

УДК 004.421

МЕТОДИКА АЛГОРИТМИЗАЦИИ ЗАДАЧ РЕШАЕМЫХ НА ЭВМ

Долгов А. И., Преснухин В.В., Шихов Д.В.

«Ростовский военный институт РВ им. М. И. Неделина», г. Ростов-на-Дону

E-mail: ity79@mail.ru

Аннотация. Предлагаемая методика впервые представляет собой главным образом систему методов с использованием которых может быть получен требуемый результат в алгоритмизации прикладных задач и предназначена для широкого круга инженерно-технических работников, не специализирующихся в области компьютерной техники, но привлекаемых к разработке прикладных задач, решаемых на ЭВМ в интересах автоматизации самых различных видов профессиональной деятельности.

Ключевые слова: алгоритм, процесс алгоритмизации, моделируемый объект, метод интеграции, метод дифференциации.

Введение

Появление средств автоматизации немислимо без разработки соответствующего математического обеспечения, важнейшую составную часть которого представляют алгоритмы прикладных задач. Являясь результатом труда огромной армии алгоритмистов, в настоящее время алгоритмы стали не только объектом непосредственной разработки и реализации в конкретных технических и информационных системах, но и предметом изучения во многих учебных заведениях, а также объектом продолжающихся с момента появления первых электронных вычислительных машин (ЭВМ) исследований и научных публикаций.

Благодаря бурному развитию компьютерной техники, внедрению её в разнообразные области человеческой деятельности, резкому повышению уровня компьютерной грамотности населения и появлению огромного количества персональных ЭВМ (ПЭВМ), используемых не только в офисах, но и в быту, алгоритмизация прикладных задач перестала быть уделом профессиональных математиков, и сейчас в роли алгоритмистов выступают специалисты самых различных, практически любых профессий.

Сталкиваясь с необходимостью разработки алгоритмов, эти специалисты ощущают потребность в литературе методического характера. Что касается методики разработки и описания алгоритмов, а именно это, как правило, в наибольшей степени соответствовало бы потребностям большинства специалистов, обращающихся к литературе по алгоритмизации, то соответствующие материалы в публикациях либо вообще не приводятся, либо представлены настолько поверхностно, что не могут служить методическим руководством для практической разработки и описания алгоритмов. И это не удивительно, если учесть что алгоритмизация является творческим процессом, формализация которого представляет собой исключительно сложную проблему.

В данной статье излагается методика алгоритмизации прикладных задач, рассматривается сущность и содержание основных этапов этого процесса. Главное внимание уделяется двум предлагаемым принципиально различным методическим подходам к построению математических моделей как важнейшего этапа алгоритмизации, названных методами дифференциации и интеграции (состояний моделируемых объектов). Метод интеграции соответствует макроподходу к анализу рассматриваемого в задаче объекта (процесса) в целом, когда интересуются лишь его внешним поведением. В отличие от этого, метод дифференциации соответствует микроподходу, когда рассматривается внутренняя структура объекта (процесса), поведение которого описывается через поведение составляющих его взаимосвязанных элементов.

Предлагаемая методика алгоритмизации приводит к получению описания алгоритмов на языке публикаций с использованием языка высокого уровня типа Паскаль, но без доведения получаемых описаний до конкретных машинных программ и их отладки.

Основные этапы алгоритмизации

Алгоритмизация – это процесс разработки и описания алгоритма решения задачи.

В наиболее общем случае алгоритмизация включает следующие основные этапы:

- ◆ строгая постановка задачи;
- ◆ построение математической модели рассматриваемого в задаче объекта или процесса;
- ◆ построение алгоритма решения задачи.

Все этапы являются этапами формализации задачи, при этом этапы построения математической модели и построения алгоритма представляют собой этапы собственно разработки алгоритма.

Ниже излагается сущность основных этапов алгоритмизации задач, решаемых на ЭВМ.

1. Строгая постановка задачи

Профессиональная разработка строгой постановки задачи осуществляется прикладными математиками (алгоритмистами) на основании содержательного описания, включающего краткую содержательную формулировку задачи, официально оформляемых, как правило, в составе технического задания на разработку задачи, решаемой на ЭВМ. Краткая содержательная формулировка представляет собой основу для строгой постановки задачи.

Строгая постановка – формулировка задачи, определяющая все входные и выходные переменные, а также основные исходные допущения и ограничения.

Разработку строгой постановки задачи рекомендуется осуществлять в следующей последовательности:

- ◆ выявление состава выходных переменных (определяется характером задачи и потребностями конкретных лиц, заинтересованных в её решении);
- ◆ принятие исходных ограничений (вводятся, исходя из специфики и удобств применения задачи, решаемой на ЭВМ.);
- ◆ определение состава входных переменных и исходных допущений;
- ◆ формулирование строгой постановки задачи (т.е. совокупность значений выходных и входных переменных и перечень основных исходных допущений и ограничений).

