Важное свойство А. я. — возможность регуляризации структуры программы, облегчающей ее написание, изучение и проверку правильности. Основными средствами регуляризации являются ограничения на структуру областей действия, тел процедур и циклов, а также регламентация употребления операторов передачи управления, ограничивающая ветвимость цепочек выполнения базовых операторов параллельно-последовательной структурой.

Программирование с использованием А. я. для вычислительной машины требует включения в ее программное обеспечение специальных программирующих процессоров — посредников между программами на А. я. и машиной. Процессор в своем развернутом составе выполняет следующие функции: ввод программы, лексический (выделение и классификация лексем), синтаксический и семантич. анализы с сигнализацией о формальных ошибках, синтез промежуточной формы (представление программы на внутреннем языке нек-рой абстрактной вычислительной машины, удобное для последующей обработки или выполнения программы), оптимизация (систематич. преобразование промежуточной формы, улучшающее такие характеристики программы, как ее размер, скорость и объем расходуемой памяти), генерация (построение машинной программы), выполнение программы. В процессорах транслирующего типа (трансляторы, компиляторы, программирующие программы) выполнение программы происходит после полного построения машинной программы. В процессорах интерпретирующего типа (шаговые, диалоговые, отладочные трансляторы) программа выполняется с помощью нек-рого механизма интерпретации ее либо промежуточной формы, либо дерева разбора, либо даже исходного текста.

Различают одно-, двух- и многофазные схемы трансляции программ. В однофазной схеме все функции транслятора объединены в один просмотр текста программы. Оптимизация и промежуточная форма отсутствуют, а выделяемые вершины дерева разбора тут же порождают машинные команды программы. В двухфазных трансляторах обычно на первом просмотре программы строится дерево разбора, а на втором просмотре семантич. анализ объединяется с генерацией. Многофазные схемы с использованием промежуточной формы применяются в оптимизирующих трансляторах или в системах со сменной генерацией для различных типов машин, а для нек-рых А. я. требуются в силу сложности синтаксич. и семантич. анализов.

Количество разнообразных А. я., созданных для работы на вычислительных машинах, весьма велико (больше тысячи), однако только нек-рые из них получили широкое распространение. К ним относятся языки *алгол*, *алгол-68*, *кобол*, *лисп*, *ПЛ/1*, *симула*, *фортран*, а в СССР также *алгамс*, *альфа*, *рефал*.

Лит.: [1] Ингерман П., Синтаксически ориентированный транслятор, пер. с англ., М., 1969; [2] Языки программирования, пер. с англ., М., 1972; [3] Хопгуд Ф., Методы компиляции, пер. с англ., М., 1972; [4] Хигман Б., Сравнительное изучение языков программирования, пер. с англ., М., 1974. *А. П. Ершов.*

АЛГОРИТМОВ СОЧЕТАНИЯ — название, установившееся за рядом конкретных способов конструирования новых *алгоритмов* из нескольких заданных.

В применении к *нормальным алгорифмам* наибольшую известность получили следующие А. с.: нормальная композиция ($\mathfrak{B}\circ\mathfrak{A}$) двух нормальных алгорифмов $\mathfrak{A}$ и $\mathfrak{B}$, нормальное объединение ($\mathfrak{A}\frown\mathfrak{B}$) двух нормальных алгорифмов $\mathfrak{A}$ и $\mathfrak{B}$, нормальное разветвление ($\mathfrak{A}\curlyvee\mathfrak{B}\mid\mathfrak{C}$) двух нормальных алгорифмов $\mathfrak{A}$ и $\mathfrak{B}$, управляемое нормальным алгорифмом $\mathfrak{C}$, нормальное повторение ($\mathfrak{A}\bigcirc\mathfrak{C}$) нормального алгорифма $\mathfrak{A}$, управляемое нормальным алгорифмом $\mathfrak{C}$. Если $\mathfrak{A}$, $\mathfrak{B}$, $\mathfrak{C}$ — нормальные алгорифмы в нек-ром алфавите A, то упомянутые их сочетания являются нормальными алгорифмами в нек-ром фиксированном расширении A и удовлетворяют следующим условиям: а) для любого слова P в A имеет место $(\mathfrak{B}\circ\mathfrak{A})\llcorner P\lrcorner \simeq \mathfrak{B}\llcorner\mathfrak{A}\llcorner P\lrcorner\lrcorner$ (теорема композиции); б) для любого слова P в A имеет место $(\mathfrak{A}\frown\mathfrak{B})\llcorner P\lrcorner \simeq \mathfrak{A}\llcorner P\lrcorner\mathfrak{B}\llcorner P\lrcorner$ (теорема объединения); в) для любого слова P в A

