, a_{r+s})$ новые вершины и дуги не добавляются. Но так как после выполнения команды a_r в $G(\alpha)$ и, следовательно, в $S_{r+s} = S_r$ обязательно должна быть дуга вида $(*)$, то эта дуга уже должна быть в состоянии $S_r = S_{r+s}$.

Случай, когда команда a_r имеет выход "-", анализируется аналогично. Лемма 6 доказана.

Будем говорить, что программа зацикливается в множестве команд $E = \{a_0, a_1, \dots, a_s\}$, если существует пример, на котором программа, начиная с некоторого момента, выполняет только команды из E, притом каждую из них неограниченно много раз. Ветвь дерева реализуемости $h = (q_1, q_2, \dots, q_r, \dots, q_{r+s})$ будем называть циклической, если: I) h обрезана по самоповторению (пусть обрезающей является вершина q_r), 2) от команды $a(q_r)$ по $a(q_{r+s})$ нет команды вида $X \Rightarrow t$ с переходом по "+". В данном случае множество команд $E = \{a(q_r), a(q_{r+1}), \dots, a(q_{r+s-1}), a(q_{r+s})\}$, приписанных от обрезающей по концевую вершину ветви h, будем называть циклическим для циклической ветви h.

ЛЕММА 7. Пусть h - произвольная циклическая ветвь дерева реализуемости и пусть E - циклическое множество этой ветви. Тогда программа зацикливается в множестве E на любом примере, который реализует h.

Пусть P - произвольный пример, который реализует циклическую ветвь h. Очевидно, для этого примера выполняются условия леммы 6. Отсюда вытекает справедливость леммы 7.

ЛЕММА 8. Пусть E - множество, в котором зацикливается программа T. Тогда в дереве реализуемости существует циклическая ветвь с циклическим множеством E.

Пусть программа T зацикливается в E на некотором примере P. Это означает, что примеру P соответствует бесконечный путь $\alpha = (a_1, a_2, \ldots, a_{t+s}, \ldots)$; пусть $S_1, S_2, \ldots, S_t, \ldots, S_t$ - состояния, в которых находится программа перед выполнением соответственно команд $a_1, a_2, \ldots, a_t, \ldots, a_{t+s}, \ldots$. Рассмотрим последовательность $\bar{\alpha} = ((S_1, a_1), (S_2, a_2), \ldots, (S_t, a_t), \ldots$ $\ldots, (S_{t+s}, a_{t+s}), \ldots)$. Из леммы 6 вытекает, что начиная с некоторого места, эта последовательность станет периодической, т.е. существуют такие t и s, что программа зацикливается на последовательности $F = ((S_t, a_t), (S_{t+1}, a_{t+1}), \ldots$ $\ldots, (S_{t+s}, a_{t+s}))$. Будем считать, что этот период является наименьшим, т.е. все пары из F различны. Ищем в дереве реализуемости первую вершину q_p (в порядке построения этого дерева) такую, что пара $(S(q_p), a(q_p)) \in F$. Для простоты будем считать, что $(S(q_p), a(q_p)) = (S_t, a_t)$ и q_p принадлежит ветви $h = (q_1, q_2, \ldots, q_p, \ldots)$. В силу утверждения А из доказательства леммы 5 такая вершина в дереве обязательно найдется. Теперь нетрудно убедиться, что из q_p выходит продолжение $(q_{p+1}, q_{p+2}, \ldots, q_{p+s})$, где $(S(q_{p+1}), a(q_{p+1})) = (S_{t+1}, a_{t+1}), (S(q_{p+2}), a(q_{p+2})) =$ $= (S_{t+2}, a_{t+2}), \ldots, (S(q_{p+s}), a(q_{p+s})) =$ $= (S_{t+s}, a_{t+s})$ и это продолжение целиком принадлежит дереву реализуемости. Действительно, так как путь α реализуем и $(S(q_p), a(q_p)) = (S_t, a_t)$, то по лемме 3 получаем, что ветвь $h = (q_1, q_2, \ldots, q_p, \ldots$ $\ldots, q_{p+s})$ реализуема и $S(q_p) = S_t, S(q_{p+1}) = S_{t+1}, \ldots$ $\ldots, S(q_{p+s}) = S_{t+s}$. Отсюда следует, что ветвь h не может быть обрезана ни в одной из вершин $q_p, q_{p+1}, \ldots, q_{p+s-1}$, так как это противоречило бы выбору вершины q_p и тому, что все пары $(S_t, a_t), (S_{t+1}, a_{t+1}), \ldots, (S_{t+s}, a_{t+s})$ различны. Эта ветвь будет обрезана только в вершине q_{p+s} по самоповторению с вершиной q_p. Согласно лемме 6, к ветви h не может быть приписано продолжение до команды СТОП, поэтому h не является СТОП-ветвью. Таким образом, для полного доказательства леммы остается убедиться, что путь $(a_i, a_{i+1}, \cdots, a_{i+s}) = (a(q_p), a(q_{p+1}), \cdots, a(q_{p+s}))$ не содержит команд вида $X \Rightarrow t$ с переходом по "+". Это вытекает из того, что пример P конечный, но путь $(a_i, a_{i+1}, \cdots, a_{i+s})$ выполняется неограниченно много раз и поэтому не может содержать команду вида $X \Rightarrow t$ с переходом по "+". Лемма доказана.

Из лемм 7 и 8 непосредственно вытекает существование алгоритма, требуемого в теореме 2: если в дереве реализуемости данной программы T существуют циклические ветви, то программа может зацикливаться; в противном случае она не может зацикливаться. Теорема доказана.

Далее возникает вопрос об алгоритмическом выявлении всех ситуаций, в которых программа зацикливается. Систему примеров Σ будем называть циклически полной для программы T, если в любом множестве команд E, в котором программа зацикливается на каком-нибудь примере, программа зацикливается на некотором примере из Σ.

ТЕОРЕМА 3. Существует алгоритм, который для любой программы T строит конечную циклически полную систему примеров.

Алгоритм построения искомой циклически полной системы примеров аналогичен алгоритму построения полной системы примеров в теореме I, только вместо СТОП-ветвей дерева реализуемости мы рассматриваем циклические ветви этого дерева и вместо леммы 5 используем леммы 7 и 8.

Циклически полная система примеров для программы из рис.2 состоит из одного примера $\Sigma = (P_9)$ (см.дерево реализуемости на рис.5, ветвь 9), где $P_9 = (A=0, B'=0)$

4°. Базисный язык программирования, описанный в I с точки зрения практического применения, является весьма ограниченным. Но он обладает тем свойством, что с точки зрения построения полной системы примеров, к нему сводим практически весьма широкий класс программ. Для иллюстрации этой ситуации мы построим полную систему примеров для одной программы, записанной на КОБОЛе. В данной статье мы не будем четко формулировать ограничения на программы КОБОЛа (хотя это не представляет трудностей), при которых описанный выше алгоритм применим. Выбранный нами пример КОБОЛ-программы заимствован из книги [6]. Он является достаточно типичным примером для задачи обработки данных.

ПРИМЕР. Рассматривается несколько упрощенная задача для ежемесячного расчета суммы заработка, выдаваемой на руки, и обновления массивов облагаемых налогом сумм. Входными массивами при расчете являются:

а) массив начислений, содержащий табельные номера и начисленные в текущем месяце суммы заработка работающих;

б) массив облагаемых налогом сумм, содержащий табельные номера и начисленные в прошедшем месяце суммы заработка работающих;

в) массив фамилий, содержащий фамилии, табельные номера и шифры налогоплательщиков.

Выходные массивы расчета:

а) обновленный массив облагаемых налогом сумм;

б) ведомость на выплату зарплаты, содержащая табельные номера, фамилии, суммы, начисленные в текущем месяце, удержки за прошедший месяц и суммы, выдаваемые на руки.

Величина удержаний рассчитывается, исходя из суммы облагаемой налогом и в зависимости от шифра налогоплательщика (более подробно см. в программе).

ПРОГРАММА

```
РАЗДЕЛ ИДЕНТИФИКАЦИИ.
ПРОГРАММА. РАСЧЕТ УДЕРЖАНИЙ И ОБНОВЛЕНИЕ МАССИВА ОБЛАГАЕ-
           МЫХ НАЛОГОМ СУММ.
АВТОР. ИВАНОВ ИВАН ИВАНОВИЧ.
ДАТА-НАПИСАНИЯ. ЯНВАРЬ 1972 ГОДА.
```

```
РАЗДЕЛ ОБОРУДОВАНИЯ.
 СЕКЦИЯ КОНФИГУРАЦИИ.
  ИСХОДНАЯ-МАШИНА. МИНСК-32.
  РАБОЧАЯ-МАШИНА. МИНСК-32.
  СПЕЦИАЛЬНЫЕ-НАЗВАНИЯ.
      ПЧ I ЕСТЬ ЫШ.
  СЕКЦИЯ ВВОДА-ВЫВОДА.
   УПРАВЛЕНИЕ-МАССИВАМИ.
      ДЛЯ МАССИВ-НАЧИСЛЕНИЙ ПРЕДНАЗНАЧИТЬ ВК I.
      ДЛЯ МАССИВ-ОБЛАГАЕМЫХ-СУММ ПРЕДНАЗНАЧИТЬ МЛ I.
      ДЛЯ МАССИВ-ФАМИЛИЙ ПРЕДНАЗНАЧИТЬ МЛ 2.
      ДЛЯ ОБНОВ-МАССИВ ПРЕДНАЗНАЧИТЬ МЛ 3.
      ДЛЯ ВЕДОМОСТЬ ПРЕДНАЗНАЧИТЬ ПЧ I.
РАЗДЕЛ ДАННЫХ.
 СЕКЦИЯ МАССИВОВ.
  ОМ МАССИВ-НАЧИСЛЕНИЙ; МЕТКИ ОПУЩЕНЫ; В БЛОКЕ 50 ЗАПИСЕЙ.
   I ЗАПИСЬ-Н.
    2 ТАБ-НОМ-Н; ШАБЛОН 9(4).
    2 НАЧИСЛЕНО; ШАБЛОН 9(4) Т99.
ОМ МАССИВ-ОБЛАГАЕМЫХ-СУММ;
   МЕТКИ СТАНДАРТНЫ; ШИФР МАССИВА 'ОБЛАГ';
   В БЛОКЕ 50 ЗАПИСЕЙ.
 I ЗАПИСЬ-С.
  2 ТАБ-НОМ-С; ШАБЛОН 9(4).
  2 ОБЛАГ-СУММА; ШАБЛОН 9(4) Т99.
ОМ МАССИВ-ФАМИЛИЙ; МЕТКИ ОПУЩЕНЫ; В БЛОКЕ 50 ЗАПИСЕЙ.
 I ЗАПИСЬ-Ф.
  2 ФАМИЛИЯ; ШАБЛОН X(I8).
  2 ШИФР-НАЛОГА; ШАБЛОН 9.
  2 ТАБ-НОМ-Ф; ШАБЛОН 9(4).
 ОМ ОБНОВ-МАССИВ: МЕТКИ СТАНДАРТНЫ; ШИФР МАССИВА 'ОБЛАГ';
   ПЕРИОД ГОДНОСТИ 3I; В БЛОКЕ 50 ЗАПИСЕЙ.
 I ЗАПИСЬ-ОБН.
  2 ТАБ-НОМ-ОБН; ШАБЛОН 9(4).
  2 ОБЛАГ-СУММА-ОБН; ШАБЛОН 9(4) Т99.
```

```
ОМ ВЕДОМОСТЬ; МЕТКИ ОПУЩЕНЫ.
 I СТРОКА-ВЕДОМОСТИ.
  2 ЗАПОЛНИТЕЛЬ; ШАБЛОН X(4); ЗНАЧЕНИЕ ПРОБЕЛ.
  2 ТАБ-НОМЕР; ШАБЛОН 9(4).
  2 ЗАПОЛНИТЕЛЬ; ШАБЛОН X(IO); ЗНАЧЕНИЕ ПРОБЕЛ.
  2 Ф-И-О; ШАБЛОН X(I8).
  2 ЗАПОЛНИТЕЛЬ; ШАБЛОН X(II); ЗНАЧЕНИЕ ПРОБЕЛ.
  2 НАЧИСЛ-РЕД; ШАБЛОН ППП9.99.
  2 ЗАПОЛНИТЕЛЬ; ШАБЛОН X(IO); ЗНАЧЕНИЕ ПРОБЕЛ.
  2 УДЕРЖ-РЕД; ШАБЛОН ПП9.99.
  2 ЗАПОЛНИТЕЛЬ; ШАБЛОН X(IO); ЗНАЧЕНИЕ ПРОБЕЛ.
  2 НА-РУКИ; ШАБЛОН П(3)9.99.
СЕКЦИЯ РАБОЧЕЙ ПАМЯТИ.
  77 УДЕРЖАНО; ШАБЛОН 999Т99.
  I ЗАГОЛОВОК.
   2 ПУСТО; ШАБЛОН X(40); ЗНАЧЕНИЕ ПРОБЕЛ.
   2 НАЗВАНИЕ; ШАБЛОН X(29); ЗНАЧЕНИЕ 'ВЕДОМОСТЬ НА ВЫПЛА-
     ТУ ЗАРПЛАТЫ'.
РАЗДЕЛ ПРОЦЕДУР.
НАЧАЛО. ОТКРЫТЬ ВХОДНОЙ МАССИВ-НАЧИСЛЕНИЙ, МАССИВ-ОБЛАГА-
   ЕМЫХ-СУММ, МАССИВ-ФАМИЛИЙ, ВЫХОДНОЙ ОБНОВ-МАССИВ, ВЕДО-
   МОСТЬ.
   ВЫДАТЬ ЗАГОЛОВОК НА ЫШ.
ЧТЕНИЕ.
   ЧИТАТЬ МАССИВ-НАЧИСЛЕНИЙ В КОНЦЕ ПЕРЕЙТИ К КОНЕЦ.
ЧТФАМ.
   ЧИТАТЬ МАССИВ-ФАМИЛИЙ В КОНЦЕ ПЕРЕЙТИ К КОНЕЦ.
    ЕСЛИ ТАБ-НОМ-Н=ТАБ-НОМ-Ф ПЕРЕЙТИ К ЧТОБЛ.
    ЕСЛИ ТАБ-НОМ-Н>ТАБ-НОМ-Ф ПЕРЕЙТИ К ЧТФАМ;
      ИНАЧЕ ВЫДАТЬ 'ФАМИЛИИ С ТАБЕЛЬНЫМ НОМЕРОМ', ТАБ-НОМ-
      Н, 'НЕТ' НА ЫШ; ПЕРЕЙТИ К ЧТЕНИЕ.
ЧТОБЛ.
    ЧИТАТЬ МАССИВ-ОБЛАГАЕМЫХ-СУММ В КОНЦЕ ПЕРЕЙТИ К ПI.
    ЕСЛИ ТАБ-НОМ-Н=ТАБ-НОМ-С ПЕРЕЙТИ К РАСЧЕТ.
    ЕСЛИ ТАБ-НОМ-Н > ТАБ-НОМ-С ПЕРЕЙТИ К ЧТОБЛ.
```

```
II.
   ПОМЕСТИТЬ 0 В УДЕРЖАНО; ПЕРЕЙТИ К ПЕЧАТЬ-СТРОКИ.
РАСЧЕТ. ЕСЛИ ОБЛАГ-СУММА < 81; ПОМЕСТИТЬ 0 В УДЕРЖАНО; ПЕ-
   РЕЙТИ К ПЕРЕКЛЮЧ.
   ЕСЛИ ОБЛАГ-СУММА МЕНЬШЕ 101 ПЕРЕЙТИ К ФОРМ1.
ФОРМ2.
   ВЫЧИСЛИТЬ УДЕРЖАНО=8.20+(ОБЛАГ-СУММА - 100)* 0.13;
   ПЕРЕЙТИ К ПЕРЕКЛЮЧ.
ФОРМ1.
   ВЫЧИСЛИТЬ УДЕРЖАНО=5.80+(ОБЛАГ-СУММА - 80)* 0.12.
ПЕРЕКЛЮЧ.
   ПЕРЕЙТИ К ПОД-ХОЛ, ПЕЧАТЬ-СТРОКИ, НАЛОГ-СО-СКИДКОЙ В
   ЗАВИСИМОСТИ ОТ ШИФР-НАЛОГА.
ПОД-ХОЛ.
   ВЫЧИСЛИТЬ УДЕРЖАНО=УДЕРЖАНО+0.06 * ОБЛАГ-СУММА.
   ПЕРЕЙТИ К ПЕЧАТЬ-СТРОКИ.
НАЛОГ-СО-СКИДКОЙ.
   ВЫЧИСЛИТЬ УДЕРЖАНО=УДЕРЖАНО * 0.7.
ПЕЧАТЬ-СТРОКИ.
   ВЫЧИСЛИТЬ НА-РУКИ=НАЧИСЛЕНО-УДЕРЖАНО;
   ПОМЕСТИТЬ УДЕРЖАНО В УДЕРЖ-РЕД;
   ПОМЕСТИТЬ НАЧИСЛЕНО В НАЧИСЛ-РЕД;
   ПОМЕСТИТЬ ФАМИЛИЯ В Ф-И-О; ПОМЕСТИТЬ ТАБ-НОМ-Н В ТАБ-
   НОМЕР.
   ПИСАТЬ СТРОКА ВЕДОМОСТИ.
ОБНОВЛЕНИЕ.
   ПОМЕСТИТЬ НАЧИСЛЕНО В ОБЛАГ-СУММА-ОБН.
   ПОМЕСТИТЬ ТАБ-НОМ В ТАБ-НОМ-ОБН;
   ПИСАТЬ ЗАПИСЬ-ОБН. ПЕРЕЙТИ К ЧТЕНИЕ.
КОНЕЦ.
   ЗАКРЫТЬ МАССИВ-НАЧИСЛЕНИЙ,
   МАССИВ-ОБЛАГАЕМЫХ-СУММ, МАССИВ-ФАМИЛИЙ, ОБНОВ-МАССИВ,
   ВЕДОМОСТЬ.
ВЫХОД.
   ВЫЙТИ ИЗ ПРОГРАММЫ.
```