Рассмотрим более подробно сущность процесса разработки определения состава входных переменных и исходных допущений. Данная стадия разработки строгой постановки задачи является наиболее трудной и ответственной. Это обусловлено тем, что свойства реальных объектов зависят от бесконечно большого количества факторов. Учесть все эти факторы, а тем самым определить все входные переменные в строгой постановке задачи и в последующем, при построении математической модели рассматриваемого в задаче объекта или процесса, невозможно. Это обстоятельство приводит в необходимости учёта ограниченного количества наиболее существенных факторов, отражающих самые важные для решаемой задачи свойства реальных объектов или процессов. Остальными факторами, а следовательно, и свойствами реальных объектов следует пренебречь введением соответствующих допущений.

Допущения – это предположения об упрощении реального объекта или процесса, вводимые при формализации задачи.

В подавляющем большинстве случаев специалисты, занимающиеся разработкой строгой постановки задачи, полагаются на свой опыт и интуицию, и практика свидетельствует о том, что глубокие и всесторонние знания процессов, подлежащих автоматизации, позволяют правильно определить состав входных переменных и исходных

допущений. При этом используется справочная литература, привлекаются результаты, полученные с помощью ранее разработанных математических моделей, проводятся эксперименты, экспертные опросы и т.д.

В случаях, когда строгая постановка задачи эвристическим способом (т.е. на основе опыта и интуиции) встречает непреодолимые субъективные затруднения (недостаточны или противоречивы знания о реальных объектах и процессах, не хватает опыта, подводит интуиция и т.д.), можно воспользоваться эмпирическим способом, с использованием методов многомерного статистического анализа (факторный анализ, дисперсионный анализ, корреляционный анализ, регрессионный анализ и т.д.).

Достоинством указанных методов статистического анализа при всех их недостатках, выражающихся, главным образом, в трудоёмкости, является то, что они дают принципиальную возможность научного обоснования постановок задач на основе объективного определения значимости входных переменных.

2. Построение математической модели

Построение (разработка) математической модели сводится к установлению математических и логических зависимостей, отражающих физическую сущность рассматриваемого в задаче объекта (процесса) и позволяющих вычислять значения всех выходных переменных величин, указанных в строгой постановке задачи, при заданных конкретных значениях входных переменных.

К разрабатываемым моделям должны быть предъявлены два основных требования – достаточная содержательность и по возможности наибольшая (разумная) простота.

Требование содержательности означает необходимость правильного, в достаточной степени достоверного (адекватного) отображения в модели наиболее существенных свойств моделируемых объектов (процессов) и их элементов. Разрабатываемые модели должны соответствовать положениям действующих руководящих документов, базироваться на ранее накопленном опыте, при (желательном) всестороннем и глубоком изучении последних достижений теории и практики, в том числе и за рубежом.

Требование простоты означает необходимость использования при разработке моделей минимального по сложности математического аппарата, но достаточного для удовлетворения требования к содержательности моделей. Соблюдение требования простоты способствует повышению оперативности разработки и решения задач, облегчает их изучение и использование в практической деятельности.

Профессиональная разработка математических моделей осуществляется прикладными математиками (алгоритмистами) в тесном контакте со специалистами той области деятельности, для автоматизации которой применяется ЭВМ, и представляет собой весьма сложный и ответственный этап алгоритмизации, так как требования содержательности и простоты противоречивы по своему существу. В самом деле, стремясь удовлетворить требование содержательности, разработчик пытается учесть в модели как можно точнее большее количество факторов. Математические соотношения для определения значений выходных переменных оказываются зависимыми от большого количества аргументов, модель становится сложной. С другой стороны, желание удовлетворить требование простоты побуждает разработчика к упрощению привлекаемого математического аппарата, что ведёт к более грубому учёту реальных свойств объектов (процессов), а, в конечном счёте, к снижению содержательности модели.

Искусство разработчика как раз и состоит в том, чтобы на основе диалектического подхода добиться разумного компромисса – не переусложнить модель и в то же время не выхолостить существа реального объекта или процесса.

Построение математической модели выполняется в два этапа:

- 1) декомпозиция моделируемого объекта или процесса на элементы;
- 2) синтез математических моделей элементов и объекта или процесса в целом.

Декомпозиция и синтез представляют собой творческие процессы, для эффективной реализации которых могут быть использованы излагаемые ниже методические рекомендации.

2.1. Декомпозиция моделируемого объекта

Декомпозиция моделируемого объекта представляет собой выделение на основе проводимого анализа в рассматриваемом объекте (или процессе) групп однотипных моделируемых элементов, свойствами которых (с учётом принимаемых допущений) обусловлены значения выходных переменных, указанных в строгой постановке задачи.

