Г.В. Суходольский

АЛГОРИТМЫ И АЛГОРИТМИЧЕСКИЕ СТРУКТУРЫ[1]

Алгоритм — это правило или программа действий для решения какой-либо задачи. В классической математике алгоритмы назывались алгорифмами, но это устарело. Алгоритм как математический объект может быть задан в любой из пяти форм: словесной, символической, графической, табличной и аналитической. Математические функции, обозначенные символами, записанные в виде уравнений, все являются алгоритмами. Например, когда мы пишем tg, за этим символом тригонометрической функции скрывается правило её вычисления: взять отношение катета, противолежащего углу, к катету, ему прилежащему. А уравнение прямой y = ax + b в неявном виде описывает, как получить значения одной из функционально связанных переменных, умножая на константу значения другой переменной (аргумента) и суммируя с ещё одной константой. Этот предписательный смысл уравнения стал бы более явным, если бы его записывать так: xa + Ъ = y.

Следует различать однозначные и многозначные, нестохастические, полные и неполные, а также простые и сложные алгоритмы.

В однозначном алгоритме действия следуют друг за другом как причины и следствия до получения результата, ради которого был сконструирован алгоритм. Однозначные алгоритмы традиционно используются в математике и технике. По ним производятся вычисления функций и работают механизмы и машины. Такие алгоритмы вполне заслуживают названия «механистические».

Многозначные алгоритмы более сложны: они состоят из множества однозначных подалгоритмов, или реализаций, которые «включаются» при определённых условиях. Многозначные алгоритмы пригодны для решения задач определённого класса при более широком круге условий, чем однозначные алгоритмы. Обычно однозначные подалгоритмы представляют собой варианты, в совокупности приводящие к решению задачи. Поэтому можно представить многозначный алгоритм в виде множества однозначных вариантов и называть его «вариативным» алгоритмом. Вот характерный пример.

Со школьной скамьи и по справочникам всем известно, что $(a + +b)^2 = a^2 + 2ab + Ь^2$. Но если учесть, что (a + bf = (a+ Ъ) (a+ Ъ) и всевозможные перестановки порядка действий, то окажется: за этой привычной формулой скрываются 38 вариантов вычислений. Например: $(b + a)^2 = 62a + Ь^2 + a^2$. И я, наверно, не все варианты перечислил.

[1] *Суходольский Г.В.* Математические методы в психологии. 3-е изд., испр. Харьков: Изд-во «Гуманитарный центр», 2008.

Вообще говоря, следует понять, что любая «человеческая» задача вариативна, т.е. имеет много способов решения. К сожалению, они обычно не известны и не используются, как в примере с $(a + b)^2$. Даже для несложных задач один человек с трудом находит несколько возможных вариантов решения. А более или менее полное множество вариантов существует как чёрный ящик, который можно «просветлить» лишь большому числу работающих над ним исследователей[2].

По вариативным алгоритмам тоже работают машины — это компьютеры. В вариативных алгоритмах ещё сохраняется определённая доля механистичности: виды действия следуют друг за другом однозначно, машинообразно. Человеку, конечно, приходится работать по таким алгоритмам, но это однообразие не каждому и не всегда приносит удовлетворение. Ведь человек — не машина, и он непроизвольно разнообразит и жизнь, и труд.

Обычный вариативный алгоритм не стохастичен, он представляет собой фактор-множество вариантов. Однако по мере многократного исполнения человеком одни варианты реализуются чаще, чем другие. Это зависит и от условий применения алгоритма, и от субъективных знаний и склонностей человека-исполнителя. Важно, что в конце концов вариативный алгоритм превращается в мультимножество вариантов, в котором кратности имеют смысл частоты или частости применения вариантов. Так появляются стохастические алгоритмы.

Стохастический алгоритм проще всего представить как вариативный алгоритм, варианты которого реализуются и приводят к результату лишь с определённой вероятностью, а все вместе — достоверно. Можно сказать и так, что стохастический алгоритм задаётся распределением вероятностей вариантов решения задач данного класса.

Стохастический алгоритм может быть полным в том смысле, что его варианты образуют полную группу несовместных событий. Иначе он неполный. Я уже говорил, что перечислить все варианты решения даже простой задачи — дело трудное, если не невозможное. Поэтому многие и многие стохастические алгоритмы не полны. А значит, искомый результат уже не достигается достоверно, но лишь случайно, с большей или меньшей вероятностью, отличающейся от единицы. Такие неполные (частичные) стохастические алгоритмы называются эвристическими, или эвристиками.

Итак, мы выделили четыре типа алгоритмов — механистические, вариативные, стохастические и эвристические. По первым двум работают машины, два последних типа свойственны человеку. Рассмотрим простейшие примеры.

