

УДК 510.51+519.212

В. И. ШИНКАРЕНКО - д.т.н, профессор, кафедра КИТ, Днепропетровский национальный университет железнодорожного транспорта имени академика В. Лазаряна
Т. Н. ВАСЕЦКАЯ – ассистент, кафедра КИТ, Днепропетровский национальный университет железнодорожного транспорта имени академика В. Лазаряна,
tetyana@vasetsky.com

О ДУАЛИЗМЕ СЛУЧАЙНОГО И ДЕТЕРМИНИРОВАННОГО В АЛГОРИТМАХ

Введение

Подобно тому, как свет проявляет свойства и частицы и волны, любая величина, являющаяся результатом измерений, проявляет свойства случайной и детерминированной одновременно.

При подбрасывании монеты выпадение орла или решки, с одной стороны детерминировано и неуклонно подчиняется законам физики. Исход события строго предопределен рядом факторов, таких как начальная скорость и ускорение, момент вращения, эксцентриситет монеты, угол вылета, наклон плоскости падения, материал и свойства поверхности падения, скорость движения воздуха во всех слоях падения и т.п.

С другой стороны, факторов слишком много и их значения неизвестны и трудноизмеримы, и можно считать исход такого события случайным, что обычно и делается в теории вероятностей.

Даже, казалось бы, такая величина как количество студентов в группе, являясь с одной стороны совершенно детерминированной, с другой – ее можно рассматривать как случайную. В конкретных условиях значение такой величины вполне определенное, в то же время оно предопределено целым рядом случайных факторов. Количество студентов в группе зависит от приема студентов (престижности ВУЗа, престижности специальности, количества мест, количества абитуриентов, престижности других ВУЗов и специальностей и т.п.), отсева (потенциальных возможностей студентов,

отношения их к работе, влияния кураторов, семьи и администрации и т.д.) и присева (также множество факторов).

Случайность проявляется в силу большого числа слабо контролируемых факторов, которые в некоторой степени могут компенсировать друг друга.

При разработке алгоритмов, их выполнении очень важно понимать какой результат будет получен, и какие факторы могут на него повлиять. Очень важно понимать природу алгоритмов.

Объектом данных исследований является проявление случайности и детерминированности алгоритмов.

Целью данной работы является показать проявление дуализма в алгоритмах и программах и необходимость четкого понимания этого во избежание трудно устранимых ошибок.

Детерминированность алгоритмов и вычислительных устройств

Любой алгоритм реализует некоторую функцию (но не для любой функции существует алгоритм, ее реализующий) [1]:

$$Y = F(X), \quad (1)$$

где $X \in X^*$ и $Y \in Y^*$ являются элементами некоторых множеств.

Детерминированность или определенность является одним из основных свойств алгоритмов [2, 3]. Для любого $X \in X^*$ и при любом выполнении алгоритма будет выполнена одна и та же последовательность действий и получен один и тот же результат.

Естественно это свойство алгоритмов поддерживается всеми вычислительными устройствами как абстрактными, так и реальными. Известная машина Тьюринга [1, 4], например, имеет команды вида

$$T_{ij} : q_i a_j \rightarrow q_k a_l X, \quad (2)$$

где $T_{i,j}$ – текущая команда; $q_i a_j$, a_j – текущее состояние машины и символ в текущей ячейке ленты соответственно; $q_k a_l$ – состояние, в которое перейдет машина и символ, который будет записан в текущую ячейку, в результате выполнения команды; X – выбор следующей ячейки (которая станет текущей в результате выполнения команды) посредством перехода к следующей, предыдущей или текущей ячейке.

Если имеется определенным образом заполненная символами лента, машина Тьюринга находится в определенном начальном состоянии q_0 и определена текущая ячейка, согласно первой команде вида (2) будут однозначно определены новые состояния, текущая ячейка и содержимое ленты.

Методом математической индукции не сложно показать, что любая программа из команд вида (2) при определенных начальных условиях получит строго определенный результат.

Это же касается и других абстрактных машин Поста, Колмогорова, Шёнхаге [1, 5, 6] и т.п.

В реальных ЭВМ, к примеру на процессорах Intel, все команды обработки данных имеют входные и выходные данные, и последние точно и однозначно определяются входными.

Подытоживая, отметим, что алгоритм детерминировано, точно, определенно представляет некоторую функцию вида (1).

Случайность и алгоритмы

Однако, существует класс стохастических алгоритмов. Как этот факт согласуется с детерминированностью алгоритмов? Каким образом результаты детерминирован-

ных алгоритмов могут быть случайными? Единственной возможностью является случайное представление входных данных. Тогда результат функции (1) (реализуемой алгоритмом) над случайными входными данными будет тоже случайным.

Ввиду того, что как показано во введении данные можно рассматривать как случайные так и детерминированные, рассмотрим три аспекта проявления дуализма случайного и детерминированного с точки зрения практики программирования.