В наиболее развёрнутом виде декомпозиция моделируемого объекта (процесса) сопровождается оптимизацией и кроме выделения элементов модели включает:

- ◆ построение исходных графов состояний моделируемых элементов объекта (процесса);
- ◆ оптимизация графов состояний с целью их упрощения и обеспечения (в случае возможности) уменьшения количества состояний, учитываемых при дальнейшей алгоритмизации;
- ◆ выявление избыточных графов, которые могут не учитываться при построении математической модели;
- ◆ выделение опорных графов, которые должны учитываться при построении математической модели.

Графы должны отображать учитываемые при построении модели состояния элементов и возможные переходы из одних состояний в другие.

Состояния элементов – это абстрактные аналоги их реальных состояний, различимых при моделировании.

Состояния определяются, исходя из свойств реальных элементов, определяемых перечнем входных и выходных переменных, предусмотренных строгой постановкой задачи, и принятыми допущениями.

Графы состояний наряду с основными допущениями всегда отражают большее количество неоговоренных допущений. Поэтому графы состояний должны согласовываться со специалистами, в интересах которых разрабатывается задача, а основные из них обязательно приводиться в документации.

2.2. Синтез математических моделей

Синтез математических моделей представляет собой составление математических и логических зависимостей, описывающих рассматриваемые элементы моделируемого объекта (процесса) и их взаимосвязь (взаимодействие) друг с другом.

Результатом синтеза является получение математической модели рассматриваемого в задаче объекта или процесса.

Место синтеза математических моделей элементов и рассматриваемого объекта или процесса в целом в общем процессе алгоритмизации иллюстрируется рис. 2.6.

Синтез может осуществляться двумя принципиально различными методами, которые могут применяться и в сочетании. Один из них в дальнейшем называется методом интеграции, а другой – методом дифференциации. Соответственно этим двум методам разработанную на основе полученной математической модели задачу, решаемую на ЭВМ, принято называть методикой машинного расчёта (ММР), либо методикой машинного моделирования (МММ).

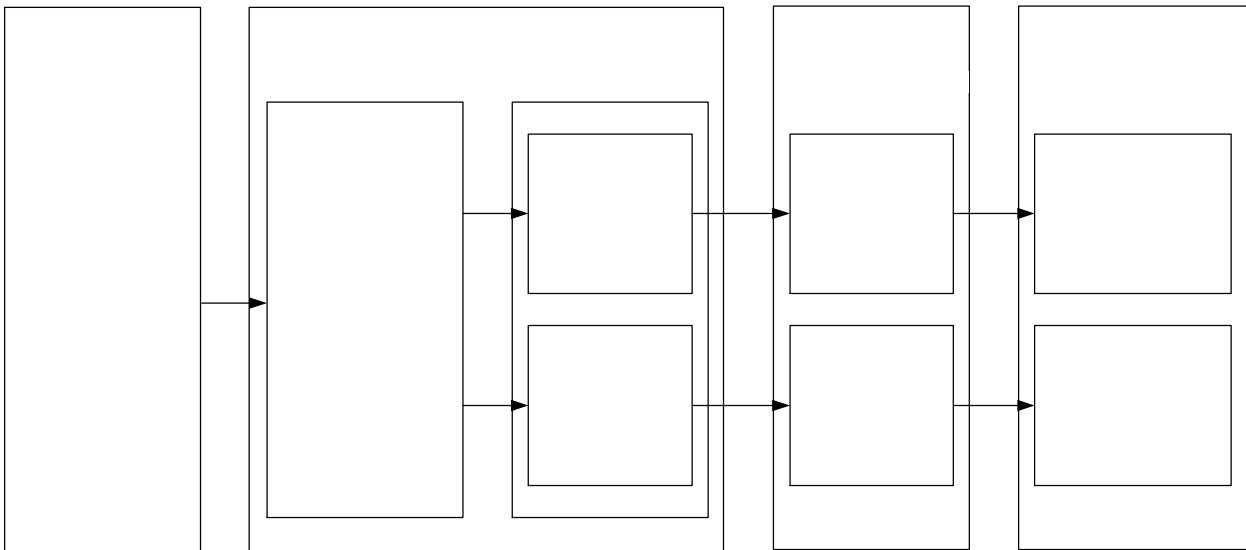


Рис. 2.1 Содержание процесса алгоритмизации

Метод дифференциации предполагает отыскание математических и логических зависимостей, описывающих развитие моделируемого процесса (поведение объекта) через смену состояний его элементов в последовательные моменты времени. При данном методе в качестве промежуточных переменных используются обозначения состояний элементов (применяются кодирование состояний), при этом состояние любого элемента в очередной момент времени описывается функцией его состояния и состояний других элементов в предшествующий момент времени.

