Разрешимость и перечислимость[1]

Б.Я. Солон

Одним из основных понятий современной науки, и математики, в частности, является понятие алгоритма. Сам термин "алгоритм" или старинное его написание "алгорифм" (ныне почти не употребляемое, кроме словосочетания "нормальный алгорифм Маркова") возник как производное от географического названия, входящего в имя великого ученого средневекового Востока - Муххамад ибн Муса ал-Хорезми, (т.е. происхождением из Хорезма). Латинский перевод написанного на арабском языке арифметического трактата ал-Хорезми начинался словами "Dixit algorizmi", т.е. "Сказал ал-Хорезми". Сначала слово "алгоритм" обозначало правила выполнения арифметических действий в позиционной системе счисления (например, сложение "столбиком"), а затем - для обозначения произвольных процедур, носящих механистический и массовый характер.

Частота использования термина *алгоритм* в современной научной (да и повседневной) лексике едва ли не самая большая. Но и в истории математики остались вычислительные процедуры, названия которых содержат этот термин. Например, *алгоритм Евклида*, который по двум данным натуральным числам находит их наибольший общий делитель. Другой пример - это *решето Эратосфена* - алгоритм, дающий n-ое простое число для любого натурального числа n.

Понятие алгоритма является основным понятием теории вычислимости – одного из наиболее бурно развивающихся разделов математической логики. Благодаря усилиям английского математика Б. Купера в 1998 году была организована ассоциация и серия конференций «Вычислимость в Европе» (Computability in Europe (CiE) см. сайт: www.maths.leeds.ac.uk/cie/). Автор данной статьи является членом этой ассоциации с момента ее образования и участником конференций, состоявшихся в университетах городов Хайдельберг (Германия), Амстердам (Голландия), Свонси (Уэльс), Сиена (Италия), Афины (Греция).

Само название науки об алгоритмах – *теория вычислимости* – возникло недавно по предложению Р. Соара[2]. В СССР основателем теории вычислимости был А.И. Мальцев[3], который в 1965 году опубликовал монографию «Алгоритмы и рекурсивные функции», часть названия которой служило в качестве наименования науки об алгоритмах: *теория рекурсивных функций*. Эта книга возникла в результате обработки конспектов лекций по математической логике, теории алгоритмов и их приложений, которые читал А.И. Мальцев в 1956-1959 годах в Ивановском педагогическом институте. Курс лекций А,И. Мальцева оказал влияние на научные интересы в последствии известных математиков Д.А. Захарова, Е.А. Полякова, А.И. Щеглова, работы которых были посвящены изучению алгебр рекурсивных функций (известных под названием *алгебр Робинсон*). В 1971 году профессор Е.А. Поляков стал научным руководителем аспирантов М.Г. Розинаса и автора данной статьи. Темы их диссертаций были посвящены изучению фундаментального понятия теории вычислимости - *понятия сводимости по перечислимости*.

В 1972 вышел русский перевод[4] монографии Х. Роджерса, название[5] которой закрепило термин вычислимость в теории алгоритмов. Отметим, что первые работы, связанные с формализацией понятия алгоритма, появились значительно раньше – в 30-е годы 20-го века.

---

[1] Полный текст доклада автора на научной конференции «Математические чтения, посвященные 100-летию со дня рождения профессора С.В. Смирнова», 18 февраля 2011 года

[2] Роберт И. Соар, Вычислимо перечислимые множества и степени, Казань, «Казанское математическое общество», 2000. Robert I. Soare, Recursively Enumerable Sets and Degrees, Springer-Verlag, Berlin Heidelberg New York, 1987

[3] А.И. Мальцев, Алгоритмы и рекурсивные функции, «Наука», Москва, 1965

[4] Hartley Rogers, Jr., Theory of Recursive Functions and Effective Computability, McGraw-Hill Book Company, New York, 1967

[5] Х. Роджерс, Теория вычислимых функций и эффективная вычислимость, «Мир», Москва, 1972

По мнению В.А. Успенского и А.Л. Семенова самым главным открытием в науке об алгоритмах является открытие самого понятия алгоритма, как отдельной и самостоятельной сущности. Все формальные модели этого понятия (машины Тьюринга и Поста, нормальные алгорифмы Маркова, машины Колмогорова и др.) являются уточнениями или формальными аналогами интуитивного понятия алгоритма, само же это понятие существовало до первых попыток его формализации и существует независимо от них.