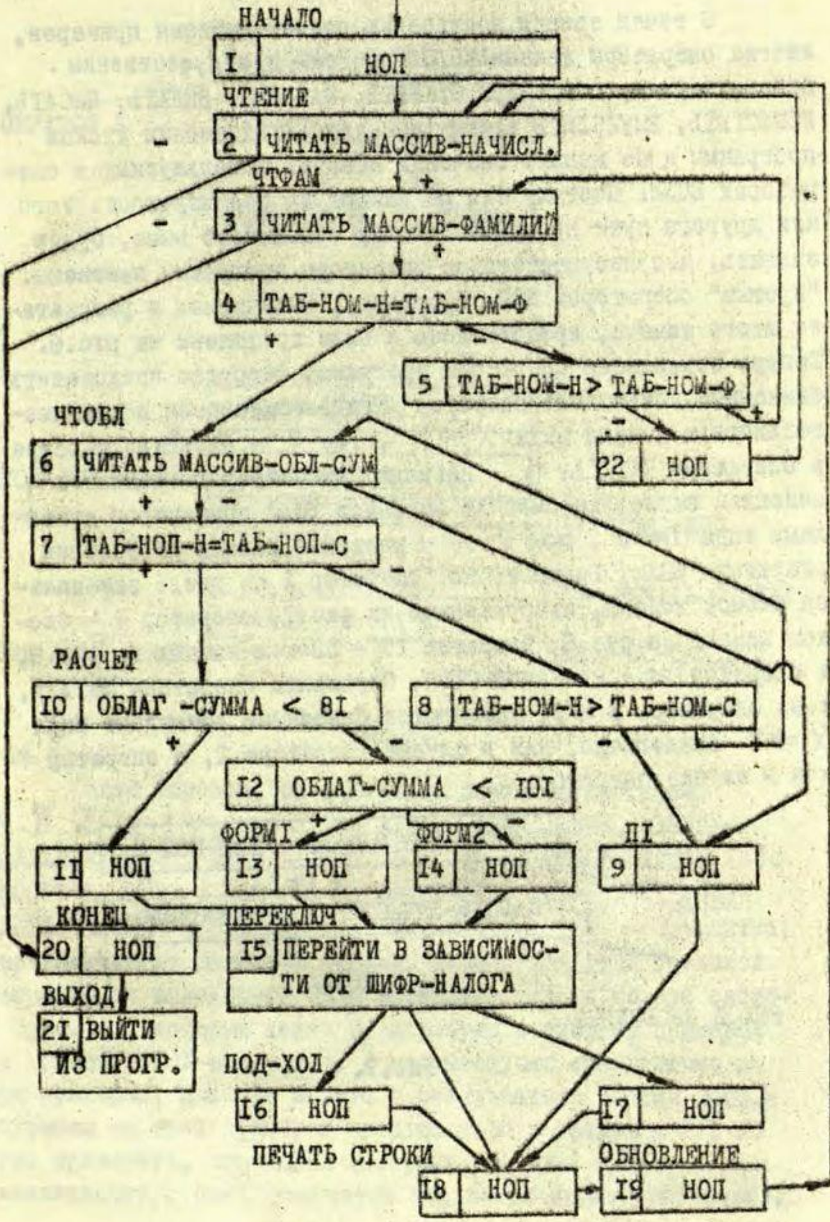

НАЧАЛО
I НОП
ЧТЕНИЕ
2 ЧИТАТЬ МАССИВ-НАЧИСЛ.
ЧТФАМ
3 ЧИТАТЬ МАССИВ-ФАМИЛИЙ
4 ТАБ-НОМ-Н=ТАБ-НОМ-Ф
5 ТАБ-НОМ-Н > ТАБ-НОМ-Ф
ЧТОБЛ
6 ЧИТАТЬ МАССИВ-ОБЛ-СУМ
22 НОП
7 ТАБ-НОП-Н=ТАБ-НОП-С
РАСЧЕТ
IO ОБЛАГ -СУММА < 8I
8 ТАБ-НОМ-Н > ТАБ-НОМ-С
I2 ОБЛАГ-СУММА < IOI
ФОРМI
ФОРМ2
ПI
II НОП
I3 НОП
I4 НОП
9 НОП
КОНЕЦ
20 НОП
ПЕРЕКЛЮЧ
I5 ПЕРЕЙТИ В ЗАВИСИМОСТИ ОТ ШИФР-НАЛОГА
ВЫХОД
21 ВЫЙТИ ИЗ ПРОГР.
ПОД-ХОД
I6 НОП
I7 НОП
ПЕЧАТЬ СТРОКИ
ОБНОВЛЕНИЕ
I8 НОП
I9 НОП

Рис. 6.

С точки зрения построения полной системы примеров, многие операторы данной КОБОЛ-программы несущественны. Действительно, операторы ОТКРЫТЬ, ЗАКРЫТЬ, ВЫДАТЬ, ПИСАТЬ, ПОМЕСТИТЬ, ВЫЧИСЛИТЬ всегда принадлежат линейным кускам программы и не меняют значений данных, используемых в операторах ЕСЛИ. Поэтому они не влияют на реализуемость того или другого пути программы. Ввиду сказанного выше, будем считать, что несущественные операторы программы заменены "пустым" оператором НОП. Программа, полученная в результате этого замена, представлена в виде графсхемы на рис.6. Теперь оставшиеся операторы программы нетрудно представить базисными командами: оператор ЧИТАТЬ заменяется последовательностью команд вида $X \Rightarrow t_i$, где X - название массива в операторе ЧИТАТЬ, t_i - названия элементарных данных в описании записи массива X; оператор ЕСЛИ заменяется командами вида $t < v$, где t, v - названия данных из условия оператора ЕСЛИ. Более точно, оператор 2 из рис.6 заменяется блоком команд, изображенным на рис.7, оператор 4 - блоком команд на рис.8, оператор 15 - блоком команд на рис.9, а оператор 21 - командой СТОП. Остальные операторы ЧИТАТЬ, т.е. операторы 3 и 6, заменяются базисными командами вида $X \Rightarrow t$ аналогично, как в случае оператора 2, а оператор 7- как в случае оператора 4.

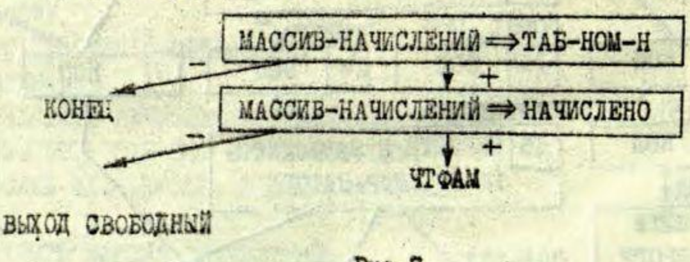


Рис.7.

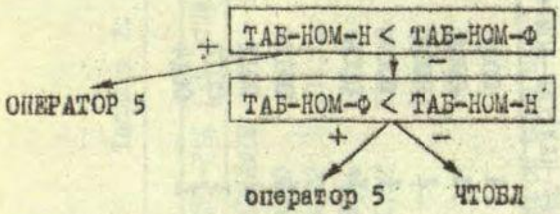


Рис.8.

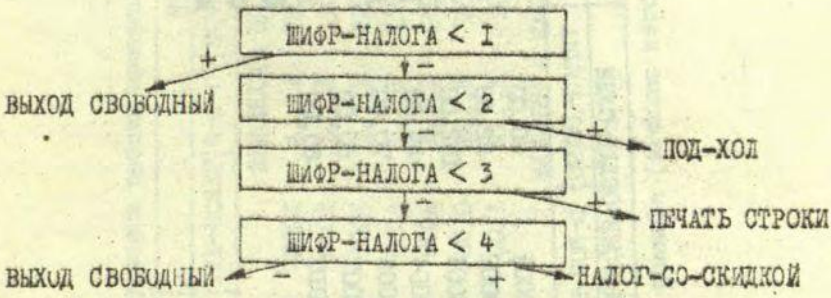


Рис.9.

Таким образом, мы построили базисную программу $\{\mathcal{X}, \mathcal{Y}, \mathcal{Z}, \Pi\}$, где $\mathcal{X} = \{$ МАССИВ-НАЧИСЛЕНИЙ, МАССИВ-ОБЛАГАЕМЫХ-СУММ, МАССИВ-ФАМИЛИИ $\}$, $\mathcal{Y} = \{$ ОБНОВ-МАССИВ, ВЕДОМОСТЬ $\}$, $\mathcal{Z} = \{$ ТАБ-НОМ-Н, НАЧИСЛЕНО, ТАБ-НОМ-Ф, ШИФР-НАЛОГА, ФАМИЛИЯ, ТАБ-НОМ-С, ОБЛАГ-СУММА $\}$, Π - графсхема, составленная из базисных команд. Далее, для этой базисной программы по описанному выше алгоритму строим полную систему примеров, которую затем преобразуем в систему примеров для данной КОБОЛ-программы. В нашем случае она состоит из двух примеров, которые вместе с результатами работы КОБОЛ-программы на этих примерах представлены в таблицах I и 2. Легко проверить, что КОБОЛ-программа на этих примерах останавливается, и они реализуют все ветви данной программы.

Примеры (входные массивы) Таблица I.