В отличие от этого, при использовании метода интеграции в состав промежуточных переменных обозначения состояний элементов не включают. При этом методе синтез математической модели сводится к отысканию аналитических зависимостей, описывающих непосредственную взаимосвязь между входными, промежуточными и выходными переменными, наиболее часто в виде непрерывных функций.

Особенности синтеза с использованием каждого метода целесообразно пояснить в отдельности.

2.2.1. Метод интеграции

При поиске математических и логических зависимостей используется весь доступный арсенал методов математики, а также специальных наук, к области которых относятся рассматриваемые в задаче объекты или процессы.

Здесь следует иметь в виду, что многие из требуемых зависимостей уже имеют достаточно удобное и проверенное практикой математическое описание, поэтому при синтезе моделей широко используют справочники, учебники, монографии, научные статьи, отчёты о научно-исследовательских работах и другие публикации. Стремясь вписаться в типовой аппарат математики и других наук, широко прибегают к выявлению аналогий (типа электрический колебательный контур – механический маятник), имеющих одинаковое математическое описание.

Отыскивая математические и логические зависимости, преследуют цель учесть все выходные и входные переменные, указанные в строгой постановке задачи, при этом поступают следующим образом.

Если для какой-либо выходной переменной выявлено несколько различных зависимостей, то надо сравнить их по сложности и точности вычислений, по составу учитываемых в них переменных и выбрать наиболее подходящую.

Может оказаться, что для какой-либо выходной переменной не удаётся найти подходящую зависимость, при этом остаётся возможность перехода к построению модели методом дифференциации.

Среди учитываемых переменных могут оказаться не указанные в качестве входных в строгой постановке задачи. В этом случае следует либо откорректировать строгую постановку задачи дополнением состава входных переменных, либо принять соответствующие переменные за промежуточные.

Для промежуточных переменных поиск математических и логических зависимостей осуществляется таким же образом, как и для выходных, при этом могут возникать новые промежуточные переменные или (что менее желательно) производиться новые коррекции строгой постановки задачи, которые целесообразно каждый раз согласовывать с составителями технического задания на разработку задачи, решаемой на ЭВМ.

В ряде случаев, в частности для задач, реализуемых в реальном масштабе времени, нельзя найти математические соотношения, определяющие значения выходных и промежуточных переменных по значениям поступающих входных данных без явного учёта предыстории процесса, выражаемой корректируемыми значениями промежуточных показателей, учитывающих состояния моделируемых элементов. В таких случаях синтез модели методом интеграции может оказаться неосуществимым.

Если же найденные зависимости приводят к формулам, в которых промежуточные и выходные переменные выражаются в непосредственном виде, через входные переменные, то применяют методы тождественно-эквивалентных преобразований в интересах максимального упрощения синтезируемой математической модели.

В ходе построения модели могут приниматься новые допущения, при этом иногда возникает своеобразный итерационный процесс: как только разработчик при построении модели "заходит в тупик", он вводит дополнительные допущения, несколько упрощающие задачу, и снова пытается построить модель (или её отдельные части).

Если, пользуясь описанными приёмами, все же для каких-либо выходных или промежуточных переменных найти требуемые математические и логические зависимости не удастся, а возможности поиска исчерпаны, то можно попытаться прибегнуть к методам математической статистики.

Вместе с тем следует иметь в виду, что существенные трудозатраты на реализацию всех этих методов математической статистики в ряде случаев оказываются своеобразной расплатой либо за отсутствие требуемых знаний о рассматриваемом в задаче объекте (процессе), позволяющих выбрать для его описания подходящий математический аппарат, либо за то, что соответствующий математический аппарат ещё не создан.

Если, применяя все указанные приёмы, удастся найти расчётные соотношения для вычисления значений всех выходных и промежуточных переменных непосредственно через входные переменные, то нет никаких препятствий для перехода к построению алгоритма машинного расчёта (см. рис. 2.1). В ином случае, когда необходимые расчётные соотношения найти не удастся и возможности поиска исчерпаны, следует попытаться синтезировать математическую модель методом дифференциации.

2.2.2. Метод дифференциации

Метод дифференциации, применяется для задач, к которым не применим метод интеграции, но часто он может быть использован также и для задач, для которых метод интеграции применим.

При данном способе синтеза сначала формулируются условия смены состояний моделируемых элементов, соответствующие ранее построенным графам, после чего осуществляется построение математических и логических зависимостей, характеризующих условия смены состояний и вычисление значений выходных переменных.

а) Формулирование условий смены состояний

Под простым условием понимается любое условие, необходимое, но недостаточное для смены состояния элемента.

Полным условием называется совокупность простых условий, выполнение которых является необходимым и достаточным для смены состояния элемента. Простые условия могут быть независимыми и зависимыми от других простых условий.