Алгоритм, как самостоятельное понятие, принадлежит к числу тех фундаментальных понятий, которые не могут быть выражены через другие понятия (в том числе, теоретико-множественные), а потому должно рассматриваться как неопределяемое. Мы можем лишь пояснить это понятие примерами, некоторые из которых приведены выше, или выделить существенные черты вычислительной процедуры, которую мы называем алгоритмом. Надо отметить здесь, что термины *алгоритм* и *эффективная вычислительная процедура*, как "вычисления, которые могут быть реально осуществлены"[6], использовались в качестве синонимов еще до появления каких-либо формализаций.

А.А. Марков[7] пояснял, что "алгоритм - это точное предписание, которое задает вычислительный процесс, начинающийся с произвольного (но выбранного из фиксированной для данного алгоритма совокупности) исходного данного и направленный на получение полностью определяемого этим исходным данным результата." Из этого интуитивного представления, в частности, следует, что понятие алгоритма, (как и понятие терма и формулы в математической логике) обладает семантическим свойством "иметь смысл". Более точно, смысл терма или формулы в том, что он (или она) обозначает предмет (или факт), смысл же алгоритма в том, что он должен работать, то есть в том, что он предназначен для получения некоторого результата по исходному данному.

Поэтому для нас алгоритм - это процедура для вычисления некоторой функции (не обязательно, всюду определенной). Одна и та же функция может вычисляться с помощью нескольких алгоритмов. Будем называть функцию, для вычисления значений которой существует хотя бы один алгоритм, *алгоритмически вычислимой* или просто *вычислимой*.

Важно отметить, что необходимо делать различие между понятием алгоритма и понятием алгоритмически вычислимой функции. Это различие особенно ярко проявляется в примерах алгоритмически вычислимых функций, для вычисления значений которых нельзя построить конкретного алгоритма. Такие примеры дают не решенные до настоящего времени задачи из теории чисел. Пусть

$$f(x)=\begin{cases}1, & \textit{если существуе хотя бы } x \textit{ совершенных чисел},\\ 0, & \textit{в противном случае.}\end{cases}$$

В настоящее время не известно даже конечно или бесконечно множество совершенных чисел. (Напомним, n - совершенное число, если n - натуральное и равно сумме всех своих делителей, отличных от n. Первое совершенное число – 6, второе – 28.) Если множество совершенных чисел конечно и их, скажем k, то

$$f(x)=\begin{cases}1, & \textit{если } x\le k,\\ 0, & \textit{если } x>k,\end{cases}$$

а если множество совершенных чисел бесконечно, то $f(x)=\lambda[1]$ (т.е. – тождественная единица). В любом случае $f(x)$ - алгоритмически вычислимая функция, но алгоритма для ее вычисления мы предъявить не в состоянии.

Множество A называется *перечислимым*, если оно либо совпадает с множеством значений некоторой всюду определенной вычислимой функции λxf , либо - пустое. Требование определенности функции f для всех значений ее аргумента x связано с интуитивным со-

[6] Borel E. Le calcul des integrales definies//Journal de Mathematiques pures et appliquees. Ser. 6-V.8,№2(1912)-P.159-210]

[7] А.А. Марков, Математическая логика и вычислительная математика // Вестник АН СССР, №8(1957).-С.21-255

держанием термина "перечислимость". Если Ξ - алгоритм, вычисляющий данную функцию *f*, то подавая на вход Ξ последовательно числа 0,1,2,… на выходе будем получать элементы множества *A*, то есть Ξ "перечисляет" *A*.

Будем говорить, что алгоритм Ξ является *разрешающим* для множества *A*, если он вычисляет его характеристическую функцию c_A. Множество *A* называется *разрешимым*, если для него существует разрешающий алгоритм. Проблема отыскания такого алгоритма называется *проблемой разрешения* для *A*. Таким образом, множество разрешимо тогда и только тогда, когда его проблема разрешения имеет решение.

Интуитивного понятия алгоритма вполне хватало для тех математических задач, решение которых заключалось в построении некоторого вычислительного процесса. Другими словами, пока вопрос существования алгоритма для той или иной проблемы решался положительно, не было предпосылок для возникновения теории алгоритмов. Однако целый ряд классических математических задач, связанных с вопросом существования алгоритмов, не мог получить положительного решения. При этом появилось определенное мнение, что алгоритмов для решения этих задач просто нет. Для доказательства отсутствия требуемого алгоритма необходима математическая формализация понятия алгоритма.