ПРИ-МЕР	ЗА-ПИСЬ	МАССИВ-НАЧИСЛЕНИЙ		МАССИВ-ОБЛАГ-СУММ		МАССИВ-ФАМИЛИЙ		
		ТАБ-НОМ-Н	НАЧИСЛЕНО	ТАБ-НОМ-С	ОБЛАГ-СУММА	ФАМИЛИЯ	ШИФР-НАЛ	ТАБ-НОМ-Ф
I	I	000I	I00,00	000I	80,00	А...А	I	000I
	2	0002	II0,00	0002	82,00	Б...Б	3	0002
	3	0003	I20,00	0003	I02,00	В...В	2	0003
	4	0004	I30,00	0004	84,00	Г...Г	I	0004
	5	0006	I40,00	0005	70,00	Д...Д	I	0006
	6	0008	I50,00	0007	60,00			
2	I	00I0	60,00	00I0	80,00	Е...Е	I	00I0
	2	00II	50,00			Ж...Ж	2	00II
	3	00I3	40,00			И...И	3	00I2
	4					К...К	I	00I4

Результаты работы программы. Таблица 2.

ПРИ-МЕР	ЗА-ПИСЬ	ОБНОВ-МАССИВ		ВЕДОМОСТЬ				
		ТАБ-НОМ-ОБН	ОБЛАТ-СУММА-ОБН	ТАБ-НОМЕР	Ф-И-О	НАЧИСЛ-РЕД	УДЕРЖ-РЕД	НА-РУКИ
					ВЕДОМОСТЬ НА ВЫПЛАТУ ЗАРПЛАТЫ			
I	I	000I	I00,00	000I	А...А	I00,00	4,80	95,20
	2	0002	II0,00	0002	Б...Б	II0,00	4,22	I05,78
	3	0003	I20,00	0003	В...В	I20,00	8,46	III,54
	4	0004	I30,00	0004	Г...Г	I30,00	II,32	II8,68
	5	0006	I40,00	0006	Д...Д	I40,00	-	I40,00
2	I	00I0	60,00	00I0	Е...Е	60,00	-	60,00
	2	00II	50,00	00II	Ж...Ж	50,00	-	50,00
					ФАМИЛИИ С ТАБЕЛЬНЫМ НОМЕРОМ 00I3 НЕТ			

Таким образом, в таблице I предоставлена полная система примеров для данной КОБОЛ-програмы.

Из рассмотренного примера уже видно, что наиболее существенными ограничениями для применения описанного выше алгоритма при построении полной системы примеров являются две:

I. данные, используемые в условных операторах (включая оператор ПЕРЕЙТИ), не должны быть предварительно деформированы арифметическими операторами,

2. все массивы должны просматриваться последовательно, и каждый массив разрешается открывать не более одного раза (т.е. каждый массив разрешается считывать не более одного раза).

Однако на практике для достаточно большой доли задач обработки данных эти ограничения естественно выполняются. Поэтому описанный алгоритм предоставляет практический интерес. Проверка данного алгоритма на ряде примеров показала, что соответствующие полные системы примеров получаются весьма небольшими и они хорошо выявляют ошибки програмы. Более подробное теоретическое и эвристическое обоснование этого явления требует дальнейших исследований.

ЛИТЕРАТУРА

1. McCarthy J., Painter J. Correctness of a compiler for arithmetic expressions.- Proceedings of a Symposium in Applied Mathematics 19, American Mathematical Society, 1967.
2. Scott D. Outline of a Mathematical Theory of Computation. - Proc. 4th Princeton Conference on Information Science and Systems, 1970.
3. Milner R., Weyhrauch R. Proving compiler correctness in a mechanized logic.- Machine Intelligence 7, Edinbourgh, Edinbourgh University Press, 1972.

4. Weyhrauch R., Milner R. Program semantics and correctness in a mechanized logic.- Proceedings of USA-Japan Computer Conference, Tokyo, 1972.
5. Калниньш А.А., Бичевский Я.Я., Барздинь Я.М. Разрешимые и неразрешимые случаи проблемы построения полной системы примеров.- Настоящий сборник. стр.I88-205.
6. Куликовская В.П., Романовская Л.М. и др. КОБОЛ ЭВМ "Минск-32", I973.

РАЗРЕШИМЫЕ И НЕРАЗРЕШИМЫЕ СЛУЧАИ ПРОБЛЕМЫ ПОСТРОЕНИЯ ПОЛНОЙ СИСТЕМЫ ПРИМЕРОВ

А.А.Калниньш, Я.Я.Бичевский, Я.М.Барздинь

I. Введение. В статье [I] настоящего сборника было рассмотрено построение полной системы примеров для программ одного простого языка программирования. Приведенный в статье пример на языке КОБОЛ показывает, что введенная там система команд (называемая базовой) достаточна для исследования проблемы построения полной системы примеров для многих реальных задач обработки данных. Но имеется много сравнительно простых программ, которые анализировать с помощью базовой системы команд нельзя. В основном это происходит из-за того, что в логических командах таких программ используются простым образом деформированные реквизиты или эти программы имеют более сложную логику работы с лентами (массивами). Так как для произвольных программ в реальных языках программирования практически нельзя эффективно построить полные системы примеров, то представляет интерес поиск таких систем команд, которые находятся на границе разрешимости проблемы построения полной системы примеров. Мы будем дополнять базовую систему команд: $X \Rightarrow t {\nearrow \atop \searrow}, t \Rightarrow y, t \Rightarrow v, c \Rightarrow v, t < v {\nearrow \atop \searrow}, t < c {\nearrow \atop \searrow}$, НОП и СТОП новыми командами. В настоящей работе рассматриваются четыре такие системы команд K_1, K_2, K_3, K_4 (базовую систему команд будем обозначать через K_0). Первые три из этих систем команд допускают простые арифметические преобразования элементарных данных, участвующих в логических командах, а именно -добавление и вычитание константы. Эти расширения базовой системы команд охватывают типичные применения счетчиков в реальных программах. Четвертая система команд K_4 допускает повторное считывание входных лент. К сожалению, уже довольно скоро проблема построения полной системы примеров становится неразрешимой.

Понятия програмы, ветви программы и примера для рассматриваемых нами систем команд полностью аналогичны соответствующим понятиям из [I], так как они не зависят от выбора конкретной системы команд. Поэтому сохраняется понятие полной системы примеров (ПСП), введенное в [I], как системы допустимых примеров, на которых реализуемы все ветви программы, которые вообще реализуемы на каком-нибудь допустимом примере (допустимый пример - это пример, на котором программа доходит до команды СТОП и по пути не имеет аварий, т.е. не попадает на свободный выход какой-нибудь команды).

Сохраняется также понятие зацикливания программы - как бесконечной работы программы на конечном примере.

2. Система команд K_1. В каждой программе выделяется набор специальных внутренних ячеек - счетчиков $z_1, \ldots, z_r$ (начальные значения этих ячеек, как и всех внутренних ячеек, равны 0). Для счетчиков вводится новая команда $z_i + 1 \Rightarrow z_i$ (смысл ее очевиден). Из команд системы K_0 для счетчиков допускаются только команды $c \Rightarrow z_i$ и $z_i < c$. Для лент и остальных внутренних ячеек допускаются все команды системы K_0. Полученную таким образом новую систему команд обозначим через K_1.

ТЕОРЕМА I. Существует алгоритм, который строит ПСП для любой программы в системе команд K_1.

Доказательство этой теоремы аналогично доказательству теоремы I из [I]. Изменяется только определение состояния. Согласно [I], состояние - это граф, характеризующий отношения между записями, содержащимися в ячейках $t_1, \ldots, t_p$. В нашем случае к состоянию добавляется еще информация о счетчиках - вектор $B = \langle b_1, \ldots, b_r \rangle$ (r - число счетчиков). Пусть $\bar{c} = \max |c_j|$, где максимум берется по всем константам c_j нашей программы, входящим в команды вида $c_j \Rightarrow z_i$ или $z_i < c_j$. Тогда значение вектора B в данной точке пути (перед выполнением очередной команды) определяется следующим образом: $b_i = \langle z_i \rangle$ (содержимое z_i), если $|\langle z_i \rangle| \leq \bar{c}$, и $b_i = \bar{c}+1$, если $\langle z_i \rangle > \bar{c}$. (Случай $\langle z_i \rangle < -\bar{c}$ в программах системы K_1 появиться не может). При таком определении состояния выполняется точный аналог леммы 3 из [I]. Доказательство этого аналога в точности совпадает с доказательством леммы 3 из [I], за исключением случая, когда на рассматриваемом пути значения счетчика становятся больше $\bar{c}$. Но в данном случае, как легко понять, поведение программы уже не зависит от конкретного значения счетчика, бо́льшего $\bar{c}$, так как в командах $z_j < c_l$ все время будет осуществляться переход по "-". Такой переход может прекратить только появление команды вида $c_k \Rightarrow z_j$, но в таком случае аннулируется прежнее значение счетчика. Это означает, что если в рассмотренном случае мы заменим значение счетчика на $\bar{c}+1$, то дальнейшее поведение программы от этого не изменится.

Поскольку аналог леммы 3 из [I] выполняется, то дальнейшее доказательство теоремы I полностью совпадает с доказательством теоремы I из [I].

Столь же просто можно доказать аналог теорем 2 и 3 из [I] для системы команд K_1.

3. Система команд K_2 отличается от системы команд K_1 тем, что для любого счетчика дополнительно разрешается пользоваться командой $z_i - 1 \Rightarrow z_i$. Таким образом, в данном случае под счетчиком понимается обычный счетчик, к которому можно как прибавить, так и отнять константу.

Известно [2], что добавление одного счетчика к конечному автомату сохраняет разрешимость основных алгоритмических проблем над автоматами. Аналогичная ситуация имеет место и здесь.

ТЕОРЕМА 2. Существует алгоритм, который строит ПСП для любой программы в системе команд K_2 с одним счетчиком.

Основная идея доказательства будет заложена в лемме I. Пусть дана программа P в системе команд K_2 с одним счетчиком z. Говоря содержательно, мы хотим показать, что если в программе P какая-то команда (или дуга) достижима, то она достижима на таком пути, на котором значение счетчика по абсолютной величине не превосходит значения некоторой эффективной функции от размера P (числа команд, числа ячеек памяти, величины и числа констант). Один из вариантов такой функции можно задать на языке состояний.

Если в программе P заменить команды $z+1 \Rightarrow z$, $z-1 \Rightarrow z$, $c \Rightarrow z$ на команды НОП, а $z<c \lessgtr^{+}_{-}$ на НОП с фиксированным переходом по "+", то получим программу P' в системе команд K_0 (аналогично P'', если в $z<c \lessgtr^{+}_{-}$ фиксируются переходы по "-"). Если теперь считать начальной какую-нибудь команду a программы P', то эффективно можно получить множество всевозможных состояний $\bar{S}_a$ этой программы в смысле [I]. Теперь определим $n(P)$ равным максимуму от $|\bar{S}_a|$ по всем командам a программ P' и P''. Пусть $\bar{c}$, как и выше, равно максимальному $|c_i|$ в командах вида $z<c_i$ или $c_i \Rightarrow z$. Далее, пусть q - число команд в программе P, тогда произведение nq будем использовать для характеристики размеров программы, а $2(qn)^3 + \bar{c}$ будет служить ограничивающей функцией для счетчика.

ЛЕММА I. Если в программе P в системе команд K_2 для некоторой ветви программы существует реализуемый СТОП-путь α (т.е. заканчивающийся командой СТОП), содержащий эту ветвь, то существует также реализуемый СТОП-путь β, содержащий эту ветвь, который обладает тем свойством, что на нем все время $|\langle z\rangle| < 2(qn)^3+\bar{c}$.

Для доказательства леммы рассмотрим для пути α все его максимальные отрезки, в которых $\langle z\rangle > \bar{c}$ или $\langle z\rangle < -\bar{c}$.

Поскольку для $\langle z\rangle < -\bar{c}$ все рассуждения симметричны, рассмотрим только случай $\langle z\rangle > \bar{c}$. Пусть $a_1,\ldots,a_u$ один такой отрезок; $s_1,\ldots,s_u$ - состояния в смысле [I] (т.е. без учета значения счетчика) соответственно перед выполнением команд $a_1,\ldots,a_u$; $Z_1,\ldots,Z_u$ - значения счетчика z соответственно перед выполнением команд $a_1,\ldots,a_u$; s_{u+1} и Z_{u+1} - соответственно состояние и значение счетчика после выполнения команды a_u. Уточним начало и конец выбранного отрезка так, что $Z_1 \leq \bar{c}$, $Z_{u+1} \leq \bar{c}$, а все остальные $Z_i > \bar{c}$. Очевидно, Z_{i+1} равно либо $Z_i + 1$, либо $Z_i - 1$, либо Z_i. (Случай команды вида $c \Rightarrow z$ - он может относиться только к переходу от Z_u к Z_{u+1} - пока не будем рассматривать). Находим наименьшее i_0, при котором в данном отрезке достигается максимум $\langle z\rangle$, т.е. $Z_{i_0} = \max\limits_{1\leq i\leq u+1} Z_i$. Часть пути до i_0 назовем восходящей, часть с i_0 до u - нисходящей.

Пока мы находимся в выбранном отрезке, реализуемость пути α не зависит от конкретного значения z, так как во всех командах $z < c$ осуществляется переход по "-". Считая указанные переходы в командах $z < c$ временно фиксированными, мы фактически работаем с системой команд K_0, а в таком случае можно применить лемму 3 из [I]. Поэтому имеет место следующее основное утверждение: если в пути α на выбранном отрезке $a_1,\ldots,a_u$ существуют i_1 и i_2, такие, что $s_{i_1} = s_{i_2}$ и $a_{i_1} = a_{i_2}$, то реализуем путь α', начало которого до выбранного отрезка совпадает с началом α, а дальше идет выбранный отрезок, из которого исключен кусок $a_{i_1},\ldots,a_{i_2-1}$ (конечно, при условии, что в α' сохраняется условие $Z_i > \bar{c}$). Но в дальнейшем из выбранного отрезка мы постараемся исключить такие группы команд, чтобы продолжение α' за этим отрезком снова совпало с α и было реализуемо.