Полные условия предопределяют переход элементов в состояния, которые в дальнейшем будут также делиться на зависимые и независимые. К зависимым относятся состояния, пребывание в которых соответствует взаимной занятости, индивидуальному закреплению или какому-либо другому отношению между элементами различных групп.

Зависимые состояния отражают ограниченные возможности элементов по вступлению в отношения взаимной занятости, закрепления и т.п., когда в каждое из таких отношений могут вступать не все элементы любой рассматриваемой группы, а лишь некоторая часть из них. Если возможности элементов (в только что указанном смысле) не ограничены, то соответствующие состояния относят к независимым.

При построении модели методом дифференциации состояний целесообразно в порядке продолжения оптимизации опорных графах зависимость их состояний отразить в явном виде.

Среди состояний элементов можно выделить зависимые и независимые.

Среди полных условий смены состояний элементов, соответствующих опорным графам, могут быть равносильные.

В опорных графах, в отличие от избыточных, рекомендуется на дугах указывать обозначения условий смены состояний.

Моделируемые элементы, которым соответствуют опорные графы, будут в дальнейшем называться опорными элементами.

Формулируемые полные условия смены состояний моделируемых элементов сводятся в таблицу. Так как по состояниям опорных элементов можно определить состояния всех других моделируемых элементов, полные условия смены состояний целесообразно формулировать только для опорных элементов.

Основным требованием, предъявляемым к формулируемым полным условиям, является содержательная полнота. Требование содержательной полноты означает необходимость учёта в полных условиях всех без исключения простых условий, существенных для моделирования.

В целях упрощения алгоритма, реализующего синтезируемую математическую модель, к полным условиям перехода в зависимые состояния целесообразно предъявить требование ординарности. Оно состоит в том, что при каждой однократной проверке всех полных условий любое полное условие перехода в зависимое состояние должно выполняться не более чем для одного элемента каждой группы.

Если требование ординарности не обеспечивается совокупностью основных простых условий, то вводят вспомогательные простые условия.

Под основными понимаются простые условия, вытекающие из реальных свойств моделируемого объекта (процесса) и из вводимых допущений.

Вводя вспомогательные условия, следует иметь в виду, что они могут повлиять на получаемые результаты решения задачи. Поэтому в их выборе должны участвовать специалисты той области профессиональной деятельности, для автоматизации которой разрабатывается задача, решаемая на ЭВМ.

Для выявления расчётных соотношений применяется весь арсенал приёмов, используемых при построении аналитических моделей ранее рассмотренным методом интеграции, однако этот метод применяется теперь не к объекту или процессу в целом, а к описанию свойств составляющих его элементов.

3. Построение алгоритма

Построение алгоритма сводится к определению совокупности предписаний, задающих последовательность действий, которые надо выполнить при варьируемых значениях входных переменных для получения значений выходных переменных в соответствии с найденными при построении математической модели математическими и логическими зависимостями. В результате построения алгоритма осуществляется переход от аналитической модели к имитационной математической модели. Реализация алгоритма приводит к воспроизведению (имитации) свойств моделируемого объекта или процесса.

Построение алгоритма в общем случае включает:

- ◆ приведение расчётных соотношений к виду, реализуемому на ЭВМ;
- ◆ построение схемы алгоритма и подготовку текстуального описания.

Приведение расчётных соотношений к виду, реализуемому на ЭВМ, осуществляется при необходимости и сводится к их выражению через операции и процедуры (функции), осуществимые в ЭВМ, при этом широко применяются численные методы математики [8].

При приведении расчётных соотношений к виду, реализуемому на ЭВМ, надо иметь в виду, что большое количество функций для многих типов ЭВМ может быть реализовано с помощью уже созданных стандартных программ. Поэтому перед построением алгоритма, предназначенного для реализации на ЭВМ конкретного типа, полезно ознакомиться с составом имеющейся библиотеки стандартных программ.

Построение схемы алгоритма машинного расчёта на основе аналитической модели, синтезированной методом интеграции, обычно трудностей не вызывает, так как сводится к отображению с помощью символов, установленных ГОСТ [26], порядка вычислений по формулам расчётных соотношений.

Алгоритм машинного расчёта обычно включает две основные части – задающую и вычислительную. Задающая часть является вспомогательной и содержит начальные предписания алгоритма, предопределяющие подготовку исходных данных и установку начальных значений переменных. Вычислительная часть является основной. Она содержит предписания, определяющие действия по расчёту и выводу значений выходных переменных. Обобщённая схема такого алгоритма представлена на рис. 3.1.

