смотренным в заголовке процедуры с данным идентификатором. Выражение может содержать указатель функции, означающий вызов процедуры, предназначенной для вычисления одной величины. Допускается рекурсивный вызов процедур, т. е. такой вызов, при к-ром в процессе выполнения процедуры происходит вызов той же процедуры. В конкретных представлениях А. часто сокращаются языковые возможности, имеющиеся в эталонном А. С целью унификации таких сокращений разработан алгоритмич. язык, наз. подмножеством алгол-60 и представляющий собой упрощенный вариант эталонного А., из к-рого исключены возможности, вызывающие особые трудности при разработке трансляторов для небольших машин. В качестве преемника алгола-60 был предложен язык *алгол-68*, к-рый существенно отличается по структуре от алгола-60, содержит много новых понятий и возможностей и рассчитан на более мощные машины.

Лит.: [1] Алгоритмический язык АЛГОЛ-60, пер. с англ., М., 1965; [2] Лавров С. С., Универсальный язык программирования (АЛГОЛ-60), 2 изд., М., 1967; [3] Ван Вейнгаарден А. [и др.], Сообщение об алгоритмическом языке АЛГОЛ-68, «Кибернетика», 1969, № 6, с. 23—145; 1970, № 1, с. 13—160. *В. В. Мартынюк.*

АЛГОЛ-68 — универсальный *алгоритмический язык*, разработанный в 1964—68 коллективом ученых 12 стран в составе рабочей группы по *алголу* Международной федерации по обработке информации для обмена алгоритмами, для эффективного их выполнения на различных вычислительных машинах и как средство для изучения алгоритмов. Сохраняя стилистич. связь с алголом-60, А.-68 существенно отличается от него богатством и общностью конструкций. Основными видами данных, в дополнение к типам алгола-60 «вещественный», «целый» и «логический», могут быть «литерный» (для буквенно-цифровой информации), «форматный» (для описания формата внешней информации), имя и процедура. Таким образом, имена и процедуры могут «вычисляться» при выполнении программы на А.-68, хотя это вычисление ограничивается динамич. выбором значения имени или процедуры из явно заданной конечной совокупности. Из основных видов можно индуктивно строить новые, составные виды, представляющие либо однородные индексируемые последовательности данных одного вида (мультизначения), либо упорядоченные наборы данных произвольного вида (структуры).

В дополнение к обычному аппарату описания процедур А.-68 содержит средства для описания так наз. инфиксных операций типа $x+y$. Наличие описания приоритета позволяет задавать отношения старшинства между вводимыми инфиксными операциями. Характерное для А.-68 описание тождества является универсальной конструкцией для описания переменных, задания начальных значений, организации подстановки фактич. параметров в процедуры и для задания синонимии.

В А.-68 в позиции выражения может стоять оператор присваивания или даже любая цепочка операторов, вырабатывающая нек-рое значение. В сочетании с возможностью вычисления имен и процедур, а также введением парных скобок для условных выражений это приводит к допустимости в А.-68 конструкций, поясняемых следующим примером:

1) Алгол-68:

$$\textbf{если } x > 0, \textbf{ то } u \textbf{ иначе } z \textbf{ все}$$
$$:= a + (m < n \mid sin \mid cos)\,(t := x \uparrow 2),$$

2) Алгол-60:

$$t := x \uparrow 2;$$
$$r := a + \textbf{если } m < n \textbf{ то } sin\,(t) \textbf{ иначе}$$
$$\cos\,(t); \textbf{ если } x > 0 \textbf{ то } u := r \textbf{ иначе } z := r.$$

Программа в А.-68 состоит из замкнутых, последовательных, условных и совместных предложений. Первые три обобщают такие понятия алгола-60, как блок, составной оператор и условные выражение и оператор. Совместные предложения обозначают неупорядоченные совокупности составляющих фраз, являясь, в частности, основным средством для указания параллельных ветвей в общем ходе выполнения программы.

Описание семантики А.-68 характерно углубленной проработкой основных концепций алгоритмич. языков, позволяющей с помощью небольшого числа независимых фундаментальных понятий точно описывать процесс выполнения программы. Различаются внешние (относящиеся к конструкциям программы) и внутренние (относящиеся к данным, в том числе к процедурам и именам) объекты. Аксиоматически вводятся отношения между внешними (E) и внутренними (I) объектами, напр. «E_1 содержит E_2», «E_1 идентифицирует E_2», «E обладает I», «I_1 именует I_2», «I_1 является компонентой I_2» и т. п. Выполнение программы описывается в терминах введенных отношений как функция разбора программы.