Пусть нужно переставить объект с одного места на другое. Однозначный (механистический) алгоритм решения этой задачи представляет собой цепочки действий: «взять — поднять — перенести — поставить — отпустить» (рис. 1, б, 1).

[2] К примеру, в конце 60-х гг. Арон Флейшман вместе со мной исследовал вариативность задачки из задачника для 4-го класса. Нам удалось независимо друг от друга найти по 4–6 способов решения. И только когда эти поиски в виде эксперимента «попробуй реши ещё каким-либо другим способом» (на гибкость мышления) были проведены в физико-математических классах, среди студентов, инженеров — человек 300, тогда было найдено 12 способов решения (от 4 действий до 7 действий).

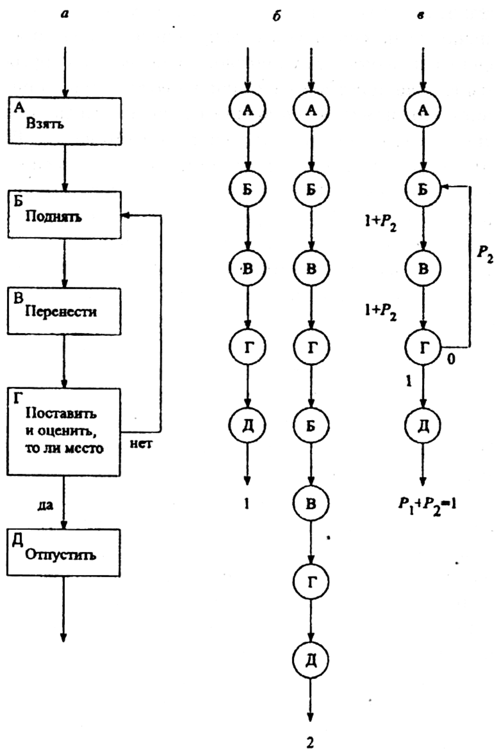

Рис. 1. Графы алгоритмов:

а — ЛСА (логическая схема вариативного алгоритма): А, Б, В, Г, Д — обозначения действий; б — две однозначные реализации вариативного алгоритма: 1 — без повторений, 2 — с однократным повторением; ***в*** — граф стохастического вариативного алгоритма, полученный обобщением взвешенных вероятностями P_1 и P_2 реализаций

Но может случиться, что первоначально намеченное место, куда переставлен объект, чем-то неадекватно и объект нужно снова перенести на другое место. В таком случае действие «поставить» сочетается с оценочным действием «то ли место»; в результате появляются две возможные оценки — «да», «место то», или «нет», «место не то». Вот, если «место не то», приходится снова поднимать объект, переносить его и ставить на новое место. Вообще говоря, это место снова оценивается и может не подойти субъекту, реализующему алгоритм.

Тогда вместо реализации с двумя переносами (рис. 1, б, 2) появится реализация с тремя, четырьмя и т.д. переносами. Но я для примера ограничусь двумя реализациями.

Будучи объединены, эти реализации образуют т.н. логическую схему алгоритма, широко распространённую в приложениях алгоритмического подхода (рис. 1, *а).* А если эти же реализации взвесить вероятностями и обобщить, то получится граф стохастического алгоритма (рис. 1, *в).*

Как было сказано, следует различать простые и сложные алгоритмы. Конечно, простота и сложность относительны. Однако многие реальные задачи требуют решений, состоящих из многих десятков, сотен, а то и тысяч «элементарных» действий («поднять», «поставить», «включить», «выключить» и т.д.). Таковы, в частности, задачи, решаемые в процессе работы операторами энергоблоков, инженерами-космонавтами. Ясно, что такие задачи, а тем более всю деятельность субъектов труда нельзя описать одним алгоритмом.

Так вот в качестве «простого» можно принять алгоритм, число действий в котором соответствует миллеровскому числу 7 ± 2. При большем количестве действий приходится расчленять задачу на подзадачи, решаемые простыми алгоритмами, и синтезировать из них сложные алгоритмы, которые целесообразно, по сути дела, называть «алгоритмическими структурами».

Алгоритмические структуры обычно иерархичны, и для их синтеза разработаны специальные процедуры, включающие в себя операции свёртки-развертки, взвешивания, обобщения и несколько видов соединений. Это достаточно объёмные процедуры, и они рассмотрены в других публикациях[2]. Здесь ограничимся рассмотрением простейшей алгоритмической структуры действий преподавателя в ситуации зачёта.