Конец XIX и начало XX веков ознаменовались одним из самых главных открытий в современной математической логике: *был построен точный символический язык, на котором можно сформулировать все теоремы и проводить все доказательства математики и с помощью которого можно дать критерий правильности доказательства*. В *Principia mathematica* Уайтхед и Рассел (1910 г.) изложили значительную часть математики на таком точном символическом языке. Затем математики стали искать универсальную алгоритмическую «разрешающую» процедуру для определения истинности суждений этого языка. Многие математики верили в существование такого алгоритма, был достигнут частичный прогресс в этом направлении, казалось, что успех близок.

Однако в 1931 году в своей эпохальной *теореме Геделя о неполноте* Карл Гедель[8] доказал, что любая достаточно богатая аксиоматизируемая теория (например, содержащая элементарную теорию чисел) неполна, т.е. содержит утверждение, которое само не доказуемо и его отрицание также недоказуемо в этой теории. Из этой теоремы последовал обескураживший многих математиков вывод о том, что невозможен универсальный алгоритм из довольно широкого класса подобных разрешающих процедур. Последующие работы, приведшие формализации понятия алгоритма, в том числе работы Тьюринга, которого вдохновил результат Геделя, показали, что на самом деле широкий класс алгоритмов, к которым применим метод Геделя, включает все алгоритмы, и, следовательно, не существует универсальной разрешающей процедуры.

Кроме работ Геделя и других логиков, стимулирующее влияние на процесс формализации понятия алгоритма оказала 10-я проблема Гильберта. В 1900 году на Втором международном конгрессе математиков немецкий математик Давид Гильберт огласил список из 23 самых важных, по его мнению, нерешенных математических проблем, которые оказали значительное влияние на дальнейшее развитие математики. Проблема под номером десять касалась так называемых диофантовых уравнений, трудность решения которых состоит в ограничении на допустимые значения неизвестных, которые должны быть целыми положительными числами. Уравнения названы в честь греческого математика Диофанта, который жил в третьем веке нашей эры. К концу 19-го века, были найдены решения большого количества диофантовых уравнений, и было доказано, что многие из них решений не имеют. Однако дальнейшее их рассмотрение имело большое значение, поскольку для их решения математикам приходилось изобретать все новые и новые методы. В 10-ой проблеме Гильберт просил решить не конкретные диофантовы уравнения, а найти общий, универсальный метод

[8] Gödel K., Űber formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme//I, Monatsh.Math. und Phys., 38(1931), 173-198

(алгоритм), который будучи применен к конкретному диофантову уравнению, давал ответ на вопрос, имеет ли это уравнение решения или нет.

Ее блестящее решение было сделано в 1970 году русским математиком Юрием Матиясевичем. Им была завершена серия работ американских математиков Джулии Робинсон, Мартина Девиса и Хиллари Путман, приведшая к неразрешимости 10-ой проблемы Гильберта – требуемого в ней алгоритма не существует. В наши дни алгоритм можно отождествить с программой на любом языке программирования. Таким образом, полученное "отрицательное решение" 10-ой проблемы Гильберта состояло в доказательстве невозможности написать программу, которая говорила бы нам, имеет ли то или иное уравнение решения или нет.

Какая же может быть польза от этого доказательства невозможности? Академик Матиясевич сравнил ситуацию с законом сохранения энергии, который делает невозможным построение "вечного двигателя", избавляя тем самым изобретателей от заведомо бесполезной траты времени на его построения, а патентные бюро – от рассмотрения заявок горе-изобретателей. Аналогично доказанная неразрешимость 10-ой проблемы Гильберта дает математикам "моральное право" больше не тратить время на попытки найти универсальный метод решения диофантовых уравнений, а финансирующим органам – отказывать в поддержке проектов, обещающих разработать такой метод.

Теория вычислимости, как самостоятельная наука, началась с момента появления представительных классов алгоритмов. *Представительность* означает, что рассматриваемый класс содержит алгоритм, вычисляющий ту же функцию, что и любой алгоритм (в интуитивном смысле). Исторически первыми примерами представительских классов являются классы алгоритмов, реализованных на вычислительных моделях Тьюринга[9] и Поста[10]. Каждая вычислительная модель представляет собой класс *формально заданных* алгоритмов, то есть по Нагорному[11] "конструктивных объектов определенного вида, содержащих в себе закодированную по фиксированным для алгоритмов данного типа правилам полную информацию об этих алгоритмах." Обычно вычислительная модель формируется таким образом, чтобы построение формального аналога исходного алгоритма и восстановление алгоритма по его формальному аналогу из данной вычислительной модели осуществлялись более просто. Например, в вычислительной модели Тьюринга роль формальных аналогов алгоритмов в интуитивном смысле исполняют машины Тьюринга, представляющие собой, грубо говоря, системы команд.