Для дальнейшего доказательства леммы опишем еще несколько конструкций. В восходящей части выбранного отрезка $a_1,\ldots,a_u$ для каждого v, где $\bar{c} < v < Z_{i_0}$, находим

l_v такое, что $Z_{l_v}=v$, а для всех j, где $l_v<j<i_0$; $Z_j>v$. Такое l_v всегда существует согласно свойствам Z_i. Аналогично в нисходящей части для этих же v находим l'_v такие, что $Z_{l'_v}=v$, а для всех j, где $i_0\leq j<l'_v$, $Z_j>v$. Рассмотрим две последовательности пар: $\{A_v\}$, $A_v=(s_{l_v},a_{l_v})$ и $\{A'_v\}$, $A'_v=(s_{l'_v},a_{l'_v})$, где $v=\bar{c}+1,\ldots,Z_{i_0}-1$. Если команда $z<c$ находится в выбранном отрезке, то у нее фиксирован переход по "-", и поэтому можно считать, что мы работаем с программой P'', считая команду a_1 начальной. Поэтому число различных достижимых состояний s_i в отрезке не превосходит $|\bar{S}_{a_1}|<n$, и тем самым число различных пар в последовательностях $\{A_v\}$ и $\{A'_v\}$ не превосходит nq. Теперь в восходящей части выделяем интервалы повторения пар. Для этого сначала отмечаем точки повторения v_k следующим образом: v_1 равно минимальному j, такому, что $\exists v'_1\ (\bar{c}<v'_1<j\ \&\ A_{v'_1}=A_j)$, а v_k, $k>1$ равно минимальному j, такому, что $\exists v'_k(v_{k-1}\leq v'_k<j\ \&\ A_{v'_k}=A_j)$ Поиск v_k повторяем до тех пор, пока не будет достигнуто $A_{z_{i_0}-1}$. k-тый интервал повторения содержит пары $A_{v'_k},A_{v'_k+1},\ldots,A_{v_k-1}$, где v'_k - число, входящее в определение v_k, т.е., $A_{v'_k}=A_{v_k}$; этому интервалу соответствуют команды $a_{l_{v'_k}},a_{l_{v'_k}+1},\ldots,a_{l_{v_k}-1}$. Изменение $\langle z\rangle$ во время k-того интервала повторения назовем z-длиной этого интервала и обозначим через Δ_k; тогда $\Delta_k=v_k-v'_k$. Из определения точек повторения следует, что $\Delta_k\leq v_k-v_{k-1}\leq nq$ для $k>1$, а $\Delta_1<v_1-\bar{c}<nq$. Аналогично выделяем интервалы повторения в нисходящей части, начиная с $v_1>\bar{c}$. Если некоторые из интервалов повторения содержат исследуемую ветвь, то интервалы, первый раз содержащие ее, не включаются в общий счет интервалов; ясно, что общее число команд в них меньше $2n$.

В результате выделения интервалов получаем, что в восходящей части интервал с z-длиной m встречается γ_m раз, а в нисходящей части - δ_m раз, где $\gamma_m,\delta_m\geq 0$; $m=1,\ldots,nq$. Ясно, что исключение интервала повторения из восходящей части пути уменьшает значение $<z>$ в конце отрезка на z -длину интервала, а из нисходящей - увеличивает. Постараемся исключить несколько интервалов из обеих частей сразу так, чтобы значение $<z>$ в конце отрезка не менялось. В результате мы получаем путь α', имеющий после рассматриваемого отрезка то же самое продолжение, что и α, и этот путь, согласно основному утверждению, должен быть реализуем, так как после отрезка у α и α' совпадают состояния и значения $<z>$ (т.е., s_{u+1} и Z_{u+1}) и следуют одни и те же команды. Выполнение требования из основного утверждения о том, что в пути α' на выбранном отрезке все $Z_l > \bar{c}$, гарантируется способом выбора l_v и l_v'. Пусть из восходящей части интервал с z -длиной m исключается γ_m' раз, а с нисходящей - δ_m' раз, при этом обязательно $0 \leq \gamma_m' \leq \gamma_m$, $0 \leq \delta_m' \leq \delta_m$. Тогда для сохранения Z_{u+1} должно выполняться равенство:

$$\sum_{m=1}^{nq} m\gamma_m' = \sum_{m=1}^{nq} m\delta_m' . \tag{I}$$

Если выполняется условие $\sum_{m=1}^{nq} \gamma_m \geq (nq)^2$ и $\sum_{m=1}^{nq} \delta_m \geq (nq)^2$, то существует m_1 такое, что $\gamma_{m_1} \geq nq$ и m_2, такое, что $\delta_{m_2} \geq nq$. Следовательно, при этом же условии существует набор допустимых значений γ_m' и δ_m', удовлетворяющих (I), а именно -положим $\gamma_{m_1}' = m_2$, $\delta_{m_2}' = m_1$, а остальные γ_m' и δ_m' равными 0. Но тогда реализуем путь α', отличающийся от пути α только на выбранном отрезке, где исключены соответствующие интервалы и тем самым уменьшился максимум $<z>$. Указанную операцию можно продолжать, по крайней мере, до тех пор, пока в новом пути α' $\sum_{m=1}^{nq} \gamma_m < (nq)^2$ и $\sum_{m=1}^{nq} \delta_m < (nq)^2$. Но $\sum_{m=1}^{nq} \gamma_m$ является общим числом интервалов в восходящей части. Учитывая, что $v_1 < nq + c$, $v_k - v_{k-1} \leq nq$ и сумма длин неучитываемых частей пути (от последнего v_k до Z_{l_0} и части, содержащей исследуемую ветвь) меньше $3nq$, получаем оценку для максимума $<z>$ на выбранном отрезке пути α': $Z_{l_0} < (nq)^3 + 3nq + \bar{c}$, то при $nq > 1$ можно заменить на $2(nq)^3+\bar{c}$. То же самое проделывается для всех отрезков пути, где $|<z>|>\bar{c}$; получается путь α°. (Для отрезков, заканчивающихся на команду $c \Rightarrow .z$, применяя соображения, использованные при доказательстве теоремы I, легко получаем оценку $nq+\bar{c}$). Ясно, что путь α° содержит исследуемую ветвь. Заметим еще, что в пути α может существовать последний отрезок, где до конца пути $|<z>|>\bar{c}$. Тогда эта часть пути по существу равносильна программе без счетчика, и поэтому последний отрезок может быть сделан короче, чем nq.

Лемма доказана.

Чтобы завершить доказательство теоремы, остается заметить, что, согласно лемме I, при построении ПСП можно ограничиться путями, в которых $|<z>|<2(nq)^3+\bar{c}$. Поэтому можно использовать следующий вид состояния: к состоянию в смысле [I] добавляется число b (обрезанное значение счетчика), где $b=<z>$, если $|<z>|<2(nq)^3+\bar{c}$, и $b=2(nq)^3+\bar{c}+1$ – в противном случае. Для построения ПСП используется тот же алгоритм теоремы I из [I]; единственное дополнение состоит в том, что пути, достигшие состояния с $b=2(nq)^3+\bar{c}+1$, обрезаются.

Теорема доказана.

Аналогично доказывается также следующая

ТЕОРЕМА 3. Существует алгоритм, который для любой программы в системе команд K_2 с одним счетчиком выясняет, может ли она зацикливаться.

В этом алгоритме используется то же обрезание путей по значению $|<z>|$, превосходящему $2(nq)^3+\bar{c}$, и то же понятие состояния, что в теореме 2. Используется также алгоритм из аналогичной теоремы 2 в [I], но к нему необходимо дополнение, а именно,- более сложно определяется самоповторение. Дело в том, что при бесконечной работе программы значение $<z>$ может как периодически колебаться с ограниченной амплитудой, так и неограниченно возрастать. Натрудно показать, что если после того, когда произведено последнее считывание с ленты, значение $|\langle z\rangle|$ достигло $2nq+\varepsilon$ и продолжает возрастать, то оно возрастает неограниченно; поэтому амплитуда колебаний должна быть меньше $2nq+\varepsilon$. Кроме того, так же просто показывается, что если программа может зацикливаться с бесконечным возрастанием $|\langle z\rangle|$, то это зацикливание такого же вида, как в системе команд K_0, причем тогда реализуем бесконечный путь, где значение $|\langle z\rangle|$ в начале зацикливания меньше $3nq+\varepsilon$. Поэтому самоповторение в дереве реализуемости определяем следующим образом: если на некоторой ветви повторяется команда α соответственно в состояниях s'_β и s'_γ (состояния в смысле теоремы 2), то для самоповторения необходимо совпадение граф-компонент этих состояний. Для обрезанных значений счетчика b_β и b_γ должно выполняться следующее: если $|b_\gamma| \leq 3nq+\varepsilon$, то должно быть $b_\beta = b_\gamma$, если $|b_\gamma| > 3nq+\varepsilon$, то должно быть $|b_\gamma| \geq |b_\beta|$.

Имеет место также аналог теоремы 3 из [I].

К сожалению, если программа в системе команд K_2 использует не один, а несколько счетчиков, то для таких програмы уже не существует алгоритма построения ПСП. Это связано с тем, что в системе команд K_2, используя два счетчика, можно реализовать любую двухленточную машину Минского ([3], [4]), а проблема построения ПСП тесно связана с алгоритмически неразрешимой проблемой останова для этой машины. Из двух эквивалентных определений машины Минского мы используем данное в [4] - т.е. определяем ее как машину с двумя регистрами (счетчиками), которая может выполнять команды: прибавить I, вычесть I, сравнить с 0 значение регистра и СТОП, т.е. команды, входящие в систему K_2.

Более точно под алгоритмом построения ПСП мы понимаем общерекурсивную функцию, которая коду программы сопоставляет код набора массивов, образующих ПСП (или код пустого набора, если ни одна ветвь нереализуема на примере, который доходит до команды СТОП).

ТЕОРЕМА 4. Не существует алгоритма, который строит ПСП для любой программы в системе команд K_2 с двумя счетчиками.

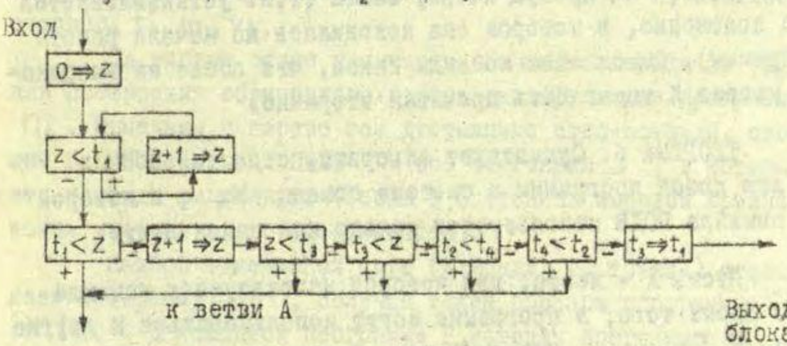
Для доказательства остается только показать, как свести проблему останова машины к проблеме построения ПСП. Для любой двухленточной машины Минского M и числа x эффективно строим программу $P(M,x)$ в системе команд K_2. $P(M,x)$ использует одну ленту T, одну ячейку t и два счетчика z_1 и z_2 (соответствующие регистрам M). Программа $P(M,x)$ начинается с команды $T \Rightarrow t$ с переходом на следующую команду K независимо от выхода "+" или "-". Следующая команда K равна $x \Rightarrow z_1$ (x здесь выступает как константа c), затем следует $\circ \Rightarrow z_2$, а далее - команды M, начиная с первой. Ясно, что если M останавливается на x, то любой массив T образует ПСП, а в противном случае ПСП является пустым набором.

Совершенно аналогично доказывается следующее утверждение: не существует алгоритма, выясняющего, может ли данная программа в системе команд K_2 зацикливаться. Можно доказать также, что не существует алгоритма, строящего конечный набор примеров, на которых реализуемы все ветви данной программы, которые вообще реализуемы на каком-нибудь примере - не важно, останавливается на нем программа или нет.

4. Система команд K_3 получается из системы команд K_1 добавлением команд вида $z_i < t_j$ и $t_j < z_i$, где z_i - счетчик, а t_j - обычная внутренняя ячейка. Оказывается, что для системы команд K_3 проблема построения ПСП неразрешима даже для программ с одним счетчиком.

ТЕОРЕМА 5. Не существует алгоритма, строящего ПСП для любой программы в системе команд K_3 с одним счетчиком.