$$(\mathfrak{A}\curlyvee\mathfrak{B}\mid\mathfrak{C})\llcorner P\lrcorner \simeq \begin{cases} \mathfrak{A}\llcorner P\lrcorner, & \text{если } \mathfrak{C}\llcorner P\lrcorner \mathrel{\underline{\overline{\circ}}} \Lambda, \\ \mathfrak{B}\llcorner P\lrcorner, & \text{если } \mathfrak{C}\llcorner P\lrcorner \mathrel{\not\underline{\overline{\circ}}} \Lambda, \end{cases}$$

причем если $(\mathfrak{A}\curlyvee\mathfrak{B}\mid\mathfrak{C})\llcorner P\lrcorner$ определено, то определено и $\mathfrak{C}\llcorner P\lrcorner$ (теорема разветвления); г) для любых слов P и Q в A *графическое равенство* $(\mathfrak{A}\bigcirc\mathfrak{C})\llcorner P\lrcorner \mathrel{\underline{\overline{\circ}}} Q$ имеет место тогда и только тогда, когда может быть указан ряд слов $P_0, P_1, \ldots, P_k$ $(k\geqslant 1)$ в алфавите A таких, что

$$P_0 \mathrel{\underline{\overline{\circ}}} P,$$

$$P_k \mathrel{\underline{\overline{\circ}}} Q,$$

$$P_i \mathrel{\underline{\overline{\circ}}} \mathfrak{A}\llcorner P_{i-1}\lrcorner \quad (i=1, \ldots, k),$$

$$\mathfrak{C}\llcorner P_k\lrcorner \mathrel{\not\underline{\overline{\circ}}} \Lambda \quad (i=1, \ldots, k-1),$$

$$\mathfrak{C}\llcorner P_k\lrcorner \mathrel{\underline{\overline{\circ}}} \Lambda$$

(теорема повторения). Аналогичные теоремы могут быть получены и для *Тьюринга машин*. В теории *рекурсивных функций* наибольшее употребление нашли их сочетания, доставляемые оператором подстановки, оператором примитивной *рекурсии* и μ-оператором.

Теоремы об А. с. вскрывают весьма существенную особенность осуществленных стандартизаций общего понятия алгоритма — их «устойчивость» по отношению к естественным способам А. с. Это обстоятельство является одним из наиболее веских доводов в пользу основной гипотезы теории алгоритмов (*Чёрча тезиса*). Теоремы об А. с. составляют важный раздел общей теории алгоритмов. Будучи доказаны однажды, они позволяют в дальнейшем убеждаться в осуществимости сложных и громоздких алгоритмов без фактического выписывания определяющих их схем.

Значительный интерес для общей теории алгоритмов представляет вопрос о разыскании базиса, позволяющего при фиксированном наборе способов А. с. порождать любой алгоритм к.-л. интересующего нас класса.

Лит.: [1] Марков А. А., Теория алгорифмов, «Тр. Матем. ин-та АН СССР», 1954, т. 42, с. 94—145; [2] Клини С. К., Введение в метаматематику, пер. с англ., М., 1957; [3] Успенский В. А., Лекции о вычислимых функциях, М., 1960; [4] Мальцев А. И., Алгоритмы и рекурсивные функции, М., 1965. *Н. М. Нагорный.*