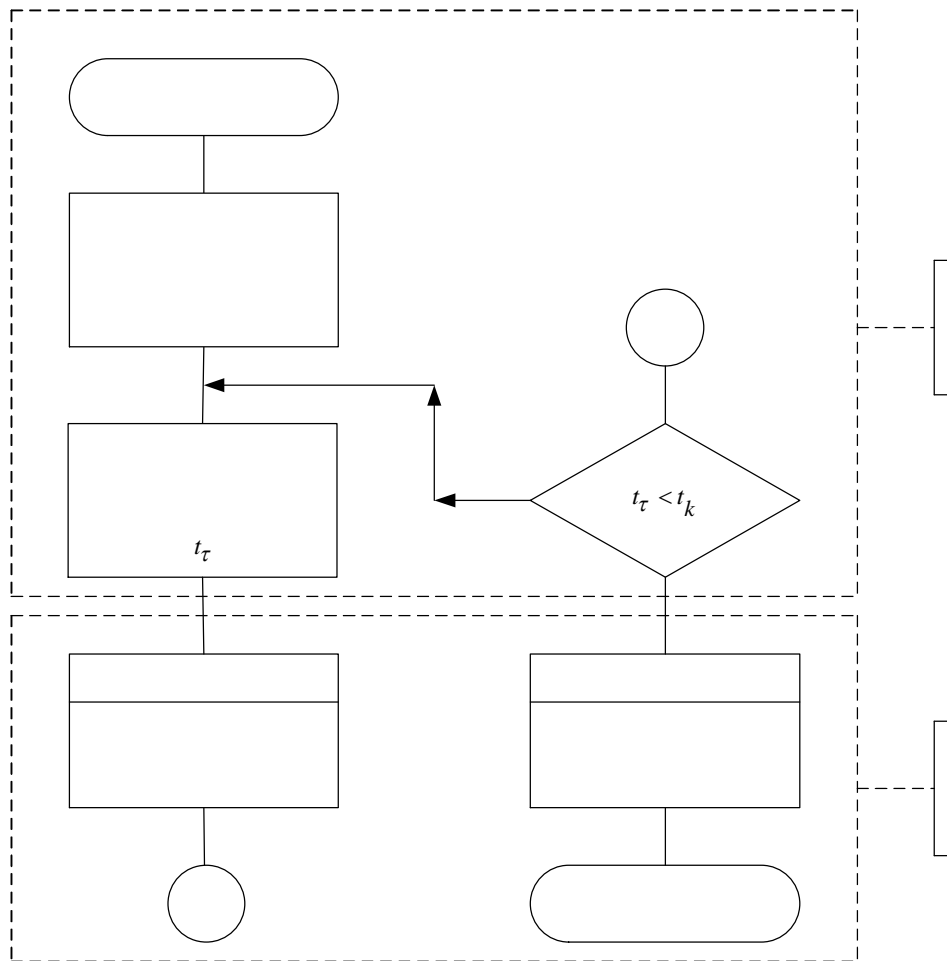


Рис. 3.1 Обобщённая схема алгоритма машинного моделирования

Алгоритм в данном случае состоит из задающей части и вычислительной части, при этом в вычислительной части можно, в свою очередь, выделить воспроизводящую и завершающую части.

Задающая часть содержит совокупность предписаний, определяющих следующие действия:

- ◆ ввод исходных данных, представляющих конкретные значения входных переменных;
- ◆ установка начальных значений переменных, характеризующих состояния элементов;
- ◆ задание или вычисление начального значения времени в моделируемом интервале;
- ◆ изменение значений времени в ходе воспроизведения (имитации) моделируемого объекта или процесса;
- ◆ прекращение воспроизведения по достижении конечного значения времени, которое может задаваться в исходных данных либо вычисляться в задающей части (а иногда и корректироваться в ходе воспроизведения).

Изменение значений времени при моделировании с использованием ЭВМ осуществляется дискретно. На практике применяют два принципиально различных способа изменения времени – способ стандартных приращений и способ существенных приращений.

При реализации способа стандартных приращений значение времени в каждом шаге воспроизведения модели увеличивается на одну и ту же величину Δt . К достоинствам способа стандартных приращений следует отнести простоту реализации, а к недостаткам – значительное снижение точности результатов при увеличении Δt и существенное возрастание затрат машинного времени при уменьшении Δt . Избежать указанных недостатков позволяет способ существенных приращений. При этом способе в качестве

очередного дискретного значения времени t выбирается ближайшее значение $t_{\text{ол}}$, которое соответствует условиям возможной смены состояния хотя бы одного элемента.

Следует отметить, что способ существенных приращений в задачах, реализуемых в искусственном масштабе времени, реализуем не всегда, поскольку часто, особенно в сложных моделях, не удастся определить все значения времени, при которых возможны смены состояний элементов модели. В отличие от этого, способ стандартных приращений реализуем всегда, то есть является универсальным.