Важно отметить, что тексты, описывающие вычисления на данной модели, могут содержать информацию, которая не входит в формальное задание конкретного алгоритма. Например, для машин Тьюринга такой информацией является объяснение таких понятий, как "лента", "ячейка", "головка", "сдвиг вправо" и т.д. Далее, формальное задание касается только механизма непосредственной переработки входных и промежуточных данных (в нашем случае – натуральных чисел), но входная и выходная процедуры могут не входить в формальное задание.

Машины Тьюринга с фиксированным ленточным алфавитом и произвольным числом внутренних состояний образуют представительную модель. Это утверждение составляет содержание *тезиса Черча*, который в нашем случае утверждает, что каждая алгоритмически вычислимая арифметическая функция вычислима на подходящей машине Тьюринга.

Сделаем маленькое отступление. Можно заметить, что частота употребления фамилии *Тьюринг* в различных словосочетаниях и контекстах велика. Это – далеко не случайность, именно Алан Тьюринг внес фундаментальный вклад в теорию вычислимости. Он родился в 1912 году, поэтому следующий год – 2012 – будет юбилейным. К столетию со дня рождения Тьюринга планируется провести международную конференцию по теории вычислимости в Англии – на родине Тьюринга (см. сайт: www.turingcentenary.eu).

---

[9] Turing A.M., On computable numbers, with an application to the Entscheidungsproblem // Proceedings of LMS. Ser.2.-V.42,№3,4(1936),-P.230-265]

[10] Post E.L. Finite combinatory processes - formulation I // JSL. - V.1, №3(1936). -P.103-105.

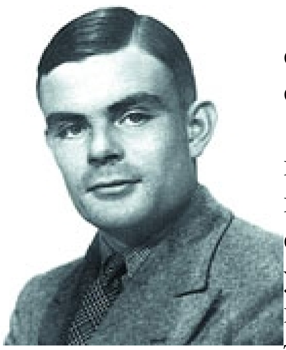
[11] Алгоритма изображение // МЭ. Т.1.(1977)-С.210

Судьба Алана Тьюринга оказалась весьма трагичной. 10 сентября 2009 года Премьер-министр Великобритании Гордон Браун сказал, выражая мнение многих математиков, логиков, программистов и кибернетиков: «*От имени британского правительства и всех тех, кто живёт на свободе благодаря вкладу Алана, я со всей искренностью говорю: прости нас, ты заслуживаешь гораздо лучшего*». Таким образом, он принес публично принёс извинения за те методы, которым был подвергнут Алан Тьюринг: «*С Аланом и с многими тысячами других мужчин-геев, осуждённых по гомофобным законам, обошлись ужасно. А многие миллионы тех, кто не были осуждены, годами жили в постоянном страхе быть осуждёнными за то, что они такие, какие они есть*».

Тьюринг был гомосексуалом. В то время в Великобритании гомосексуальные половые акты были запрещены законом, а гомосексуализм считался психическим заболеванием. В 1952 году ему были предъявлены обвинения в «грубой непристойности» (*gross indecency*) за то, что он был геем. Тьюринг был осужден, и ему предоставили выбор между двухлетним тюремным заключением и гормональной терапией, которая, по сути, была химической кастрацией. Тьюринг выбрал терапию. Одним из эффектов терапии оказалась растущая грудь и снижение либидо. Кроме того, в результате осуждения он потерял право работать в области криптографии.

Через год после вынесения приговора он умер от отравления цианидом, который, видимо, содержался в яблоке, половину которого Тьюринг съел перед смертью. Было признано, что он покончил жизнь самоубийством. Тем не менее, его мать считала, что он отравился случайно, так как всегда небрежно работал с химикатами.

Кстати, то, что логотип корпорации Apple, надкушенное яблоко - дань уважения Тьюрингу. Впрочем, яблоко – это еще и библейский символ познания и греха. Именем Тьюринга названа высшая награда в области программирования, которую можно считать эквивалентом Нобелевской премии в данной области.