Аспект первый. Входные данные алгоритмов скорее детерминированные, чем случайные. На вход программы (как реализации алгоритма) подаются вполне определенные данные. Как показано ранее, при любом выполнении алгоритма с такими данными должны быть получены вполне определенные и одинаковые выходные. Однако, это не всегда так. Причиной могут быть неинициализированные переменные, которые принимают некоторое случайное значение, уже содержащееся в оперативной памяти.

При этом "срабатывают" общепринятые заблуждения. Многие считают, что входом алгоритма, являются данные известные до его выполнения, программы – вводимые данные, процедуры – параметры. Даже Д. Кнут [2] считает, что алгоритм может не иметь входов (входных данных). На самом деле входом алгоритма являются все данные образующие элемент $X \in X^*$, которые являются аргументами функции (1). И не важно эти данные являются параметрами, локальными данными процедуры, глобальными данными процесса, в рамках которого выполняется процедура, либо другого прикладного или системного процесса. Неважно также, имеются ли эти данные к началу выполнения программы (или алгоритма) или получены в процессе ее выполнения от внешних устройств: клавиатуры, модема, внешнего накопителя информации, таймера или, скажем, датчика температуры накопителя на магнитных дисках.

Входными данными алгоритмов (и программ) могут быть и результаты выполнения других алгоритмов, и не обязательно выполненных до начала выполнения рассматриваемого. Так, при вводе данных с внешних устройств (например, клавиатуры) в процессе выполнения программы входные данные являются результатом работы драйверов устройств, которые кроме пересылки выполняют и преобразование данных. Пример, процедура ввода `read` языка программирования ПАСКАЛЬ предоставляет входные данные для других программ.

В таком контексте, неинициализированные переменные (как, впрочем, и инициализированные), если они влияют на результаты программы, следует считать входными данными соответствующих алгоритмов. И, если результаты программы неустойчивы, по указанной причине, неинициализированные данные следует рассматривать как случайные входные данные.

Аспект второй. Если входом некоторого алгоритма являются результаты подбрасывания монеты, количество студентов в группе и подобные данные, которые можно считать как детерминированными, так и случайными, результат его выполнения также можно рассматривать двояко. Как функцию детерминированной величины – детерминированным, а как функцию случайной – случайной.

В качестве примера можно привести алгоритм расчета средней успеваемости студентов в группе по результатам сессии. Оценки вполне определенные величины и результат алгоритма детерминированный. Однако, каждую оценку можно рассматривать как случайную величину, хотя бы в силу ее субъективности и случайности выбора экзаменационного билета. Следовательно, и результат алгоритма – среднюю оценку можно рассматривать как случайную.

Другим примером является алгоритм, реализующий метод анализа иерархий (МАИ) [7, 8]. С одной стороны данный алгоритм будет детерминированным, так как

оперирует с оценками эксперта, которые являются вполне определенными числами. Но с другой стороны, эти оценки являются результатом случайности, так как возникают как результат жизненного опыта эксперта [9]. Таким образом, и результат МАИ можно рассматривать как случайную величину.

Особенно следует отметить совокупность выполнения алгоритма в составе программных средств в конкретных условиях функционирования. Даже при детерминированном представлении X для каждого конкретного выполнения, множество $X_i \in X^*$ для совокупности выполнений алгоритма можно считать множеством случайных X_i . Знание вероятностных или статистических характеристик входных данных позволяет получить частотные характеристики выполнения всех шагов алгоритма и его временные характеристики [10].

Возможность вероятностного подхода в этом случае позволяет решать практические задачи программирования, а именно повышения качества программных средств путем выбора и разработки эффективных алгоритмов на основании оценки их временных характеристик.

Аспект третий. Стохастические алгоритмы, реализующие методы: случайного поиска, генетические и т.п., предполагают использование случайных величин. В силу детерминированности алгоритмов, сами такие величины они выработать не могут. Единственный выход: использование случайных значений некоторых измерений. В современных ЭВМ чаще всего случайным исходным измерением выбирается время.

Учитывая массовые потребности в случайных величинах, практически все современные средства разработки программ, имеют процедуры генерации случайных чисел (ГСЧ). В некотором смысле, ГСЧ подобен драйверу устройства. Он считывает значения времени и преобразует в последовательность случайных чисел.

Для корректного применения процедур ГСЧ необходимо знать их особенности. В первую очередь это касается того, насколько соответствуют реализуемые случайные величины декларируемым законам распределения и насколько велика неповторяемая последовательность случайных чисел.

Конечно же, разработчики подобных ГСЧ на это обращают особое внимание. К примеру, проведенный численный эксперимент с генератором случайных чисел в Delphi 6 показал, что в последовательности из 107 чисел типа Double (и Int64), не оказалось не только повторяющихся подпоследовательностей, но и даже одинаковых чисел.