Для доказательства этой теоремы используется та же связь с двухленточной машиной Минского, что и в теореме 4, только здесь невозможно прямое моделирование машины. Вместо этого строится программа в K_3, которая проверяет, является ли данный массив последовательностью конфигураций данной машины. Более точно, для данной двухленточной (двухрегистровой) машины Минского M и аргумента y, записанного в первом регистре, последовательность конфигураций $x_1, x_2, x_3, \ldots$ определяется следующим образом: $x_1 = y$, $x_2 = 0$ и пара x_{2k+1}, x_{2k+2} – это числа, записанные в первом и втором регистрах после k – го такта работы M (т.е. после того, как M выполнила k команд). Если M останавливается на u -том такте (т.е. в этом такте выполняется стоп-команда), то последовательность обрывается на x_{2u+2}, в противном случае она бесконечна. Для машины Минского M и аргумента y мы эффективно строим программу $P(M,y)$ в системе команд K_3, использующую одну ленту X, 4 внутренние ячейки t_1, t_2, t_3, t_4 и один счетчик z, который останавливается, если массив X содержит всю конечную последовательность конфигураций M на y, и зацикливается в противном случае. Это означает, что $P(M,y)$ может останавливаться, причем только на одном примере, тогда и только тогда, когда M останавливается на y. $P(M,y)$ работает следующим образом: сначала считываются первые две записи X в ячейки t_1 и t_2 и сравниваются соответственно с константой y и 0. При совпадении работа продолжается, в противном случае программа переходит на специальную ветвь А. Далее выполняется стандартный цикл – выполнение одного из блоков, соответствующего команде машины M. Блок, соответствующий команде машины M, имеет следующий вид. Сначала считываются две записи X в ячейки t_3 и t_4 (тогда в ячейках t_1, t_2, t_3, t_4 находятся соответственно записи $X^{2k-1}, X^{2k}, X^{2k+1}, X^{2k+2}$, показывающие, как изменяется конфигурация во время k – го такта M). Если массив X кончился, осуществляется переход на ветвь А. Далее идет группа команд, непосредственно моделирующая саму команду М. Если эта команда - сравнение с нулем одного из регистров, то проверяются равенства $t_1 = t_3$ и $t_2 = t_4$ (так как конфигурация не меняется), а затем осуществляется сравнение t_3 или t_4 с нулем и переход по соответствующему выходу блока. Для команд прибавления или вычитания I из регистра соответствующая группа команд более сложная. Например, для прибавления единицы к первому регистру она имеет следующий вид (в этом случае должно проверяться, верно ли, что $<t_3> = <t_1> + 1$ и $t_2 = t_4$):



Другие случаи реализуются аналогично. Во всех "неправильных" ситуациях осуществляется переход на ветвь А, а в конце помещается команда подготовки к следующему циклу ($t_3 \Rightarrow t_1$ или $t_4 \Rightarrow t_2$). Выход блока подключается к входу следующего блока (согласно программе М). Блок, соответствующий стоп-команде, сначала считывает соответствующие записи X в ячейки t_3 и t_4 и проверяет равенства $t_1 = t_3$ и $t_2 = t_4$, а затем действительно выполняет команду СТОП. Ветвь А состоит из тривиального цикла, например, $\rightarrow \boxed{t_1 \Rightarrow t_1}$

Из конструкции $P(M, y)$ ясно, что она обладает необходимыми свойствами. Это означает, что ПСП для $P(M, y)$ непуста тогда и только тогда, когда М останавливается на y. Теорема доказана.

Аналогично доказывается, что не существует алгоритма, выясняющего, может ли данная программа в K_3 зациклиться.

Полученный выше результат показывает также, почему к системе команд K_0 нельзя добавить хотя бы самую простую арифметическую команду для обычных ячеек, например, вида $t+1 \Rightarrow t$, если мы хотим сохранить разрешимость проблемы построения ПСП.

5. Система команд K_4 получается из K_0 добавлением одной новой команды ПОДВ X, где X - название ленты. По этой команде головка, связанная с лентой X, передвигается на правую ячейку ленты (т.е. устанавливается в то положение, в котором она находилась до начала работы программы). Смысл этой команды таков, что после ее выполнения массив X может быть прочитан вторично.

ТЕОРЕМА 6. Существует алгоритм, строящий ПСП для любой программы в системе команд K_4 , в которой команда ПОДВ используется только для одной ленты.

Пусть X - лента, для которой используется команда ПОДВ. Кроме того, в программе могут использоваться и другие ленты $Y, T, \ldots$, читаемые, как обычно, - только один раз сначала. Алгоритм, строящий ПСП для программ в системе команд K_4 , также является модификацией соответствующего алгоритма для системы команд K_0 , только более существенной, чем для систем команд K_1 и K_2 .

Основную часть алгоритма построения ПСП, описанного в [I]$_1$ - алгоритм построения обрезанного дерева развертки - мы будем применять несколько раз - первый раз для образования путей в программе, как и в [I], второй раз-для образования пар путей в двух программах и т.д. Из исходной программы P мы образуем несколько программ $П_0, П_1, П_2$ следующим образом. Все команды ПОДВ X в P заменяем стоп-командами. Для программы $П_0$ начальной командой считаем начальную команду P, начальным состоянием - начальное состояние P. Другие программы $П_1, П_2, \ldots$ сопоставляются парам (вхождение команды ПОДВ X, состояние s) в программе P, причем начальной командой в соответствующей программе Π_i считается команда, в которую ведет дуга из данного вхождения команды ПОДВ X в программе P. Состояние s, в свою очередь, считается начальным состоянием. Каким именно командам ПОДВ X и состояниям сопоставляются программы, будет указано дальше, но ясно, что общее число возможных таких Π_i конечно. Лента X остается в каждой из программ, но для остальных лент $Y, T, \ldots$ и внутренних ячеек $t, y, v, \ldots$ в каждой из Π_i вводится свой экземпляр $Y_i, T_i, \cdots$ и, соответственно, $t_i, y_i, v_i, \cdots$.

На первом этапе применяем обычный алгоритм из [I] для построения обрезанного дерева развертки для программы Π_0. Отмечаем в дереве все достижимые стоп-команды, стоящие на месте команд ПОДВ X и все состояния s, в которых эти команды выполняются (если в состоянии имеется признак конца массива X, то он снимается).

Каждой отмеченной паре (команда, состояние) сопоставляем программу Π_i, получая таким образом программы $\Pi_1, \ldots, \Pi_{k_1}$. В исходной программе P команды программ Π_i, $i=1,\ldots,k_1$ выполняются после программы Π_0 с использованием того же массива X и продолжения массивов $Y, T, \ldots$. Таким образом, путь α в программе P можно разбить на два пути : α_1 - часть, принадлежащая программе Π_0, и α_2 - программе Π_i. При этом в обеих программах читается один и тот же массив X. Массивы $Y_i, T_i, \cdots$, читаемые в Π_i, можно рассматривать как продолжения массивов $Y_0, T_0, \cdots$, вместе дающие массивы $Y, T, \ldots$. Сам путь α получается приписыванием пути α_2 в конце пути α_1 с обратной заменой СТОП на ПОДВ X. При этом, конечно, поставленная повторно команда ПОДВ X и состояние $s(\alpha_1)$ должны быть парой, определяющей Π_i. Если временно отбросить это требование, то можно рассматривать произвольную согласованную пару путей α_1 в Π_0 и α_2 в Π_i как такую пару путей, для которой существует пара примеров $X, Y_0, T_0, \cdots$ и $X, Y_i, T_i, \cdots$ (X -

общий для обоих), такая, что α_1 реализуем на $X, Y_0, T_0, \ldots$ и α_2 реализуем на $X, Y_1, T_1, \ldots$. Из согласованности α_1 и α_2 следует, что системы неравенств $N(\alpha_1)$ и $N(\alpha_2)$ (введенные в [I]), начинающиеся с соответствующих состояний, непротиворечивы. Заметим, что противоречия могут возникать вообще только для массива X (возможность продолжения исчерпанного массива Y не допускается состоянием). Для облегчения проверки непротиворечивости неравенств по X рассмотрим только синхронное построение пар путей α_1 и α_2 в программах Π_0 и Π_1 , т.е. такое, что все считывания массива X в обоих путях производятся одновременно. Более точно, с самого начала или после текущей команды чтения X, оба пути строятся команда за командой до тех пор, пока в одном достигается считывание X. Если в другом пути еще не достигнуто считывание X, то построение первого приостанавливается, пока в другом не достигается считывание X или СТОП. После достижения СТОП в одном из путей другой путь строится дальше как обычный один путь. При синхронном построении пар путей для выяснения согласованности может быть использован следующий вид состояния. Состояние $s(\alpha_1, \alpha_2)$ программ после выполнения путей α_1 и α_2 получается из объединения графов реализуемости [I] $G_1(\alpha_1)$ и $G_2(\alpha_2)$ таким же образом, как $s(\alpha)$ получается из $G(\alpha)$ в [I]. При объединении $G_1(\alpha_1)$ и $G_2(\alpha_2)$,конечно, имеется в виду, что каждому элементу X^i сопоставляется одна общая для обоих графов вершина, а другие переменные вершины остаются в каждом графе свои. Еще заметим, что у вершины с основной меткой вида X^i вспомогательные метки могут быть как вида t_0 , так и t_1 , но для сохранения вершины в состоянии она должна иметь метку хотя бы одного из этих видов.

Ясно, что для так определенного состояния выполняется аналог леммы 3 из [I]:

ЛЕММА 3'. Если в программах Π_0 и Π_1 имеются две пары согласованных синхронно построенных путей (α_1, α_2)

и (α_1', α_2'), оканчивающиеся на одну и ту же пару команд, и $s(\alpha_1,\alpha_2) = s(\alpha_1',\alpha_2')$, то для любой пары путей (β,γ) из согласованности пары $(\alpha_1+\beta, \alpha_2+\gamma)$ следует согласованность пары $(\alpha_1'+\beta, \alpha_2'+\gamma)$.

Теперь остается описать второй этап алгоритма построения деревьев развертки для пар программ $(\Pi_0, \Pi_1), \dots, (\Pi_0, \Pi_{k_1})$. Для пары (Π_0, Π_i) дерева развертки строится следующим образом (естественное обобщение алгоритма из [I]): пара начальных команд присваивается корню (0- му ярусу), далее из любой пары предыдущего яруса идут дуги на те пары в следующем ярусе (их не более 4), на которые идут дуги в паре программ (Π_0, Π_i). Если требуется приостановка из-за синхронизации, то одна компонента пары просто на время не меняется. То же самое делается, если одна из компонент равна команде СТОП, но в этом случае, как уже было сказано, прекращается синхронизация. Вершинам присваиваются также соответствующие состояния. Правила обрезания дерева развертки обычные - по останову, нереализуемости, несогласованности и повторению (пара команд, состояние).

После построения деревьев для всех пар (Π_0, Π_i) отмечаются достигнутые пары команд, где компонента из Π_i содержит команду СТОП, стоящую на месте ПОДВ X, и соответствующие состояния (в смысле программы Π_i, когда отбрасываются вспомогательные метки из Π_0). Далее выбираются те из отмеченных пар команд, где первая компонента (т.е. из Π_0) содержит ту команду СТОП в том состоянии (в смысле Π_0), что пара (команда, состояние) соответствует началу Π_i. Каждая из этих компонент, исходя из построенного выше дерева, задает определенный путь в программе Р. Заметим, что только из таких пар путей, соответствующих выбранным парам команд, получаются пути в Р, приводящие второй раз к команде ПОДВ X.

Далее образуются новые программы $\Pi_{k_1+1}, \dots, \Pi_{k_2}$ и начинается третий этап, где совершенно аналогичным образом строятся тройки согласованных путей в трех программах (опять с синхронизацией во всех трех путях). Этапы продолжаются до тех пор, пока в любой из совместно рассматриваемых множеств програми $\{\Pi_j\}$ некоторая программа Π_i появляется дважды. Тогда, как легко видеть, при втором появлении в программе Π_i можно реализовать только такие пути, которые можно было реализовать уже при первом появлении. После окончания всех этапов образуется одно общее обрезанное дерево развертки, в котором восстанавливаются команды ПОДВ и естественным образом из пар, троек и т.д. согласованных путей образуются пути в исходной программе Р.

Дальше алгоритм работает, как в [I].

Теорема доказана.

Зацикливание програми в системе команд K_4 с ПОДВ для одного массива выясняется аналогично.

Если же допустить команду ПОДВ для двух или более лент, то проблема построения ПСП сразу становится неразрешимой. Основная причина этого явления состоит в том, что второе считывание двух лент может происходить в совершенно другом порядке, чем первое считывание, и поэтому невозможна синхронизация, как при доказательстве теоремы 6.

ТЕОРЕМА 7. Не существует алгоритма, строящего ПСП для любой программы в системе команд K_4, в которой команда ПОДВ используется для двух лент (даже один раз).

Для доказательства этой теоремы рассмотрим класс двухленточных автоматов [5] над алфавитом $\Sigma=\{1,\dots,k\}$. Язык, представляемый данным автоматом А, т.е. множество пар слов из Σ^*, на которых А достигает конечного состояния, обозначим через L_A. Рабином и Скоттом доказано [5], что алгоритмически неразрешима следующая проблема: для данных двухленточных автоматов А и В определить, является ли пересечение $L_A \cap L_B$ непустым. Идея доказательства теоремы 7 состоит в том, что по двум автоматам А и В эффективно строится программа P(A,B) в системе команд K_4 с двумя лентами такая, что для P(A,B) существует пример, на котором она останавливается, тогда и только тогда, когда $L_A \cap L_B \neq \varnothing$. Нетрудно видеть, что для каждого двухленточного автомата A можно построить моделирующую его программу P_A с 2 лентами X и У в системе команд K_0 , которая остановится на тех и только тех парах массивов (X,У), где $(x,y) \in L_A$. Тогда P(A,B) имеет следующий вид:

$$\boxed{P_A} \to \boxed{\text{подв } X} \to \boxed{\text{подв } Y} \to \boxed{P_B} .$$

Ясно, что ПСП для P(A,B) непуста тогда и только тогда, когда непусто $L_A \cap L_B$.

ЛИТЕРАТУРА

1. Барздинь Я.М., Бичевский Я.Я., Калниньш А.А. Построение полной системы примеров для проверки корректности программ. - Настоящий сборник, стр.152-188.
2. Valiant L.G., Paterson M.S. Deterministic one - counter automata. - The University of Warwick Computer Centre. Report, 1973, No.6.
3. Мальцев А.И. Алгоритмы и рекурсивные функции, М., 1965.
4. Минский М. Вычисления и автоматы, М., 1971.
5. Рабин М.О., Скотт Д. Конечные автоматы и задачи их разрешения. - Кибернетический сборник, 4, М., 1962.