Особенностью синтаксиса А.-68 является его задание в виде двухступенчатой грамматики, когда порождающие правила А.-68 являются сами допустимыми текстами в нек-ром метаязыке, заданном своей порождающей грамматикой. Грамматич. правила А.-68 имеют, напр., вид:

список ПОНЯТИЙ разделенных РАЗДЕЛИТЕЛЕМ: ПОНЯТИЕ; список ПОНЯТИЙ разделенных разделителем, РАЗДЕЛИТЕЛЬ, ПОНЯТИЕ.

оператор присваивания ВИДА: выражение вырабатывающее имя ВИДА,:=, выражение вырабатывающее вид.

Слова, выраженные большими буквами,— это граммати. единицы метаязыка, порождающие правила для к-рых имеют, напр., вид:

ПОНЯТИЕ: идентификатор, оператор.

РАЗДЕЛИТЕЛЬ: запятая, точка с запятой.

ВИД: целый, логический, имя ВИДА.

Нек-рые понятия метаязыка, напр. ВИД, могут иметь бесконечное число порождений. Собственно порождающие правила А.-68 получаются систематич. заменой понятий метаязыка в грамматич. правилах на любое одно и то же их порождение. Результирующие правила в металингвистич. обозначениях алгола-60 выглядят, напр., так:

⟨список идентификаторов разделенных запятыми⟩ ::= ⟨идентификатор⟩|⟨список идентификаторов разделенных запятыми⟩⟨запятая⟩⟨идентификатор⟩

⟨оператор присваивания целого⟩ ::= ⟨выражение вырабатывающее имя целого⟩ := ⟨выражение вырабатывающее целое⟩.

Использование двухступенчатой грамматики позволяет, во-первых, сократить число однотипных порождающих правил и, во-вторых, выразить средствами синтаксиса атрибутную информацию понятий и нек-рые контекстные зависимости, к-рые в противном случае формулируются в виде содержательных ограничений.

Лит.: [1] Ван Вейнгаарден А. [и др.], Сообщение об алгоритмическом языке АЛГОЛ-68, «Кибернетика», 1969, № 6, с. 23—145; 1970, № 1, с. 13—160; [2] Линдси Ч., Мюйлен С., Неформальное введение в АЛГОЛ-68, пер. с англ., М., 1973. *А. П. Ершов.*

АЛГОРИТМ, алгорифм,— точное предписание, к-рое задает вычислительный процесс (называемый в этом случае алгоритмическим), начинающийся с произвольного исходного данного (из нек-рой совокупности возможных для данного А. исходных данных) и направленный на получение полностью определяемого этим исходным данным результата. А. являются, напр., известные из начальной школы правила сложения, вычитания, умножения и деления столбиком; в этих А. возможными результатами служат натуральные числа, записанные в десятичной системе, а возможными исходными данными — упорядоченные пары таких чисел. Вообще говоря, не предполагается, что результат будет обязательно получен: процесс применения А. к конкретному возможному исходному данному (т. е. алгоритмич. процесс, развертывающийся начиная с этого данного) может также оборваться безрезультатно (в этом случае говорят, что произошла безрезультатная остановка) или не закончиться вовсе. В случае, если процесс заканчивается (соответственно не заканчивается) получением результата, говорят, что А. применим (соответственно неприменим) к рассматриваемому возможному исходному данному. (Можно построить такой А. α, для к-рого не существует А., распознающего по произвольному возможному для α исходному данному, применим к нему α или нет; такой А. α можно, в частности, построить так, чтобы совокупностью его возможных исходных данных служил натуральный ряд.)