АЛГОРИТМОВ ТЕОРИЯ — раздел математики, изучающий общие свойства *алгоритмов*. Содержательные явления, приведшие к образованию понятия «алгоритм», прослеживаются в математике в течение всего времени ее существования. Однако само это понятие сформировалось лишь в 20 в. и стало предметом самостоятельного изучения (по-видимому, впервые, хотя еще в расплывчатом виде) в 20-х гг. 20 в. в трудах представителей *интуиционизма* Л. Э. Я. Брауэра (L. E. J. Brouwer) и Г. Вейля (H. Weyl, см. [1]). Началом систематич. разработки А. т. можно считать 1936, когда А. Чёрч (A. Church, [2]) опубликовал первое уточнение понятия *вычислимой функции* (предложив отождествлять понятие всюду определенной вычислимой функции, имеющей натуральные аргументы и значения, с понятием общерекурсивной функции) и привел первый пример функции, не являющейся вычислимой, а А. М. Тьюринг (A. M. Turing, [3], [4]) и Э. Л. Пост (E. L. Post, [5]) дали первые уточнения понятия алгоритма (в терминах идеализированных вычислительных машин, см. *Тьюринга машина*). В дальнейшем А. т. получила развитие в трудах С. К. Клини (S. C. Kleene), Э.Л. Поста (см. [6] — [8]), А. А. Маркова (см. [9] — [11]) и др. В частности, А. А. Марков предложил уточнить понятие алгоритма с помощью введенного им понятия *нормального алгорифма* (см. [10]). Наиболее общий подход к уточнению понятия алгоритма предложил А. Н. Колмогоров (см. [12], [13]).

Поскольку алгоритмы могут иметь дело лишь с так наз. *конструктивными объектами*, то, чтобы перенести понятия и методы А. т. на случай неконструктивных объектов, надо эти последние объекты обозначить, или поименовать, конструктивными объектами. Изучение общих свойств таких поименований (прежде всего, когда в качестве обозначений, или имен, выступают натуральные числа) составляет предмет теории *нумерации* (см. [14]), образующей важный раздел А. т.

Основные понятия теории алгоритмов. Областью применимости алгоритма наз. совокупность тех объектов, к к-рым он применим. Про алгоритм $\mathfrak{A}$ говорят, что он: 1) «вычисляет функцию f», коль скоро его область применимости совпадает с областью определения f, и $\mathfrak{A}$ перерабатывает всякий x из своей области применимости в $f(x)$; 2) «разрешает множество A относительно множества X», коль скоро он применим ко всякому x из X и перерабатывает всякий x из $X \cap A$ в слово «да», а всякий x из $X \setminus A$ в слово «нет»; 3) «перечисляет множество B», коль скоро его область применимости есть натуральный ряд, а совокупность результатов есть B. Функция наз. вычислимой, если существует вычисляющий ее алгоритм. Множество наз. разрешимым относительно X, если существует разрешающий его относительно X алгоритм. Множество наз. перечислимым, если либо оно пусто, либо существует перечисляющий его алгоритм.

Детальный анализ понятия «алгоритм» обнаруживает, что: (I) область возможных исходных данных и область применимости любого алгоритма суть перечислимые множества. В свою очередь, (II) для любой пары вложенных одно в другое перечислимых множеств можно подобрать алгоритм, у к-рого большее множество служит областью возможных исходных данных, а меньшее — областью применимости. Имеют место следующие основные теоремы: (III) функция f вычислима тогда и только тогда, когда перечислим ее график, т. е. множество всех пар вида $\langle x, f(x)\rangle$. (IV) Подмножество A перечислимого множества X тогда и только тогда разрешимо относительно X, когда A и $X \setminus A$ перечислимы. (V) Если A и B перечислимы, то $A \cup B$ и $A \cap B$ также перечислимы. (VI) В каждом бесконечном перечислимом множестве X существует перечислимое подмножество с неперечислимым дополнением [в силу (IV) это перечислимое подмножество будет неразрешимым относительно X]. (VII) Для каждого бесконечного перечислимого множества X существует вычислимая функция, определенная на подмножестве этого множества и не продолжаемая до вычислимой функции, определенной на всем X. Утверждения (VI) и (II) в совокупности дают упоминаемый в статье *Алгоритм* пример алгоритма с неразрешимой областью применимости. Разрешимые и перечислимые множества составляют простейшие (и наиболее важные) примеры множеств, строение к-рых задается с помощью тех или иных алгоритмич. процедур. Систематич. изучение множеств конструктивных объектов с точки зрения таких свойств этих множеств, к-рые связаны с наличием тех или иных алгоритмов, образует так наз. алгоритмическую теорию множеств; нек-рые понятия, методы и результаты этой теории находят глубокие аналогии в *дескриптивной теории множеств*.