Эта ситуация хорошо знакома всем учившимся в вузе. Преподаватель задаёт вопросы, а студент на них отвечает — правильно или неправильно. И смотря по количеству правильных и неправильных ответов, преподаватель оценивает знания студента дихотомически — «зачёт» или «незачёт».

Опытный преподаватель не станет превращать зачёт в экзамен. Он будет задавать «ключевые» вопросы по курсу и ограничит их количество — минимально двумя, максимально тремя вопросами. Потому что два правильных ответа достаточны, чтобы поставить зачёт, так же как два неправильных ответа достаточны, чтобы зачёта не поставить. Неопределённость возникает, если студент на два вопроса даёт один правильный и один неправильный ответ. В этом случае необходим третий вопрос, по результатам ответа на который ставится «зачёт» или «незачёт».

Таким образом, имеется всего шесть вариантов действий и решений. Они в виде древовидного графа показаны на рис. 2. Замечу, что этот граф аналогичен дереву исходов, но алгоритмическая структура не является цепью случайных событий.

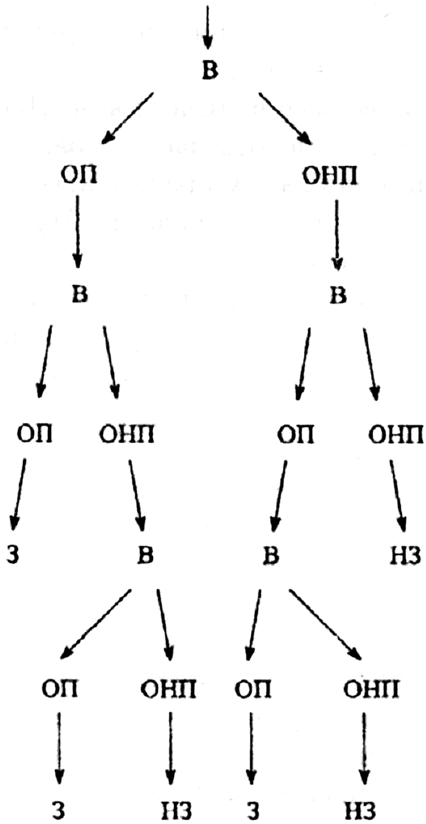

Рис. 2. Древовидный граф алгоритмической структуры зачёта, проводимого преподавателем:

В — вопрос преподавателя; ОП, ОНИ — ответ правильный или неправильный, даваемый студентом; 3, НЗ — оценки «зачёт» или «незачёт», которые ставит преподаватель

Нетрудно видеть, что образующими алгоритмическую структуру в данном случае

---

2 *Суходольский Г.В.* Структурно-алгоритмический анализ и синтез деятельности. Л., 1976; *Суходольский Г.В.* Математико-психологические модели деятельности. СПб., 1994; *Суходольский Г.В.* Введение в математико-психологическую теорию деятельности. СПб., 1998.

являются диады «вопрос—ответ», которые можно рассматривать в качестве элементарных алгоритмов. Их соединение в цепочки, управляемое характером ответов студента и оценками преподавателя, и образует данную алгоритмическую структуру. На рис. 2 она показана как вариативная нестохастическая. Но это не так.

Опытный преподаватель, озабоченный качеством и уровнем овладения студентами учебным материалом, будет записывать индивидуальные реализации вариантов и в конце зачёта получит мультимножество результатов, в котором варианты и образуют фактор-множество, а кратности позволяют оценить относительные частоты оценок «зачёт» и «незачёт», в том числе полученных за два и за три вопроса.

Выпишем это мультимножество в общем виде:

(P_1 ВОП, .ВОП, З;

P_2 ВОП, ВОНП, ВОП, З;

P_3 • ВОНП, ВОП, ВОП, З;

P_4 • ВОНП, ВОНП, НЗ;

P_5 • ВОП, ВОНП, ВОНП, НЗ;

PI • ВОНП, ВОП, ВОНП, НЗ).

Конечно, сумма этих шести вероятностей равна единице. Сумма трёх первых: $P(З) = P_1 + P_2 + P_3$ определяет общую успеваемость группы (курса). Сумма трёх последних: $P(НЗ) = P_4 + P_5 + P_6$ заставляет задуматься над дефектами обучения либо научения. Сумма $P_2 + P_3$ характеризует долю студентов, плохо освоивших 1/3 учебного материала, а сумма $P_5 + P_6$ — долю студентов, не усвоивших 2/3 необходимых знаний. Таким образом, количественно анализируя итоги реализации данной алгоритмической структуры, можно получить полезную информацию.

Редакция благодарит Издательство «Гуманитарный центр» (г. Харьков, Украина) за предоставленные материалы (Н. Томашек, Г. Суходольский).