Понятие А. занимает одно из центральных мест в современной математике, прежде всего вычислительной. Так, проблема численного решения уравнений данного типа заключается в отыскании А., к-рый всякую пару, составленную из произвольного уравнения этого типа и произвольного положительного рационального числа ε, перерабатывает в число (или набор чисел), отличающееся (отличающихся) от корня (корней) этого уравнения меньше, чем на ε. Усовершенствование цифровых вычислительных машин дает возможность реализовать на них все более сложные А. Однако встретившийся в описывающей понятие А. формулировке термин «вычислительный процесс» не следует понимать в узком смысле только цифровых вычислений: уже в школьном курсе алгебры говорят о буквенных вычислениях, да и в арифметич. вычислениях появляются отличные от цифр символы (скобки, знак равенства, знаки арифметич. действий). Целесообразно, таким образом, рассматривать А., оперирующие с произвольными символами и их комбинациями. Простейшим случаем такой комбинации является линейная последовательность символов, образующая *слово*, однако можно рассматривать и «нелинейные» комбинации — такие, как алгебраич. матрицы, знакосочетания в смысле Н. Бурбаки (N. Bourbaki), фразы того или иного языка с расставленными стрелками синтаксич. управления и, вообще, размеченные тем или иным способом графы. Наиболее общее интуитивное понимание состоит в том, что исходными данными и результатами А. могут служить самые разнообразные *конструктивные объекты*. Это открывает возможность широкого применения понятия А. Можно говорить об А. перевода с одного языка на другой, об А. работы поездного диспетчера (перерабатывающего информацию о движении поездов в приказы) и др. примерах алгоритмич. описания процессов управления; именно поэтому понятие А. является одним из центральных понятий кибернетики.

Пример алгоритма. Пусть возможными исходными данными и возможными результатами служат всевозможные слова в *алфавите* $\{a, b\}$. Условимся называть переход от слова X к слову Y «допустимым» в следующих двух случаях (ниже P обозначает произвольное слово): 1) X имеет вид aP, а Y имеет вид Pb; 2) X имеет вид baP, а Y имеет вид $Paba$. Формулируется предписание: «взяв к.-л. слово в качестве исходного, делай допустимые переходы до тех пор, пока не получится слово вида aaP; тогда остановись, слово P и есть результат». Это предписание образует А., к-рый обозначим $\mathfrak{A}$. Возьмем в качестве исходного данного слово *babaa*. После одного перехода получим *baaaba*, после второго *aabaaba*. В силу предписания мы должны остановиться, результат есть *baaba*. Возьмем в качестве исходного данного слово *baaba*. Получим последовательно *abaaba*, *baabab*, *abababa*, *bababab*, *babababa*, ... Можно доказать, что процесс никогда не кончится (т. е. никогда не возникает слово, начинающееся с *aa*, и для каждого из получающихся слов можно будет совершить допустимый переход). Возьмем теперь в качестве исходного данного слово *abaab*. Получим *baabb*, *abbaba*, *bbabab*. Далее мы не можем совершить допустимый переход, и в то же время нет сигнала остановки. Произошла безрезультатная остановка. Итак, $\mathfrak{A}$ применим к слову *babaa* и неприменим к словам *baaba* и *abaab*.

Значение алгоритмов. А. встречаются в науке на каждом шагу: умение решать задачу «в общем виде» всегда означает, по существу, владение нек-рым А. Говоря, напр., об умении человека складывать числа, имеют в виду не то, что он для любых чисел рано или поздно сумеет найти их сумму, а то, что он владеет нек-рым единообразным приемом сложения, применимым к любым двум конкретным записям чисел, т. е., иными словами, А. сложения (примером такого А. и является известное правило сложения чисел столбиком). Понятие задачи «в общем виде» уточняется при помощи понятия массовая *алгоритмическая проблема* (м. а. п.). М. а. п. задается серией отдельных, единичных проблем и состоит в требовании найти единый А. их решения (когда такого А. не существует, говорят, что рассматриваемая м. а. п. неразрешима). Так, проблема численного решения уравнений данного типа и проблема *автоматического перевода* суть м. а. п.: образующими их единичными проблемами являются в 1-м случае проблемы численного решения отдельных уравнений данного типа, а во 2-м случае — проблемы перевода отдельных фраз. Ролью м. а.п. и определяется как значение, так и сфера приложения понятия А.: напр., а алгебре возникают м. а. п. проверки алгебраич. равенств различных типов, в математич. логике — м. а. п. распознавания выводимости предложений из заданных аксиом и т. п. (для математич. логики понятие А. существенно еще и потому, что на него опирается центральное для математич. логики понятие *исчисления*, служащее обобщением и уточнением интуитивных понятий «вывода» и «доказательства»). Установление неразрешимости к.-л. м. а. п. (напр., проблемы распознавания истинности или доказуемости для к.-л. логико-математич. языка) является важным познавательным актом, показывающим, что для решения конкретных единичных проблем данной серии принципиально необходимы специфические для каждой отдельной проблемы методы.