Однако, как разработчики, так и программисты не всегда обращают внимание на источник входных данных для ГСЧ, что может привести к неустойчивым ошибкам в программном обеспечении.

Так, в Delphi 6 исходными данными для ГСЧ являются часы реального времени, значение которых обновляется 1024 раза в секунду, что может соответствовать порядку $10^3 \dots 10^4$ тактов процессора. При этом управление может получить несколько процессов. И если в этот промежуток времени они начнут строить последовательности случайных чисел они окажутся абсолютно одинаковыми.

```
const N=1000;  
type tm=array[1..N] of integer;  
var x,y:tm;  
procedure sl(var x1:tm);  
begin  
  Randomize;  
  for i:=1 to N do  
    x1[i]:= random(32000);  
end;  
begin  
  sl(x);  
  sl(y);  
end;
```

Рис 1. Фрагмент программы

Выполнен численный эксперимент. В программе на ПАСКАЛе (рис. 1) дважды вызывается процедура генерации последовательности 1000 случайных чисел [11]. В Delphi 6 в 99,8 % случаях выполнения программы последовательности оказываются абсолютно одинаковыми.

В Delphi 7 в таких случаях 100 % – последовательности различны. Разработчики ГСЧ изменили источник исходных данных для ГСЧ. С этой версии используется появившийся в архитектуре процессора Intel Pentium регистр-счетчик тактов. Надо заметить, на устаревших процессорах, не имеющих такого регистра эффект повторения последовательностей случайных чисел по-прежнему возможен.

Выводы

Детерминированность алгоритмов является их неотъемлемым свойством. Случайные результаты выполнения алгоритмов могут быть обусловлены лишь случайным представлением входных данных.

Правильное понимание вероятностных и детерминированных сторон алгоритмов позволяют избежать некоторых ошибок программирования и адекватно использовать их при разработке и исследованиях алгоритмов.

Библиографический список

1. Успенский, В. А., Теория алгоритмов: основные открытия и приложения: научное издание [Текст] / В. А. Успенский, А. Л. Семенов. - М.: Физ.-мат. лит.: Наука, 1987. – 288 с.
2. Кнут, Д. Э. Искусство программирования., том 1. Основные алгоритмы. [Текст] / Д. Э. Кнут. - М.: Издательский дом «Вильямс», 2000. – 720 с.
3. Цейтлин, Г. Е. Введение в алгоритмику.[Текст] / Г. Е. Цейтлин. - Киев: Сфера, 1998. – 310 с.
4. Игошин, В. И. Математическая логика и теория алгоритмов. [Текст] /

- В. И. Игошин. - Саратов: Изд-во Саратов. ун-та, 1991. – 256 с.
5. Колмогоров, А. Н. Теория информации и теория алгоритмов. [Текст] / А. Н. Колмогоров. - М.: Наука, 1987. – 304 с.
 6. Вьюгин, В. В. Колмогоровская сложность и алгоритмическая случайность (учебное пособие). [Текст] / В. В. Вьюгин. - М.: 2012. - 131 с.
 7. Саати, Т. Л. Принятие решений. Метод анализа иерархий. [Текст] / Т. Л. Саати. - М.: Радио и связь, 1989. – 316 с.
 8. Thomas L. Saaty Relative Measurement and Its Generalization in Decision Making Why Pairwise Comparisons are Central in Mathematics for the Measurement of Intangible Factors The Analytic Hierarchy/Network Process [Text] / T. L. Saaty // Royal Academy of Sciences, Spain, Series A. Mathematics, 2008 - November - С. 251-318.
 9. Саати, Т. Л. Об измерении неосознаемого. Подход к относительным измерениям на основе главного собственного вектора матрицы парных сравнений / Т. Л. Саати // Журнал "Cloud Of Science". 2015. - Т. 2 - № 1.
 10. Шинкаренко, В. И. Методология измерения временной эффективности программных средств [Текст] / В. И. Шинкаренко. // Теоретичні та прикладні аспекти побудови програмних систем: Тези доповідей. - К.: НаУКМА, 2005. - С. 61-65.
 11. Фленов, М. В. Библия Delphi 3-е издание [Текст] / М. В. Фленов – С-Петербург: БХВ-Петербург, 2012 – 688 с.

Ключові слова: детермінізм, випадковість, алгоритм, програма, вхідні дані, результат, дуалізм, випадкові числа.

Ключевые слова: детерминизм, случайность, алгоритм, программа, входные данные, результат, дуализм, случайные числа.

Keywords: determinism, randomness, input data, result, dualism, random digits.

Рецензенти:

к.т.н. доц. В. А. Андрющенко,
д.ф-м.н. проф. В. Е. Белозеров.

Поступила в редколлегию 07.04.2015.
Принята к печати 28.04.2015.