Вот краткая биографическая справка. А́лан Тью́ринг (А́лан Ма́тисон Тью́ринг, англ. Alan Mathison Turing; 23.06. 1912 — 7.06. 1954) — английский математик, логик, криптограф, оказавший существенное влияние на развитие информатики. Кавалер Ордена Британской империи (1945).

Сын британского чиновника в Индии, Алан учился во Франции, Англии и, затем, в США. Именно в США в 1936 году Тьюринг предложил проект простого устройства, имеющего все основные свойства современной информационной системы: программное управление, память, и пошаговый способ действий. Эта воображаемая машина, получившая название машины Тьюринга, используется в теории автоматов и компьютеров.

Когда Тьюринг из США возвратился в Англию, началась вторая мировая война. Одним из важнейших вооружений этой войны была ЭВМ «Колосс» по проекту «Ультра», начавшая в 1943 году взламывать сверхсложные шифры немцев. Работа Тьюринга в этом проекте значительно помогла в борьбе с Германией и её союзниками.

После войны в 1945 году Алан возглавил проект создания компьютера «ТУЗ» (ACE, Automatic Computing Engine), а в 1948 году Тьюринг стал работать с «МАДАМ» (MADAM, Manchester Automatic DigitAl Machine), компьютером с самой большой памятью в мире в то время. Работы Алана по сооружению первых ЭВМ и развитию методов программирования имели неоценимую важность, дав основу большинству исследований в области искусственного интеллекта.. Он полагал, что компьютеры, в конце концов, смогут мыслить как человек, и предложил простую проверку, известную как тест Тьюринга, оценивающую способность машины мыслить: побеседуйте с ЭВМ, и пусть она убедит вас, что она — человек.

Последствия суда 1952 года были катастрофическими — Алана Тьюринга уволили из шифроаналитического бюро и Манчестерского университета. Правда, потом ему все-таки вернули возможность преподавать. Тем не менее, учёный до 1954 года прожил в затворничестве, играя в свою любимую игру «Необитаемый остров», которая заключалась в получении всевозможных химических веществ из популярных продуктов. 8 июня 1954 года Алан Мэтисон Тьюринг был найден мертвым в своем доме.

Алгоритм с оракулом - это обычный алгоритм, который имеет некоторый вспомогательный устройство, позволяющий в любой момент алгоритмического процесса проверить истинность утверждения $n \in A$, где n - некоторый промежуточный результат и A – данное множество. Сам оракул не является частью данного алгоритма, и с точки зрения вычислительных средств, которые используются в данном вычислительном процессе, является неким "невычислимым" алгоритмом. Однако, каждый шаг процесса, задаваемого алгоритмом с оракулом, определяется не только возникшему к этому шагу состоянию, но и параметром $i \in \{0,1\}$, значение которого зависит от истинности предиката $n \in A$. Алгоритм с оракулом A называется также *алгоритмом относительно A*. Формально, задание алгоритма с оракулом в виде конечного текста не отличается от задания обычного алгоритма. Поэтому можно использовать перечисление всех текстов (возможно, с дополнительными символами для обращения к оракулу) для построения нумерации алгоритмов с оракулами. Обозначим через $\lambda X \Xi_n^X$ алгоритм с оракулом X с номером n (сам оракул X не является частью алгоритма Ξ_n.

С теоретической точки зрения понятие алгоритма с оракулом позволяет релятивизировать теорию алгоритмов. Другое важное значение понятия алгоритма относительно A - интуитивная подоплека такого фундаментального понятия теории вычислимости, как понятие *сводимости по разрешимости*. В оракульных терминах множество *B сводится по разрешимости* к множеству A, если существует алгоритм с оракулом A, который вычисляет характеристическую функцию множества B. Понятие алгоритма с оракулом и сам термин "оракул" впервые появились в статье Тьюринга[12], вышедшей в 1939 г. По этой причине Пост[13] ввел термин *сводимость по Тьюрингу* (или, как чаще говорят, *Т-сводимость*) для обозначения самой общей сводимости по разрешимости.

Вычислимая функция по одному числу эффективно (то есть в результате работы подходящего алгоритма) дает другое число. "Множественным" аналогом вычислимой функции является, так называемая, *вычислимая операция*, которая должна по одному множеству эффективно "породить" другое множество. Другими словами, вычислимая операция - это алгоритм, позволяющий порождать результирующее множество, если задан процесс порождения исходного множества. Первое определение вычислимой операции было дано Успенским[14] в 1955 году. Роджерс[15], по- видимому независимо, дал определение *оператора перечисления*, которое по объему совпадает с понятием вычислимой операции.