Содержательные явления, к-рые легли в основу образования понятия «А.», издавна занимали важное место в науке. С древнейших времен многие задачи математики заключались в поисках тех или иных конструктивных методов. Эти поиски, особенно усилившиеся в связи с созданием удобной символики, а также осмысления принципиального отсутствия искомых методов в ряде случаев (задача о квадратуре круга и подобные ей) — все это было мощным фактором развития научных знаний. Осознание невозможности решить задачу прямым вычислением привело к созданию в 19 в. теоретико-множественной концепции. Лишь после периода бурного развития этой концепции (в рамках к-рой вопрос о конструктивных методах в современном их понимании вообще не возникает) оказалось возможным в сер. 20 в. вновь вернуться к вопросам конструктивности, но уже на новом уровне, обогащенном выкристаллизовавшимся понятием А. Это понятие легло в основу конструктивного направления в математике.

Само слово «А.» происходит от algoritmi, являющегося, в свою очередь, латинской транслитерацией арабского имени хорезмийского математика 9 в. аль-Хорезми. В средневековой Европе А. назывались десятичная позиционная система счисления и искусство счета в ней, поскольку именно благодаря латинскому переводу (12 в.) трактата аль-Хорезми Европа познакомилась с позиционной системой.

Строение алгоритмического процесса. Алгоритмич. процесс есть процесс последовательного преобразования конструктивных объектов (к. о.), происходящий дискретными «шагами»; каждый шаг состоит в смене одного к. о. другим. Так, при применении А. $\mathfrak{A}$ к слову *baaba* возникают последовательно *baaba*, *abaaba*, *baabab* и т. д. А при применении, скажем, А. вычитания столбиком к паре ⟨307, 49⟩ последовательно возникнут такие к. о.:

$$\begin{array}{r}\_307\\ \underline{49}\\ \end{array}\quad \begin{array}{r}\_30\dot{7}\\ \underline{49}\\ 8\end{array}\quad \begin{array}{r}\_3\dot{0}\dot{7}\\ \underline{49}\\ 58\end{array}\quad \begin{array}{r}\_\dot{3}\dot{0}\dot{7}\\ \underline{49}\\ 258\end{array}$$

При этом в ряду сменяющих друг друга к. о. каждый последующий полностью определяется (в рамках данного А.) непосредственно предшествующим. При более строгом подходе предполагается также, что переход от каждого к. о. к непосредственно следующему достаточно «элементарен» — в том смысле, что происходящее за один шаг преобразование предыдущего к. о. в следующий носит локальный характер (преобразованию подвергается не весь к. о., а лишь нек-рая, заранее ограниченная для данного А. его часть, и само это преобразование определяется не всем предыдущим к. о., а лишь этой ограниченной частью).

Таким образом, наряду с совокупностями возможных исходных данных и возможных результатов для каждого А. имеется еще совокупность возможных промежуточных результатов, представляющая собой ту рабочую среду, в к-рой развивается алгоритмич. процесс. Для $\mathfrak{A}$ все три совокупности совпадают, а для А. вычитания столбиком — нет: возможными исходными данными служат пары чисел, возможными результатами — числа (все в десятичной системе), а возможные промежуточные результаты суть «трехэтажные» записи вида $\dfrac{\_p}{\underline{q}}{r}$, где q — запись числа в десятичной системе, r — такая же запись или пустое слово, а p — запись числа в десятичной системе с допущением точек над нек-рыми цифрами. Как правило, для данного А. можно выделить 7 характеризующих его (не независимых) параметров: 1) совокупность возможных исходных данных, 2) совокупность возможных результатов, 3) совокупность возможных промежуточных результатов, 4) правило начала, 5) правило непосредственной переработки, 6) правило окончания, 7) правило извлечения результата.

«Уточнения» понятия алгоритма. Понятие А. в его общем виде принадлежит к числу основных первоначальных понятий математики, не допускающих определения в терминах более простых понятий. Возможные уточнения понятия А. приводят, строго говоря, к известному сужению этого понятия. Каждое такое уточнение состоит в том, что для каждого из указанных 7 параметров А. точно описывается нек-рый класс, в пределах к-рого этот параметр может меняться. Выбор таких классов и отличает одно уточнение от другого. Поскольку 7 параметров однозначно определяют нек-рый А., то выбор 7 классов изменения этих параметров определяет нек-рый класс А. Однако такой выбор может претендовать на название «уточнения», лишь если имеется убеждение, что для произвольного А., имеющего допускаемые данным выбором совокупности возможных исходных данных и возможных результатов, может быть указан равносильный ему А. из определенного данным выбором класса А. Это убеждение формулируется для каждого уточнения в виде основной гипотезы, к-рая — при современном уровне наших представлений — не может быть предметом математич. доказательства.