Алгоритмические проблемы. Проблема построения алгоритма, обладающего теми или иными свойствами, наз. *алгоритмической проблемой* (а. п.). Как правило, свойство искомого алгоритма формулируется в терминах свойств того соответствия, к-рое должно иметь место между исходными данными и результатами алгоритма. Важные примеры а. п.: проблема вычисления данной функции (требуется построить алгоритм, вычисляющий эту функцию); проблема разрешения данного множества (требуется построить алгоритм, разрешающий это множество относительно нек-рого другого множества); проблема перечисления данного множества (требуется построить алгоритм, перечисляющий данное множество). Все примеры а. п. из различных областей математики, названные ниже в разделе «Приложения», являются проблемами разрешения. Неразрешимость а. п. означает отсутствие соответствующего алгоритма; теоремы, устанавливающие неразрешимость таких проблем, относятся к числу наиболее важных теорем А. т. Так, для алгоритма с неразрешимой областью применимости неразрешима а. п. разрешения этой области относительно области возможных исходных данных. Посредством сведения к неразрешимости этой проблемы была установлена неразрешимость большинства других проблем разрешения (в частности, всех проблем, указанных ниже в разделе «Приложения»). Вопрос о том, для любой ли неразрешимой проблемы разрешения ее неразрешимость может быть установлена таким способом, составляет так наз. проблему *алгоритмической сводимости*.

Метрическая теория алгоритмов. А. т. можно разделить на дескриптивную (качественную) и метрическую (количественную). Первая — исследует алгоритмы с точки зрения устанавливаемого ими соответствия между исходными данными и результатами; к ней относятся, в частности, те алгоритмич. проблемы, о к-рых говорилось в предыдущем разделе. Вторая — исследует алгоритмы с точки зрения сложности как самих алгоритмов (см. *Алгоритма сложность*), так и задаваемых ими «вычислений», то есть процессов последовательного преобразования конструктивных объектов. Важно подчеркнуть, что как сложность алгоритмов, так и сложность вычислений могут определяться различными способами, причем может оказаться, что при одном способе A будет сложнее B, а при другом способе — наоборот. Чтобы говорить о сложности алгоритмов, надо сначала описать к.-л. точный язык для записи алгоритмов и затем под сложностью алгоритма понимать сложность его записи; сложность же записи можно определять различными способами (напр., как число символов данного типа, участвующих в записи, или как набор таких чисел, вычисленных для разных типов символов). Чтобы говорить о сложности вычисления, надо уточнить, как именно вычисление представляется в виде цепочки сменяющих друг друга конструктивных объектов и что считается сложностью такой цепочки (только ли число членов в ней — «число шагов» вычисления, или еще учитывается «размер» этих членов и т. п.); в любом случае сложность вычисления зависит от исходного данного, с к-рого начинается вычисление, поэтому сложность вычисления есть функция, сопоставляющая с каждым объектом из области применимости алгоритма сложность соответствующей цепочки. Разработка методов оценки сложности алгоритмов и вычислений имеет важное теоретич. и практич. значение, однако в отличие от дескриптивной А. т., оформившейся в целостную математич. дисциплину (см. [11], [15], [16]), метрич. А. т. находится в процессе становления (см. [17] — [20]).

Приложения теории алгоритмов имеются во всех областях математики, в к-рых встречаются алгоритмич. проблемы. Такие проблемы возникают в *математической логике* и *моделей теории*; для каждой теории формулируется проблема разрешения множества всех истинных или доказуемых предложений этой теории относительно множества всех ее предложений (теории подразделяются на разрешимые и неразрешимые — в зависимости от разрешимости или неразрешимости указанной проблемы); в 1936 А. Чёрч (см. [21], [22]) установил неразрешимость проблемы разрешения для множества всех истинных предложений логики предикатов, дальнейшие важные результаты в этом направлении принадлежат А. Тарскому (A. Tarski), А. И. Мальцеву и др. (см. [23]). Неразрешимые алгоритмич. проблемы встречаются в алгебре (проблема тождества для полугрупп и, в частности, для групп); первые примеры полугрупп с неразрешимой проблемой тождества были найдены в 1947 независимо А. А. Марковым (см. [9]) и Э. Л. Постом (см. [8]), а пример группы с неразрешимой проблемой тождества — в 1952 П. С. Новиковым (см. [24], [25]); в топологии (проблема гомеоморфии, неразрешимость к-рой для важного класса случаев была доказана в 1958 А. А. Марковым, (см. [26]); в теории чисел (проблема разрешимости диофантовых уравнений, неразрешимость к-рой была установлена в 1970 Ю. В. Матиясевичем, (см. [27], [28]) и др. разделах математики.

А. т. тесно связана с математич. логикой, поскольку на понятие алгоритма опирается одно из центральных понятий математич. логики — понятие *исчисления*, и потому, напр., *Гёделя теорема о неполноте* формальных систем может быть получена как следствие теорем А. т. (см. [29]). Наконец, А. т. тесно связана с основаниями математики, в к-рых одно из центральных мест занимает проблема соотношения конструктивного и неконструктивного; в частности, А. т. дает аппарат, необходимый для разработки конструктивного направления в математике; в 1965 А. Н. Колмогоров предложил использовать А. т. для обоснования теории информации (см. [30], *Алгоритмическая теория информации*). А. т. образует теоретич. фундамент для ряда вопросов вычислительной математики и тесно связана с кибернетикой, в которой важное место занимает изучение алгоритмов управления.

Лит.: [1] Вейль Г., О философии математики, пер. с нем., М.— Л., 1934; [2] Church A., «Amer. J. Math.», 1936, v. 58, № 2, p. 345—63; [3] Turing A. M., «Proc. London Math. Soc.», ser. 2, 1936, v. 42, №№ 3—4, p. 230—65; [4] его же, там же, 1937, v. 43, № 7, p. 544—46; [5] Post E. L., «J. Symbol. Log.», 1936, v. 1, № 3, p. 103—05; [6] его же, «Amer. J. Math.», 1943, v. 65, № 2, p. 197—215; [7] его же, «Bull. Amer. Math. Soc.», 1944, v. 50, № 5, p. 284—316; [8] его же, «J. Symbol. Log.», 1947, v. 12, № 1, p. 1—11; [9] Марков А. А., «Докл. АН СССР», 1947, т. 55, № 7, с. 587—90; [10] его же, «Тр. Матем. ин-та АН СССР», 1951, т. 38, с. 176—89; [11] его же, Теория алгорифмов, М.— Л., 1954; [12] Колмогоров А. Н., «Успехи матем. наук», 1953, т. 8, № 4, с. 175—76; [13] Колмогоров А. Н., Успенский В. А., там же, 1958, т. 13, № 4, с. 3—28; [14] Ершов Ю. Л., Теория нумераций, ч 1—2, Новосибирск, 1969—73; [15] Мальцев А. И., Алгоритмы и рекурсивные функции, М., 1965; [16] Роджерс Х., Теория рекурсивных функций и эффективная вычислимость, пер. с англ., М., 1972; [17] Марков А. А., «Изв. АН СССР. Сер. матем.», 1967, т. 31, № 1, с. 161—208; [18] Трахтенброт Б. А., Сложность алгоритмов и вычислений, Новосибирск, 1967; [19] Проблемы математической логики. Сложность алгоритмов и классы вычислимых функций, сб. переводов, М., 1970; [20] Сложность вычислений и алгоритмов, сб. переводов, М., 1974; [21] Church A., «J. Symbol. Log.», 1936, v. 1, № 1, p. 40—41; [22] его же, там же, № 3, p. 101—02; [23] Ершов Ю. Л., Лавров И. А., Тайманов А. Д., Тайцлин М. А., «Успехи матем. наук», 1965, т. 20, № 4, с. 37—108; [24] Новиков П. С., «Докл. АН СССР», 1952, т. 85, с. 709—12; [25] его же, Об алгоритмической неразрешимости проблемы тождества слов в теории групп, М., 1955; [26] Марков А. А., «Докл. АН СССР», 1958, т. 121, № 2, с. 218—20; [27] Матиясевич Ю. В., там же, 1970, т. 191, № 2, с. 279—82; [28] его же, «Успехи матем. наук», 1972, т. 27, № 5, с. 185—222; [29] Успенский В. А., там же, 1974, т. 29, № 1, с. 3—47; [30] Колмогоров А. Н., «Проблемы передачи информации», 1965, т. 1, № 1, с. 3—11.

В. А. Успенский.

АЛГОРИТМОВ ЭКВИВАЛЕНТНОСТЬ — *бинарное отношение*, связывающее алгоритмы фиксированного типа и выражающее тот факт, что у всяких двух связанных этим отношением алгоритмов при совпадении определенного вида исходных данных совпадают и результаты работы (а также, быть может, и нек-рые дополнительные сведения относительно выполненных при этом вычислений — так наз. истории вычислений). Ниже приведено несколько типичных примеров таких отношений.

а) Рассматриваются всевозможные рекурсивные схемы — системы равенств, определяющие n-местные частично рекурсивные функции. Две схемы, определяющие функции φ и ψ, наз. функционально эквивалентными, если для любых натуральных чисел $x_1, \ldots, x_n$ имеет место условное равенство

$$\varphi(x_1, \ldots, x_n) \simeq \psi(x_1, \ldots, x_n)$$

(оно считается верным, если обе его части осмысленны одновременно и в случае осмысленности принимают одинаковые значения). С. К. Клини (S. C. Kleene; см., напр., [1], с. 314) показал, что для любого всюду определенного вычислимого оператора $\mathfrak{R}$ над рекурсивными схемами можно указать схему S такую, что S и $\mathfrak{R}(S)$ функционально эквивалентны (так наз. теорема о неподвижной точке). Отсюда, в частности, вытекает теорема Успенского — Райса о неразрешимости произвольного инвариантного относительно функциональной эквивалентности свойства рекурсивных схем при условии, что имеются схемы S_1 и S_2 такие, что S_1 обладает этим свойством, а S_2 им не обладает. Из этой теоремы следует неразрешимость и самого отношения функциональной эквивалентности.

б) Рассматриваются *нормальные алгорифмы* над фиксированным алфавитом A. Два таких алгорифма $\mathfrak{A}$ и $\mathfrak{B}$ наз. эквивалентными относительно A (см. [2], с. 51), если для любого слова P в A выполняется следующее условие: если один из этих алгорифмов перерабатывает P в нек-рое слово Q, снова оказывающееся в алфавите A, то и второй из них также перерабатывает P в Q. Те же алгорифмы наз. вполне эквивалентными относительно A, если для любого P в A выполняется условное равенство

$$\mathfrak{A}(P) \simeq \mathfrak{B}(P).$$

Оба эти отношения неразрешимы.

в) Рассматриваются логич. схемы алгоритмов (схемы Янова, см. [3]). Реализацией такой схемы наз. последовательность операторов, выполняемых при прохождении по этой схеме от начала до конца. Две схемы наз. эквивалентными, если множества их реализаций совпадают. Отношение эквивалентности схем Янова оказалось разрешимым, и для него была построена полная система эквивалентных преобразований (см. [3], [4]).

Детальное изучение отношений А. э. представляет большой интерес в связи с рядом актуальных задач (особенно в области теории схем программ), требующих для своего решения развитой техники эквивалентных преобразований алгоритмов. Таковы, напр., задача трансляции алгоритмов (перевод с одного алгоритмич. языка на другой) и их оптимизации (преобразование с целью улучшения «рабочих характеристик»). В этом круге вопросов особое внимание уделяют (см., напр., [5]) возможности отыскания таких разрешимых отношений А. э., к-рые допускали бы удобные в приложениях полные системы эквивалентных преобразований. Концепция, намеченная в [3], во многом предопределила общий подход к исследованию отношений А. э.

Лит.: [1] Клини С. К., Введение в метаматематику, пер. с англ., М., 1957; [2] Марков А. А., Теория алгорифмов, М., 1954; [3] Янов Ю. И., «Проблемы кибернетики», 1958, в. 1, с. 75—127; [4] Ершов А. П., «Проблемы кибернетики», 1968, в. 20; [5] его же, там же, 1973, в. 27.

Н. М. Нагорный.

АЛЕКСАНДЕРА ДВОЙСТВЕННОСТЬ — связь между гомологич. свойствами взаимно дополнительных подмножеств топологич. пространства, к-рая позволяет гомологич. свойства множества определять нек-рыми свойствами его дополнения. Первые теоремы такого рода