ЧАСТОТНЫЕ ВЫЧИСЛЕНИЯ С НЕБОЛЬШИМ ЧИСЛОМ ОШИБОК

Е.Б.Кинбер

Цель данной статьи заключается в исследовании частотных вычислений всюду определенных функций и частотного перечисления множеств или, по-другому, (m,n) -вычисления функций и (m,n) -перечисления множеств, когда число ошибок $n-m\leqslant 2$.

ОПРЕДЕЛЕНИЕ I. Пусть рекурсивный оператор T перерабатывает n -ку натуральных чисел $(x_1, x_2, \ldots, x_n)$ в n-ку натуральных чисел $(y_1, y_2, \ldots, y_n)$. Будем говорить, что функция $f(x)$ (m,n) -вычислима $(m\leqslant n)$ оператором T , если T каждой n -ке $(x_1, x_2, \ldots, x_n)$ где $x_i\neq x_j$, сопоставляет такую n - ку значений $(y_1, y_2, \ldots, y_n)$, что в системе равенств $f(x_1)=y_1$, $f(x_2)=y_2, \ldots,$ $f(x_n)=y_n$ по крайней мере m истинны.

Эта концепция частотного вычисления изучалась в [I] (см.также [2]) и в [4]. В [I] было доказано, в частности, что при $\frac{m}{n}>\frac{1}{2}$ любая (m,n) -вычислимая функция общерекурсивна. Однако, как показано в [4], не существует эффективной процедуры, посредством которой из рекурсивного оператора, (m,n) -вычисляющего $(m>\frac{n}{2})$ некоторую совокупность функций, можно было бы извлекать алгоритм для обычного вычисления одной из них. Отсюда возникает естественный вопрос: можно ли по крайней мере эффективно повышать частоту (m,n) -вычисления. Теорема 3 [4] дает положительный ответ: существует алгоритм, который для любого $k>0$ произвольному рекурсивному оператору T, (m,n) -вычисляющему некоторую совокупность функций, сопоставляет оператор T^1, $(2m+k, 2n+k)$ - вычисляющий по крайней мере ту же совокупность. Число ошибочных ответов, которые на произвольной n -ке может выдать такой оператор T^1, равно $2(n-m)>n-m$. Следующий пример показывает, что при повышении частоты число ошибок $n-m$ не может быть сохранено.

Пусть $V_{m,n}(T)$ - совокупность функций, (m,n) - вычислимых оператором T.

ТЕОРЕМА I. Существует такой рекурсивный оператор T от пяти переменных, что $V_{3,5}(T) \nsubseteq V_{4,6}(T')$ для любого рекурсивного оператора T' от шести переменных.

ДОКАЗАТЕЛЬСТВО. Значения нужного нам оператора на произвольной пятерке определяются следующим образом: если в пятерке отсутствует число 0, то T ставит 1 против 1 и 2 (если такие есть в пятерке) и 0 - против всех остальных чисел; если в пятерке присутствуют $0,1,2$ и 3, то 1 ставится против 0 и 1, а 0 - против всех остальных чисел. В оставшихся случаях T ставит 1 против 0 и 0 - против всех остальных чисел.

Обозначим через 0^∞ бесконечную последовательность нулей. Прямой проверкой нетрудно убедиться в том, что в $V_{3,5}(T)$ содержатся следующие предикаты: 0^∞, 1110^∞, 1010010^∞, 110010^∞, 10^∞ (а также некоторые другие); здесь i-й член последовательности соответствует числу $i-1$

Для любого набора - кандидата на роль $T^1(0,1,2,3,4,5)$ можно подобрать один из указанных выше предикатов, набор значений которого на первых шести числах отличается от $T^1(0,1,2,3,4,5)$ более чем в двух позициях, что и доказывает справедливость теоремы.

При $n-m=1$ можно, однако, эффективно повышать частоту, сохраняя число ошибок неизменным.

ТЕОРЕМА 2. Существует алгоритм, который для любого $n \geqslant 3$ произвольному рекурсивному оператору T от n переменных сопоставляет такой рекурсивный оператор T^1, что

$$V_{n-1,n}(T) \subseteq V_{n,n+1}(T^1).$$

ДОКАЗАТЕЛЬСТВО. Нам необходимо определить оператор T^1 на произвольном наборе чисел $(x_1, x_2, \dots, x_{n+1})$, где $x_i \neq x_j$, $i \neq j$. Для этой цели мы находим всевозможные наборы чисел $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_{n+1})$, которые удовлетворяют следующим двум условиям:

1) для любого i, где $1 \leqslant i \leqslant n+1$, набор $(\alpha_1, \alpha_2, \dots, \alpha_{i-1}, \alpha_{i+1}, \dots, \alpha_{n+1})$ отличается от $T(x_1, x_2, \dots, x_{i-1}, x_{i+1}, \dots, x_{n+1})$ не более чем в одной позиции;

2) в $(\alpha_1, \alpha_2, \dots, \alpha_{n+1})$ содержатся только те числа, которые имеются, по крайней мере, в одном из наборов $T(x_1, x_2, \dots, x_{i-1}, x_{i+1}, \dots, x_{n+1})$ $(1 \leqslant i \leqslant n+1)$.

Указанных наборов α имеется, очевидно, конечное число.

Легко убедиться, что все наборы α отличаются не более чем в двух позициях. Если все α отличаются не более чем в одной позиции, то в качестве значения $T^1(x_1, x_2, \dots, x_{n+1})$ можно взять любой из них. Предположим, что существуют наборы α^1 и α^2, отличающиеся в двух позициях. Пусть, например, $\alpha_1^1 \neq \alpha_1^2$ и $\alpha_2^1 \neq \alpha_2^2$. В качестве значения T^1 на $(x_1, x_2, \dots, x_{n+1})$ мы берем набор $(\beta_1, \beta_2, \alpha_3^1, \alpha_4^1, \dots, \alpha_{n+1}^1)$, где β_1 и β_2 определяются следующим образом. Если существует набор α^3, отличный от α^1 и α^2, то $\beta_1 = \alpha_1^3$ и $\beta_2 = \alpha_2^3$; в противном случае $\alpha_1^1 = \beta_1$ и $\alpha_2^2 = \beta_2$. Проверка того, что T^1 удовлетворяет условию теоремы, не составляет большого труда. Теорема доказана.

В работе автора [4] определение (m, n)-вычисления естественным образом обобщается на частичные функции. Для этого в определении I вместо рекурсивного оператора T нужно взять частично-рекурсивный (ч.р.) оператор, в произвольной n-ке значений которого $(y_1, \dots, y_n)$ некоторые y_i могут быть не определены. Такой ч.р. оператор можно представить, например, в виде n-ки частично-рекурсивных функций (ч.р.ф.) $(\varphi_1, \varphi_2, \dots, \varphi_n)$ от n переменных. В определении I мы теперь считаем, что $f(x_i) = y_i$

и в том случае, когда $f(x_i)$ и y_i одновременно не определены; если же определено только y_i или $f(x_i)$, то считается, что $f(x_i) \neq y_i$.

ОПРЕДЕЛЕНИЕ 2. Будем говорить, что множество A (m,n)-перечислимо ч.р. оператором T, если A является областью определения функции f, (m,n)-вычислимой оператором T.

Класс множеств, (m,n)-перечислимых оператором T, обозначим через $\mathfrak{A}_{m,n}(T)$. Тогда для любого множества A пусть:

$$A \in \mathfrak{A}_{m,n} \Longleftrightarrow \exists \text{ ч.р. оператор } T\,(A \in \mathfrak{A}_{m,n}(T))$$

Автор показал *), что при $m<n$ класс $\mathfrak{A}_{m,n}$ содержит множества, не являющиеся рекурсивно перечислимыми; далее в классе всех частотно-перечислимых множеств была установлена иерархия по числу ошибок $n-m$: если $n_2 - m_2 > n_1 - m_1$, то существует множество $A \in \mathfrak{A}_{m_2,n_2} \setminus \mathfrak{A}_{m_1,n_1}$. Оставался нерешенным вопрос о том, могут ли отличаться классы $\mathfrak{A}_{m_1,n_1}$, $\mathfrak{A}_{m_2,n_2}$, если число ошибок $n_2 - m_2 = n_1 - m_1$, но частота различна: например, $\frac{m_1}{n_1} > \frac{m_2}{n_2}$. Следующая теорема дает положительный ответ.

ТЕОРЕМА 3. Существует множество $A \in \mathfrak{A}_{3,5} \setminus \mathfrak{A}_{4,6}$

Доказательство теоремы основано на приоритетной процедуре, "опровергающей" главную нумерацию всех ч.р. операторов от шести переменных.

В случае $(n-1,n)$-вычислений ситуация меняется.

ТЕОРЕМА 4. Для любого $n \geq 3$ $\mathfrak{A}_{n-1,n} = \mathfrak{A}_{n,n+1}$.

ДОКАЗАТЕЛЬСТВО. Пусть $A \in \mathfrak{A}_{n-1,n}(T)$ и T - ч.р. оператор. Нужно показать, что $A \in \mathfrak{A}_{n,n+1}$.

---

*) Публикация этих результатов предусмотрена в виде отдельной статьи.

Вычисление ч.р. оператора T можно представить, очевидно, следующим образом. Машина Тьюринга $\mathfrak{M}$, вычисляющая T на произвольной n-ке $(x_1, x_2, \ldots, x_n)$, работает бесконечно долго и, время от времени, на специальной выходной ленте печатает единицу против какого-либо x_i; начиная с некоторого момента, новые единицы в n-ке значений перестают появляться. Через $T^{(t)}(x_1, x_2, \ldots, x_n)$ мы обозначим n-ку значений, которую машина $\mathfrak{M}$ выдает, проработав на $(x_1, x_2, \ldots, x_n)$ t тактов; в $T^{(t)}(x_1, x_2, \ldots, x_n)$ тем x_i, против которых значение еще не выдано, соответствует символ H (означающий "не определено").

Произвольный набор единиц и символов H $\alpha_t = (\alpha_t(0), \alpha_t(1), \ldots, \alpha_t(V))$ назовем V - допустимым в момент t, если любая n-ка значений $T^{(t)}(y_1, y_2, \ldots, y_n)$, где $\max\limits_{1 \leq \tau \leq n} \{y_\tau\}$ и $y_i \neq y_j$, $i \neq j$, отличается от набора $(\alpha_t(y_1), \alpha_t(y_2), \ldots, \alpha_t(y_n))$ не более чем в одной позиции. Пусть χ - частично-характеристическая функция множества A: $(\chi(x) = 1 \Longleftrightarrow x \in A)$.

Нетрудно показать, что если в $\mathfrak{A}_{n-1,n}(T)$ кроме A содержатся и другие множества, то все они (в том числе и A) рекурсивно перечислимы и, тем более, содержатся в $\mathfrak{A}_{n,n+1}$. Поэтому мы будем предполагать, что $\mathfrak{A}_{n-1,n}(T) = \{A\}$. Зафиксируем произвольный набор различных чисел $(x_1, x_2, \ldots, x_{n+1})$.

ЛЕММА. Существуют такие числа $z \geq \max\limits_{1 \leq \tau \leq n+1} \{x_\tau\}$ и t_0, что, как только $V \geq z$ и $t \geq t_0$, для любого V-допустимого в момент t набора $\alpha_t = (\alpha_t(0), \alpha_t(1), \ldots, \alpha_t(V))$ выполняется система равенств

$$\chi(x_i) = \alpha_t(x_i) \quad (1 \leq i \leq n+1).$$

ДОКАЗАТЕЛЬСТВО напоминает доказательство теоремы 3 из [I], поэтому мы лишь коротко наметим его основные этапы. Каждой n-ке значений $(\beta_1, \beta_2, \ldots, \beta_n) = T(y_1, y_2, \ldots, y_n)$ можно сопоставить формулу логики суждений, утверждающую, что в системе $\chi(y_i)=\beta_i$ $(1\leqslant i\leqslant n)$ по крайней мере, $n-1$ равенств истинны. Получается выполнимая система формул, причем каждой ее интерпретации соответствует функция, $(n-1,n)$-вычислимая оператором T, и наоборот. Так как $T(n-1,n)$ - вычисляет только одну функцию, то система формул имеет единственную интерпретацию. Чтобы найти $\chi(x)$, мы просматриваем систему формул до того момента, когда обнаруживается, что в качестве $\chi(x)$ можно взять только одно значение. Для установления значения $\chi(x)$ используется лишь конечное число формул, что и отражает число z в формулировке леммы. Заметим, что сопоставление формул n-кам $T(y_1,y_2,\dots,y_n)$ осуществляется, вообще говоря, неэффективно, и поэтому число z также не может быть найдено эффективно. Возможность выбора нужного t_0 в лемме становится ясной, если заметить, что для любого набора $(y_1,y_2,\dots,y_n)$ существует такой t, что $T(y_1,y_2,\dots,y_n)=T^{(t)}(y_1,y_2,\dots,y_n)$. Лемма доказана.

Допустим теперь, что в наборе $(x_1,x_2,\dots,x_{n+1})$ имеется пара различных чисел (x_r,x_j), удовлетворяющая следующему условию:

(I) если в n-ке различных чисел $(y_1,y_2,\dots,y_n)$ $y_1=x_r$ и $y_2=x_j$ и если $T(y_1,y_2,\dots,y_n)=(\beta_1,\beta_2,\dots,\beta_n)$, то $\chi(x_r)\neq\beta_1$ или $\chi(x_j)\neq\beta_2$.

По определению $(n-1,n)$-перечислимости в этом случае равенство $\chi(y_3)=\beta_3$ должно выполняться для любого $y_3\notin\{x_r,x_j\}$. Отсюда следует, что A рекурсивно перечислимо. Поэтому мы будем предполагать, что (I) не выполняется ни для одной пары (x_r,x_j). Из этого предположения, леммы и непустоты $\mathfrak{A}_{n-1,n}(T)$ вытекает существование таких бесконечных рекурсивных последовательностей $t_1<t_2<\dots<t_r<t_{r+1}<\dots$ и $\max\limits_{1\leqslant i\leqslant n+1}\{x_i\}\leqslant z_1\leqslant\dots\leqslant z_r\leqslant z_{r+1}\leqslant\dots$, что для любой пары (t_r,z_r) выполняются следующие условия.