Первые уточнения описанного типа предложили в 1936 Э. Л. Пост (E. L. Post, см. [5]) и А. М. Тьюринг (A. M. Turing, см. [3], [4]), их конструкции во многом предвосхитили идеи, заложенные в основу современных цифровых вычислительных машин. Известны также уточнения, сформулированные А. А. Марковым (см. [10], [11], *Нормальный алгорифм*) и А. Н. Колмогоровым (см. [12],[13]; последний предложил трактовать конструктивные объекты как топологич. комплексы определенного вида, что дало возможность уточнить свойство «локальности» преобразования). Для каждого из предложенных уточнений соответствующая основная гипотеза хорошо согласуется с практикой. В пользу этой гипотезы говорит и то, что, как можно доказать, все предложенные уточнения в нек-ром естественном смысле эквивалентны друг другу.

В качестве примера рассмотрим уточнение, предложенное А. М. Тьюрингом. В своем оригинальном виде это уточнение заключалось в описании нек-рой абстрактной вычислительной машины, состоящей из: 1) бесконечной ленты, разбитой на следующие друг за другом в линейном порядке ячейки, причем в каждой ячейке записан к.-л. символ из так наз. «внешнего алфавита» машины, и 2) каретки, находящейся в каждый момент в нек-ром «состоянии» (из заданного конечного списка состояний), способной перемещаться вдоль ленты и изменять содержимое ячеек; А. вычислений на такой машине («тьюрингов А.») задается в виде программы, управляющей действиями каретки. Более подробное и точное описание см. в статье *Тьюринга машина*; здесь приводится модернизированное изложение конструкции Тьюринга в терминах, указанных выше 7 параметров. Чтобы задать тьюрингов А., надо указать: а) попарно непересекающиеся алфавиты $Б$, $Д$, $Ч$ с выделенной в $Д$ буквой λ и выделенными в $Ч$ буквами α и ω, б) набор пар вида $\langle p\xi, \eta Tq\rangle$, где p, $q \in Ч$, ξ, $\eta \in Б \cup Д$, а T есть один из трех знаков $-$, 0, $+$, причем предполагается, что в этом наборе (наз. программой) нет 2 пар с одинаковыми первыми членами. Параметры А. задаются так: возможными исходными данными и возможными результатами служат слова в $Б$, а возможными промежуточными результатами — слова в алфавите $Б \cup Д \cup Ч$, содержащие не более одной буквы из $Ч$. Правило начала: исходное слово P переводится в слово $\lambda\alpha P\lambda$. Правило окончания: заключительным является промежуточный результат, содержащий ω. Правило извлечения результата: результатом объявляется цепочка всех тех букв заключительного промежуточного результата, к-рая идет вслед за ω и предшествует первой букве, не принадлежащей $Б$. Правило непосредственной переработки, переводящее A в A', состоит в следующем. Приписываем к A слева и справа букву λ; затем в образовавшемся слове часть вида $\varepsilon p\xi$, где $p \in Ч$, $\varepsilon\xi \in Б \cup Д$, заменяем на слово Q по следующему правилу: в программе ищется пара с первым членом $p\xi$; пусть второй член этой пары есть ηTq; если T есть $-$, то $Q = q\varepsilon\eta$; если T есть 0, то $Q = \varepsilon q\eta$; если T есть $+$, то $Q = \varepsilon\eta q$. Возникающее после этой замены слово и есть A'.

Лит. см. [3]—[5], [10]—[13] при ст. *Алгоритмов теория*.

В. А. Успенский.

АЛГОРИТМ В АЛФАВИТЕ A — «точное общепонятное предписание, определяющее потенциально осуществимый процесс последовательного преобразования абстрактных слов в *алфавите* A, процесс, допускающий любое слово в A в качестве исходного» (см. [1], с. 51).

А. в а. представляют собой частный случай общего понятия *алгоритма*. Исходными данными и возможными результатами применения А. в а. являются *конструктивные объекты* достаточно общего типа — *слова*, и это обстоятельство определяет роль понятия А. в а.