Воспроизводящая часть алгоритма, строящегося на основе математической модели, полученной методом дифференциации, является основной. Она содержит совокупность предписаний, определяющих действия по воспроизведению (имитации) свойств отдельных элементов и моделируемого объекта или процесса в целом. Эта часть алгоритма предписывает для каждого элемента проверку полных условий смены состояний и в случае выполнения какого-либо из условий предопределяет перевод элемента из текущего состояния, относящегося к предыдущему дискретному значению моделируемого текущего времени, в последующее состояние, соответствующее очередному значению времени.

Обобщенная схема воспроизводящей части может иметь вид, представленный на рис.3.2.

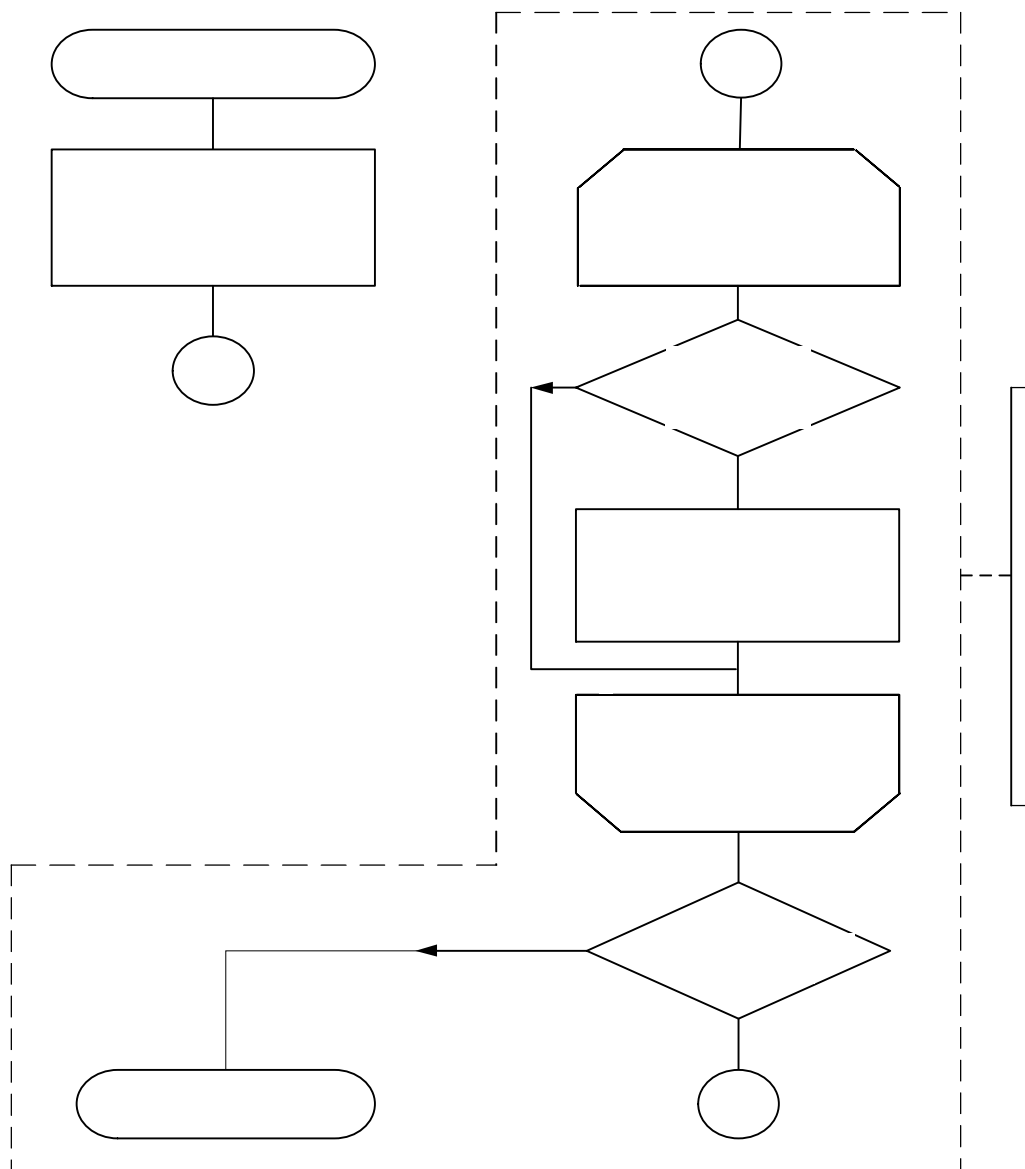


Рис. 3.2 Общая схема воспроизводящей части

В каждом шаге воспроизведения моделируемого процесса сначала для переменных величин, изменяющихся во времени независимо от состояний элементов, вычисляются их новые значения, относящиеся к очередному моменту текущего времени (рис. 3.2, блочный символ 2).

После этого в новые состояния переводятся элементы, для которых выполнены полные условия перехода из текущего состояния в очередное состояние (см. рис. 3.2, блочные символы 3-7).