(а) существует z_r-допустимый в момент t_r набор $\alpha_{t_r}=(\alpha_{t_r}(0),\alpha_{t_r}(1),\dots,\alpha_{t_r}(z_r))$;

(б) для любых двух z_t -допустимых в момент t_t наборов $\alpha^1_{t_t}$ и $\alpha^2_{t_t}$ выполняется система равенств:

$$\alpha^1_{t_t}(x_l)=\alpha^2_{t_t}(x_l) \quad (1\leqslant l\leqslant n+1);$$

обозначим эти числа через $\alpha_{t_t}(x_l)$ $(1\leqslant l\leqslant n+1)$;

(в) для любой пары (x_p, x_j), где $p\neq j$, существует такая n -ка попарно различных чисел $(y_1, y_2, \ldots, y_n)$, что $x_p=y_1, x_j=y_2, \max\limits_{1\leqslant i\leqslant n}\{y_i\}\leqslant z_t$ и если $(\beta_1,\beta_2,\ldots,\beta_n)=T^{(t_t)}(y_1,y_2,\ldots,y_n)$, то $\alpha_{t_t}(x_p)=\beta_1$ и $\alpha_{t_t}(x_j)=\beta_2$.

Определим по индукции последовательность операторов $P_1, P_2, \ldots, P_k, \ldots$:
$P_1(x_1,x_2,\ldots,x_{n+1})=(\alpha_{t_1}(x_1),\alpha_{t_1}(x_2),\ldots,\alpha_{t_1}(x_{n+1}))$; пусть $P_k(x_1,x_2,\ldots,x_{n+1})=(\gamma_1,\gamma_2,\ldots,\gamma_{n+1})$, мы полагаем $P_{k+1}(x_1,x_2,\ldots,x_{n+1})=(\sigma_1,\sigma_2,\ldots,\sigma_{n+1})$, где для всех $l\ (1\leqslant l\leqslant n+1)$:

$$\sigma_l=\begin{cases}1, & \text{если } \gamma_l=1\\ \alpha_{t_{k+1}}(x_l), & \text{если } \gamma_l\neq 1\end{cases}$$

Пусть $T^1(x_1,x_2,\ldots,x_{n+1})=\lim\limits_k P_k(x_1,x_2,\ldots,x_{n+1})$. Как легко видеть, T^1 - корректно определенный ч.р. оператор. Используя выполнение для любой пары (x_p, x_j) условия (в), нетрудно показать, что $A\in\mathfrak{A}_{n,n+1}(T^1)$, теорема доказана.

Доказательство теоремы I не дает нам процедуры, позволяющей осуществлять эффективный переход от $(n-1,n)$-перечисления множеств к их $(n,n+1)$ -перечислению. Следующая теорема показывает, что такой процедуры, в отличие от (m,n) -вычисления всюду определенных функций, не существует.

Зафиксируем $n\geqslant 3$ и главные нумерации всех ч.р. операторов от n переменных $\{T_i\}$ и от $n+1$ переменных $\{P_i\}$. Пусть D_f - область определения функции f и $B=\{i\mid \mathfrak{A}_{n-1,n}(T_i)\neq\varnothing\}$.

ТЕОРЕМА 5. Для любой ч.р.ф. g если $B\subseteq D_g$, то существует такое число a , что $\mathfrak{A}_{n-1,n}(T_a) \not\subseteq \mathfrak{A}_{n,n+1}(P_{g(a)})$.

Коротко наметим основные этапы доказательства. Первый этап заключается в построении общерекурсивной функции (о.р.ф.) $u(\tau)$ такой, что $\forall\tau(u(\tau)\in B)$ и

$$\forall j\,[(T_j \in \{T_{u(\tau)}\}_{\tau=0,1,\ldots}) \Longrightarrow (T_{u(j)} \equiv T_j)]^{*)}.$$

Каждый класс $\mathfrak{A}_{n-1,n}(T_{u(\tau)})$ состоит из нескольких конечных множеств, входящих в $\{0,1,\ldots,n\}$. Предположим, что теорема неверна. Тогда существует такая ч.р.ф. g , что $B\subseteq D_g$ и $(\mathfrak{A}_{n-1,n}(T_\tau)\neq\varnothing)\Longrightarrow(\mathfrak{A}_{n-1,n}(T_\tau)\subseteq\mathfrak{A}_{n,n+1}(P_{g(\tau)}))$ для любого τ . Используя функцию $g(u(\tau))$, которая, очевидно, о.р.ф., мы можем построить такую о.р.ф. $f(\tau)$, что $\forall\tau\,(T_{f(\tau)}\in\{T_{u(i)}\}_{i=0,1,\ldots})$, но при фиксированном τ в $\mathfrak{A}_{n,n+1}(P_{g(u(\tau))})$ не могут содержаться все множества из $\mathfrak{A}_{n-1,n}(T_{f(\tau)})$. Применяя к $f(\tau)$ теорему о неподвижной точке, мы придем к противоречию.

Автор выражает глубокую признательность Р.В.Фрейвалду за внимание к работе.

ЛИТЕРАТУРА

1. Трахтенброт Б.А. О частотном вычислении функций.- Алгебра и логика", 1963, 2, № 1.
2. Трахтенброт Б.А. Частотные вычисления.- Труды Мат.ин-та им.В.А.Стеклова АН СССР, 1973, 113.
3. Ершов Ю.А. Теория нумераций I, Новосибирск, 1969.
4. Кинбер Е.Б. Частотные вычисления общерекурсивных предикатов и частотное перечисление множеств.- "ДАН СССР", 1972, 205, № 1.

---

*) Это означает, что нумерованное множество $\{T_{u(\tau)}\}_{\tau=0,1,\ldots}$ является n-подобъектом $\{T_\tau\}_{\tau=0,1,\ldots}$ с $u(\tau)$ в качестве функции, осуществляющей морфизм (см.[3]).

СВОДИМОСТЬ КЛАССОВ ФУНКЦИЙ ПО ПРОГНОЗИРОВАНИЮ

А.В.Анджан

Через $\mathcal{G}$ обозначается класс всех всюду определенных функций, через $\mathcal{G}R$ – класс всех общерекурсивных функций. Если $f\in\mathcal{G}$, то под $\tilde{f}(n)$ понимаем номер кортежа $<f(0),\ldots,f(n)>$ в фиксированной эффективной нумерации всех конечных кортежей натуральных чисел. Если $f,g\in\mathcal{G}$, то под $f \operatorname{join} g$ понимаем функцию $h\in\mathcal{G}$, такую, что

$$\forall x \qquad h(x)=\begin{cases} f(t), & \text{если } x=2t \\ g(t), & \text{если } x=2t+1. \end{cases}$$

Под 0^∞ понимаем функцию, тождественно равную нулю.

Рассматриваются две постановки прогнозирования: прогнозирование классов функций из $\mathcal{G}$ в смысле следующего значения и прогнозирование классов функций из $\mathcal{G}R$ в смысле геделевского номера общерекурсивными стратегиями, которые отличаются от постановок А и Б в [I] лишь тем, что прогнозирование ведется только на натуральной выборке $\Omega=\{0,1,\ldots\}$. (В некоторых статьях настоящего сборника прогнозирование в смысле геделевского номера называется предельным синтезом).

Хорошо известно, что в любом из этих смыслов существуют как прогнозируемые, так и непрогнозируемые классы. Это наводит на мысль релятивизировать понятие прогнозируемости подобно тому, как сводимость по Тьюрингу релятивизирует понятие разрешимости множества.

Если $B\subset\mathcal{G}$, то через O_B^{NV} обозначим класс $\{f\mid f\in\mathcal{G}\ \&\ \forall g\in B\ \exists n_0,\ n>n_0\Longrightarrow f(\tilde{g}(n))=g(n+1)\}$.
Если $B\subset\mathcal{G}R$, то через O_B^{GN} обозначим класс $\{f\mid f\in\mathcal{G}\ \&\ \forall g\in B\ \lim_{n\to\infty} f(\tilde{g}(n))$ существует и равен какому-нибудь геделевскому номеру $g\}$.

Содержательно, O_B^{NV} (соответственно O_B^{GN}) состоит из всех всюду определенных, вообще говоря, неэффективных стратегий, прогнозирующих B в смысле следующего значения (соответственно, в смысле геделевского номера).

ОПРЕДЕЛЕНИЕ I. Если $A,B \subset \mathcal{G}$, то $A \leqslant_{NV} B$ (A сводится к B в смысле прогнозирования следующего значения), если существует машина Тьюринга с оракулом "функция", которая: а) при любой функции $h \in \mathcal{G}$ в оракуле и при любой начальной ленте $\boxed{\Pi}$ выдает результат;

б) при любой функции $h \in O_B^{NV}$ в оракуле реализует функцию из O_A^{NV}.

ОПРЕДЕЛЕНИЕ 2. Заменив в предыдущем определении $A,B \subset \mathcal{G}$ на $A,B \subset \mathcal{G}R$, а O_A^{NV} и O_B^{NV} на O_A^{GN} и O_B^{GN}, получаем определение сводимости A к B в смысле прогнозирования геделевского номера ($A \leqslant_{GN} B$).

Содержательно, $A \leqslant_{GN} B$, если мы умеем прогнозировать A в смысле геделевского номера при условии, что, подавая некоторому оракулу начальные куски любой функции f из B, мы получаем последовательность ответов, стабилизирующуюся на геделевском номере функции f.

Аналогично объясняется смысл $A \leqslant_{NV} B$.

Как NV -сводимость, так и GN -сводимость рефлексивны и транзитивны. Следовательно, классы NV -эквивалентности классов функций из $\mathcal{G}$, классы NV -эквивалентности классов функций из $\mathcal{G}R$, классы GN -эквивалентности классов функций из $\mathcal{G}R$ образуют частичные упорядочения, обозначаемые через $\mathfrak{A}_{NV}$, $\widetilde{\mathfrak{A}}_{NV}$ и $\mathfrak{A}_{GN}$ соответственно. Сами классы эквивалентности класса A обозначаются через $d_{NV}(A)$, $\widetilde{d}_{NV}(A)$ и $d_{GN}(A)$ соответственно.

ПРЕДЛОЖЕНИЕ I. $\mathfrak{A}_{NV}, \widetilde{\mathfrak{A}}_{NV}$ и $\mathfrak{A}_{GN}$ – верхние полурешетки. Если A - пустой класс, то нижней верхней гранью в $\mathfrak{A}_{NV}$ ($\widetilde{\mathfrak{A}}_{NV}$, $\mathfrak{A}_{GN}$) для $d_{NV}(A)$ и $d_{NV}(B)$ ($\widetilde{d}_{NV}(A)$ и $\widetilde{d}_{NV}(B)$, $d_{GN}(A)$ и $d_{GN}(B)$) служит $d_{NV}(B)$ ($\widetilde{d}_{NV}(B)$, $d_{GN}(B)$). Если же ни A, ни B пустыми не являются, то этой гранью служит $d_{NV}(C)$ ($\widetilde{d}_{NV}(C)$, $d_{GN}(C)$), где $C =$
$= \{ f \text{ join } g \mid f \in A \;\&\; g \in B \} \cup \{ f \text{ join } O^{\infty} \mid f \in A \} \cup$

$\cup\{0^{\infty} j o i n g \mid g \in B\}$

ПРЕДЛОЖЕНИЕ 2. В частичных упорядочениях $\mathfrak{A}_{NV}$, $\tilde{\mathfrak{A}}_{NV}$ и $\mathfrak{A}_{GH}$ существуют наименьшие степени, состоящие в точности из прогнозируемых классов, и наибольшие степени, равные $d_{NV}(\mathcal{G})$, $\tilde{d}_{NV}(\mathcal{GR})$ и $d_{GH}(\mathcal{GR})$ соответственно.

ПРЕДЛОЖЕНИЕ 3. В частичном упорядочении $\mathfrak{A}_{NV}$ существует степень промежуточной сложности $d_{NV}(\mathcal{GR})$ (По отношению к $\tilde{\mathfrak{A}}_{NV}$ и $\mathfrak{A}_{GH}$ этот вопрос не решен).

Следующая теорема показывает, что между NV -сводимостью и GH -сводимостью существуют большие различия.

ТЕОРЕМА. Существует класс $\bar{\sigma} \subset \mathcal{GR}$, принадлежащий наименьшей степени полурешетки $\mathfrak{A}_{GH}$ и наибольшей степени полурешетки $\tilde{\mathfrak{A}}_{NV}$.

Класс $\bar{\sigma}$ определяется так: $\bar{\sigma} = \{ f' \mid f \in \mathcal{GR} \}$, где $\forall t \ f'(t) = c(n, f(t))$ и n - фиксированный геделевский номер функции f. (Этот класс впервые рассмотрен Я.М. Барзинем).

Из нерешенных вопросов наиболее интересен вопрос Р.В.Фрейвалда: существует ли наибольшая NV -степень в полурешетке $\mathfrak{A}_{NV}$ среди таких степеней, для которых классы А, входящие в них, имеют непустые классы σ_A^{NV} ?

В заключение выражаю сердечную благодарность своему руководителю Р.В.Фрейвалду за постановку проблем и ценные замечания в ходе работы.

ЛИТЕРАТУРА

I. Барздинь Я.М., Фрейвалд Р.В. О прогнозировании общерекурсивных функций. - "ДАН СССР", 1972, 206, № 3.

ГОЛОГРАФНЫЕ МНОЖЕСТВА

Ю.В.Борзов

Голограммы имеют необычное свойство - отделенный фрагмент воспроизводит в точности то же, что целая пластина. Р.В.Фрейвалд предложил исследовать класс множеств натуральных чисел с аналогичными свойствами. В данной работе вводится понятие голографности, исследуется ее связь с внутрисводимостью [I] и автосводимостью [2]. Названные понятия формализуют интуитивное понятие взаимной зависимости (взаимной независимости) индивидуальных задач массовой проблемы.

Характеристическую функцию множества A обозначим - $f_A(x)$. Назовем секвенцией частично определенное отображение из N в $\{0,1\}$. Если A - множество, $\{e\}^A(x)$ является e-той функцией, частичнорекурсивной в A, W_e^A - область определения функции $\{e\}^A(x)$. Если $\{e\}^A(x)=y$, это равенство следует из конечного числа значений $f_A(n)$. Как известно, функцию $\{e\}^A(x)$ вычисляет машина Тьюринга с оракулом, к которому можно обратиться с вопросом $n \in A$. Оракул выдает ответ 1, если $n \in A$, и 0, если $n \bar{\in} A$. Если σ - секвенция, A - множество и для каждого n из области определения σ $\sigma(n)=f_A(n)$, $\{e\}^\sigma(x)$ - e-тая функция, частичнорекурсивная в σ, т.е. функцию вычисляет машина Тьюринга с оракулом, к которому можно обратиться с вопросом $n \in A$; оракул выдает 1, если $\sigma(n)=1$, ответ 0, если $\sigma(n)=0$, и 2 (пустой ответ), если $\sigma(n)$ не определена. W_e^σ является областью определения функции $\{e\}^\sigma(x)$. Если $\{e\}^\sigma(x)=y$, равенство следует из конечного числа значений $\sigma(n)$. В случае всюдуопределенности секвенции σ A - частичнорекурсивность (A - рекурсивность) в точности совпадает с σ - частичнорекурсивностью (σ - рекурсивностью).

$\{e\}^\sigma = A$ тогда и только тогда, если для каждого

$x \in N$ $\{e\}^{\sigma}(x) = f_A(x)$. $W_e^{\sigma} = A$ тогда и только тогда, если для каждого $x \in A$ $\{e\}^{\sigma}(x) = f_A(x)$.

ОПРЕДЕЛЕНИЕ I. Множество A является голографным, если для каждой секвенции σ, удовлетворяющей условиям:

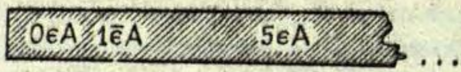
а) существует бесконечно много n, для которых $\sigma(n) = 1$;

б) существует бесконечно много n, для которых $\sigma(n) = 0$;

в) для каждого n из области определения σ $\sigma(n) = f_A(n)$; - существует номер e такой, что $\{e\}^{\sigma} = A$.

ОПРЕДЕЛЕНИЕ 2. Множество A является равномерно голографным, если существует номер e такой, что $\{e\}^{\sigma} = A$ для каждой секвенции σ, удовлетворяющей условиям а), б), в).

Поясним определения на следующей интерпретации. Дано множество A и бесконечная лента, в каждой ячейке которой записано натуральное число и обозначена его принадлежность A. Числа на ленте расположены в возрастающем порядке. Эта лента покрыта черной лентой с прорезями, позволяющими увидеть некоторые числа на нижней ленте. При этом о числах под черной лентой ничего не известно. Например,



Таким образом, черная лента с прорезями является оракулом, который дает частичную информацию о вхождении некоторых чисел в данное множество A.

Определение I можно перефразировать так: множество A является голографным, если для каждой черной ленты с бесконечным числом прорезей с числами, принадлежащими A, и бесконечным числом прорезей с числами, не принадлежащими A, существует алгоритм, разрешающий проблему вхождения для любого наперед заданного натурального числа.

Определение 2 требует существования одного алгоритма для всех таких черных лент.

ОПРЕДЕЛЕНИЕ 3. (К.Г.Джокуш, [I]). Внутрисводимые и равномерно внутрисводимые множества определяются соответственно определениями I и 2, если отбросить условие б).

В нашей интерпретации проблема вхождения для этих множеств разрешима при помощи черных лент с бесконечным числом прорезей с числами, принадлежащими A .

Прямо из определений следует, что каждое (равномерно) внутрисводимое множество и каждое (равномерно) ко-внутрисводимое множество является (равномерно) голографным. Нетрудно показать также, что имеется континуум голографных множеств (впрочем, как и неголографных).

ОПРЕДЕЛЕНИЕ 4. (Б.А.Трахтенброт, [2]). Множество A является автосводимым, если существует номер e такой, что $\{e\}^{\sigma}=A$, где для каждого $x_0 \in N$ при вычислении $\{e\}^{\sigma}(x_0)$ секвенция σ не определена лишь на x_0.

В нашей интерпретации - черная лента закрывает лишь только то натуральное число, чье вхождение в A в данный момент выясняется.

ПРЕДЛОЖЕНИЕ I. Существует континуум автосводимых множеств, которые не голографны.

ТЕОРЕМА I. Существует голографное, но не равномерно голографное множество.

ПРЕДЛОЖЕНИЕ 2. Существует неголографное рекурсивно перечислимое множество.

ПРЕДЛОЖЕНИЕ 3. Если множество A (равномерно) голографно, то и $\bar{A}$ (равномерно) голографно.

ПРЕДЛОЖЕНИЕ 4. Существуют равномерно голографные множества A_i и B_i , что не голографны $A_1 \cup B_1, A_2 \cap B_2, A_3 \setminus B_3, A_4 \text{ join } B_4, (A_5 \cap \bar{B}_5) \cup (\bar{A}_5 \cap B_5)$.

ПРЕДЛОЖЕНИЕ 5. Существуют эквивалентные по Тьюрингу равномерно голографные множества A и B такие, что $A \cup B$

не равномерно голографно.

ОПРЕДЕЛЕНИЕ 5. Множество A является голографно перечислимым, если для каждой секвенции σ, удовлетворяющей условиям а), б), в), существует номер e такой, что $W_e^{\sigma}=A$. Множество назовем равномерно голографно перечислимым, если номер e существует для всех таких σ.

ТЕОРЕМА 2. Если множество A равномерно голографно перечислимо и внутрисводимо, то оно равномерно голографно. При этом не верно, что оно всегда равномерно внутрисводимо.

ТЕОРЕМА 3. Каждое голографное множество автосводимо.

ПРЕДЛОЖЕНИЕ 6. Если A и $\bar{A}$ (равномерно) голографно перечислимы, то A - (равномерно) голографно.

ТЕОРЕМА 4. Существует равномерно голографное множество, не являющееся ни внутрисводимым, ни ко-внутрисводимым.

Таким образом установлено, что голографные множества являются естественным промежуточным звеном между внутрисводимыми и автосводимыми множествами.

Следует отметить, что если рассматривать бесконечные секвенции σ, но условий а) и б) не налагать, в равномерном случае мы получим в точности класс рекурсивных множеств.

ЛИТЕРАТУРА

I. Jockusch G.G. Uniformly introreducible sets - "Journal of Symbolic Logic", 1968, 33, No.4.

2. Трахтенброт Б.А. Об автосводимости - "ДАН СССР", 1970, 192, № 6.

О ПРЕДЕЛЬНОМ СИНТЕЗЕ ПОЧТИ МИНИМАЛЬНЫХ ГЕДЕЛЕВСКИХ НОМЕРОВ

Е.Б.Кинбер

Определения и обозначения взяты из [I].

В [I] было введено понятие предельного синтеза в смысле GN. Рассмотрим близкую задачу - предельный синтез почти минимальных геделевских номеров.

Зафиксируем произвольную главную нумерацию всех ч.р.ф. от одной переменной и через $\min(\varphi)$ обозначим минимальный геделевский номер функции φ.

ОПРЕДЕЛЕНИЕ. Будем говорить, что класс о.р.ф. V предельно синтезируем в смысле $\widetilde{GN}$, если существуют о.р.ф. $h(x)$ и о.р.стратегия F такие, что

I) F предельно синтезирует V в смысле GN

2) для любой о.р.ф. $\varphi \in V$

$$\lim_n F(<\varphi(0)\ldots\varphi(n)>) \leqslant h(\min(\varphi)).$$

Я.М.Барздинем поставлен следующий вопрос: существуют ли классы функций, которые принадлежат $GN \setminus \widetilde{GN}$. Мы покажем, что такой класс имеется уже среди эффективно перечислимых классов о.р.ф.

Пусть $\mathcal{U}$ - класс всех о.р.ф., принимающих значения из $\{0,1\}$ и равных нулю во всех точках за исключением, быть может, конечного числа. Класс $\mathcal{U}$, очевидно, эффективно перечислим и поэтому, согласно [2], предельно синтезируем в смысле GN.

ТЕОРЕМА I. $\mathcal{U} \notin \widetilde{GN}$.

ДОКАЗАТЕЛЬСТВО. Допустим, от противного, что для класса $\mathcal{U}$ и некоторых о.р. стратегии F и о.р.ф. $h(x)$ выполняются условия I) и 2).

Определим для каждого натурального i о.р.ф. $\varphi_{g(i)}$ в соответствии со следующими инструкциями.

Шаг I. Будем последовательно находить гипотезы $F(<0>), F(<00>), \ldots, F(<0^k>), \ldots$. Если оказывается, что $F(<0^{\tau}>)=n \notin \{F(<0>), \ldots, F(<0^{\tau-1}>)$ и $n \leq h(i)$, то начинаем параллельно вычислять φ_n на числах $0, 1, \ldots, \tau$, проделывая один такт вычислений одновременно с нахождением очередной гипотезы $F(<0^k>)$.

Так как $0^{\infty} \in \mathcal{U}$ и F предельно синтезирует класс $\mathcal{U}$, то когда-нибудь должно наступить одно из следующих двух событий:

A^1. найдется такое k , что $F(<0^k>) > h(i)$.

B^1. для некоторого $n = F(<0^{\tau}>) \leq h(i)$ через определенное число шагов вычисления $\varphi_n(0), \varphi_n(1), \ldots, \varphi_n(\tau)$ заканчиваются, и при этом $\varphi_n(0) = \varphi_n(1) = \ldots = \varphi_n(\tau) = 0$.

Как только впервые одно из событий A^1 или B^1 наступит, мы следующим образом определим некоторый начальный фрагмент функции $\varphi_{g(i)}$:

а) если наступит событие A^1 , то для всех $x < k$ полагаем $\varphi_{g(i)}(x) = 0$.

б) если наступит событие B^1 , то для всех $x \leq \tau$ полагаем $\varphi_{g(i)}(x) = 1$ (таким образом, $F(<0^{\tau}>)$ не может быть номером определяемой нами функции $\varphi_{g(i)}$).

Переходим к шагу 2.

Шаг m. Пусть $\varphi_{g(i)}$ определена к этому моменту на всех $x \leq s$ и $\bar{\alpha} = (\varphi_{g(i)}(0), \varphi_{g(i)}(1), \ldots, \varphi_{g(i)}(s))$.

Будем последовательно находить $F(<\bar{\alpha}0>), F(<\bar{\alpha}00>), \ldots, F(<\bar{\alpha}0^k>), \ldots$. Если оказывается, что $n = F(<\bar{\alpha}0^{\tau}>) \notin \{F(<\bar{\alpha}0>), \ldots, F(<\bar{\alpha}0^{\tau-1}>)$ и $n \leq h(i)$, то начинаем параллельный процесс вычисления φ_n на всех $x \leq s+\tau$. Так как $\bar{\alpha}0^{\infty} \in \mathcal{U}$, то когда-нибудь должно наступить одно из следующих событий:

A^m. найдется такое k , что $F(<\bar{\alpha}0^k>) > h(i)$.

B^m. для некоторого $n = F(<\bar{\alpha}0^{\tau}>) \leq h(i)$ вычисления φ_n на всех $x \leq s+\tau$ заканчиваются, причем $(\varphi_n(0), \ldots, \varphi_n(s)) = \bar{\alpha}$ и для всех x таких, что $s < x \leq s+\tau$, $\varphi_n(x) = 0$.

Если раньше наступает событие A^m, то на всех $x \in \{s+1, \dots, s+k\}$ полагаем $\varphi_{g(i)}(x)=0$; в противном случае на всех $x \in \{s+1, \dots, s+z\}$ полагаем $\varphi_{g(i)}(x)=1$. Переходим к шагу $m+1$.

Из конструкции немедленно следует, что $\varphi_{g(i)}$ - о.р.ф. Легко видеть также, что $\varphi_{g(i)} \in U$ и $g(i)$ - о.р.ф.

По теореме о неподвижной точке существует такое a, что $\varphi_a \equiv \varphi_{g(a)}$. Так как при построении $\varphi_{g(a)}$, начиная с некоторого шага, происходит только событие A^m, то, очевидно,

$$\lim_k F(<\varphi_a(0) \dots \varphi_a(k)>) > h(a).$$

Не ограничивая общности, можно считать, что $h(x)$ - неубывающая функция, и поэтому $\lim_k F(<\varphi_a(0) \dots \varphi_a(k)>) > h(a) \geqslant h(\min(\varphi_a))$. Но это противоречит выбору стратегии F и о.р.ф. $h(x)$. Теорема доказана.

ЗАМЕЧАНИЕ. Можно получить точный аналог теоремы I, используя для определения синтеза вместо функционалов предельно вычислимые операции (определение см. в [3]).

ЛИТЕРАТУРА

I. Подниекс К.М. Сравнение различных типов предельного синтеза и прогнозирования функций. - Настоящий сборник, стр.68-81.
2. Gold E.M. Limiting recursion. - "Journal of Symbolic Logic", 1965, 3, No.1.
3. Барздинь Я.М. Об одном свойстве предельно вычислимых функционалов. - Настоящий сборник, стр.20-24.

СОДЕРЖАНИЕ