Дадим определение оператора перечисления Φ в терминах (интуитивно) перечислимых множеств, как это сделано Роджерсом в его монографии. Чтобы "породить" множество

---

[12] Turing A.M., Systems of logic based on ordinals // Proceedings of LMS. Ser.2.-V.45,№3(1939)-P.161-228

[13] Post E.L., Recursively enumerable sets of positive integers and their decision problems // Bulletin of AMS.-V.50,№5(1944).-P.284-316]

[14] Успенский В.А.,О вычислимых операциях //ДАН.-Т.103,№5(1955).- С.773-776]

[15] Rogers H.,Jr., Theory of recursive function and effective computability.- New York et al.: McGraw-Hill Book Company(1967).- XIX+482 p., \S 9.7],

$\Phi(X)$ из исходного множества *X*, мы одновременно порождаем некоторое перечислимое множество *W* и конечные подмножества *X*. Всякий раз, когда элемент из уже перечисленной части *W*, который мы рассматриваем как номер пары $\langle x,u \rangle$, имеет вторую компоненту *u*, совпадающую с каноническим индексом с уже порожденным конечным подмножеством *X*, мы зачисляем соответствующую первую компоненту *x* в $\Phi(X)$. Более формально, пусть W - произвольное данное перечислимое множество. Отображение $\Phi: 2^{\omega} \to 2^{\omega}$ называется *оператором перечисления*, если $\Phi(X) = \{x : \exists u[\langle x,u \rangle \in W \ \& \ D_u \subseteq X]\}$ для любого множества $X \subseteq \omega$.

Операторы перечисления можно интерпретировать как интуитивную подоплеку еще одного фундаментального понятия теории вычислимости - *понятия сводимости по перечислимости*. Говорят, что множество *B* сводимо по перечислимости или *е-сводимо* (от английского аналога *enumeration reducibility*) к множеству *A*, если существует оператор перечисления Φ, такой, что $B = \Phi(A)$.

С помощью вычислимых операций легко определить Тьюрингову сводимость множеств: множество *B* сводимо по разрешимости к множеству *A* тогда и только тогда, когда множество $B \oplus \overline{B}$ сводимо по перечислимости к множеству $A \oplus \overline{A}$.

Класс множеств, имеющих одни и ту же «сложность» проблемы разрешения, т.е. в терминах формализации Тьюринга, взаимно сводимых по Тьюрингу друг другу, называется *тьюринговой степенью* или *Т-степенью*. Т-степень данного множества *A* часто называют степенью *неразрешимости* множества *A*. Множество Т-степеней с индуцированным на нем естественным образом отношением частичного порядка образует верхнюю полурешетку с наименьшим элементом – Т-степенью, состоящей из всех вычислимых множеств. Обозначим ее через $\boldsymbol{D}_T$.

Класс множеств, имеющих одни и ту же «сложность» проблемы перечисления, т.е. в операторов перечисления, взаимно сводимых по перечислимости друг другу, называется *степенью перечислимости* или *е-степенью*. Е-степень данного множества *A* называются степенью *перечислимости* множества *A*, хотя по аналогии с Т-степенями, ее следовало бы называть степенью *НЕперечислимости*. Множество е-степеней с индуцированным на нем естественным образом отношением частичного порядка образует верхнюю полурешетку с наименьшим элементом – е-степенью, состоящей из всех вычислимо перечислимых множеств. Обозначим ее через D_e.

Существует изоморфное вложение $\iota: \boldsymbol{D}_T \to \boldsymbol{D}_e$. Подмножество $\iota(\boldsymbol{D}_T) \subset \boldsymbol{D}_e$ образует подполурешетку в D_e, которая обозначается через $\boldsymbol{Tot}$ и состоит из так называемых *тотальных е-степеней*. Существование нетотальных е-степеней было доказано московским математиком Ю.Т. Медведевым[16], таким образом $\boldsymbol{D}_e - \boldsymbol{Tot} \neq \phi$. Изучению нетотальных е-степеней посвящена большая часть работ автора[17].

[16] Медведев Ю.Т., Степени трудностей массовых проблем// ДАН СССР, 102(1955), 501-504

[17] Солон Б.Я., Нетотальные степени перечислимости// Дисс. на соискание ученой степени доктора ф.-м.н., СПГУ, 2003.