По результатам однократной проверки всех полных условий (ввиду соблюдения ранее рассматривавшегося требования ординарности) в новые состояния переводятся не более чем по одному элементу каждой группы. Смена состояний тех или иных моделируемых элементов в общем случае создает условия для перехода других элементов в новые состояния. Поэтому проверка всех полных условий должна повторяться до тех пор, пока не окажется, что ни одно из условий перехода в зависимые состояния не выполняется (рис. 3.2, блочный символ 7). В этом случае воспроизведение моделируемого объекта (процесса) при очередном значении времени считается законченным и осуществляется переход к следующему изменению времени.

Завершающая часть алгоритма содержит совокупность предписаний по формированию и выдаче значений выходных переменных. Эта часть алгоритма, как правило, определяет порядок вычисления окончательных результатов на основе значений входных данных и промежуточных величин, полученных в ходе воспроизведения модели, а также вывода их из ЭВМ.

Построение алгоритма заканчивается подготовкой его текстуального описания, которое поясняет используемые математические и логические зависимости и предусматриваемый порядок действий при реализации соответствующей математической модели на ЭВМ.

Заключение

В целом предлагаемая методика алгоритмизации существенно отличается от известных, традиционно сводящихся к описанию на конкретных примерах того, что должно быть получено в процессе алгоритмизации. Методика впервые представляет собой главным образом систему методов, с использованием которых может быть получен требуемый результат в условиях реально возможного множества алгоритмических решений.

Литература

1. Янов Ю.И. О логических схемах алгоритмов. Проблемы кибернетики, вып. 1. – М.: Физматгиз, 1958.
2. Трахтенброт Б.А. Алгоритмы и машинное решение задач. – М.: Физматгиз, 1960.
3. Криницкий Н.А. Язык логических схем. В сб. «Цифровая техника и программирование», вып. 1. М: Сов. Радио, 1966.
4. Криницкий Н.А. Равносильные преобразования алгоритмов и программирование. – М.: Сов. Радио, 1970.
5. Блох А.Ш., Неверов Г.С. В помощь авторам алгоритмов. – Минск: Беларусь, 1971.
6. Налимов В.В. Теория эксперимента. – М.: Наука, 1971.
7. Голембо З.Б. Алгоритмизация и программирование электротехнических задач на ЭЦВМ. – М.: Высшая школа, 1974.
8. Бахвалов Н.С. Численные методы. . – М.: Наука, 1975.
9. Измайлов А.Ф. Алгоритмизация и программирование. – Л.: ВМА, 1975.
10. Мельцер М.И. Разработка алгоритмов АСУП. – М.: Статистика, 1975.
11. Тюрин Ю.Н. Что такое математическая статистика. – М.: Знание, 1975.

12. Раскин Л.Г. Анализ сложных систем и элементы теории оптимального управления. – М.: Сов. Радио, 1976.
13. Криницкий Н.А. Алгоритмы вокруг нас. – М.: Наука, 1977.
14. Ахо А., Хопкфорд Д., Ульман Д. Построение и анализ вычислительных алгоритмов. – М.: Мир, 1979.
15. Дайитбегов Д.М. Черноусов Е.А. Основы алгоритмизации и алгоритмические языки. – М.: Статистика, 1979.
16. Гудман С., Хидетниemi С. Введение в разработку и анализ алгоритмов. – М.: Мир, 1981.
17. Шуп, Терри Е. Решение инженерных задач на ЭВМ: Практическое руководство: Перевод с англ. – М.: Мир, 1981.
18. Сергеев Н.П., Домнин Л.Н. Алгоритмизация и программирование. М.: Радио и связь, 1982.
19. Завалишин С.Т., Суханов В.И. Прикладные задачи синтеза управляющих алгоритмов. – М.: Наука, 1985.
20. Васильев Ю.С., Виссарионов В.И. Кубышкин Л.И. Решение гидроэнергетических задач на ЭВМ (элементы САПР и АСНИ). – М.: Энергоатомиздат, 1987.
21. Ланчук М.П. Алгоритмизация. Программирование / Пособие для учителя. М.: Просвещение, 1988.
22. Гулд Х., Тобочник Я. Компьютерное моделирование в физике: Перевод с англ. – М.: Мир, 1990.
23. Мозговой М.В. Занимательное программирование: Самоучитель. – СПб.: Питер, 2004
24. Одинцов И.О. Профессиональное программирование. Системный подход. – 2-е изд. перераб. и доп. – СПб.: БХВ-Петербург, 2004.
25. Голицина О.Л., Попов И.И. Основы алгоритмизации и программирования: Учебное пособие. – М.: ФОРУМ: ИНФРА-М, 2005.
26. ГОСТ 19.701-90. Единая система программной документации. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения.