УДК 510.6.66

Дедуктивная и индуктивная системы правил вывода для построения формальных систем

Н.Л. Малинина.

Цель статьи – сравнительная оценка сложности дедуктивной (ДСПВ) и индуктивной (ИСПВ) систем правил вывода при построении формальных систем. При использовании правил вывода «если ***q****, то* ***P****» (ДСПВ) построение формальной системы осуществляется в лучшем случае с помощью полиномиальных алгоритмов синтеза. Поэтому характерной особенностью развития программирования в настоящее время является постоянное увеличение стоимости разработки программных средств по мере усложнения задач. Индуктивный подход («****P****, если* ***q****») не требует указания адресов передачи выходной информации, что существенно упрощает задачу составления алгоритма. Сравнение особенностей ДСПВ и ИСПВ позволяет сделать следующие выводы: ДСПВ отличается более высокой сложностью по сравнению с ИСПВ и не обеспечивает однозначного линейного упорядочения алгоритмических операторов при построении формальной системы, тогда как применение ИСПВ позволяет формализовать процесс построения алгоритма линейным образом.*

Целью статьи является сравнительная оценка сложности дедуктивной (ДСПВ) и индуктивной (ИСПВ) систем правил вывода при построении с их помощью такой формальной системы, как алгоритм.

Характерной особенностью развития программирования на базе существующих в настоящее время различных алгоритмических языков и операционных систем является постоянное увеличение стоимости разработки программных систем по мере усложнения задач, решаемых с помощью вычислительной техники. К началу 80-х годов стоимость программного обеспечения уже составляла ≈ 85% от общей стоимости ЭВМ и программного обеспечения, и это соотношение постоянно растет. Программирование складывается как очень трудно дисциплинируемая область творческой деятельности человека и существующие на сегодняшний день способы разработки и построения математического и программного обеспечения (МПО) для задач, решаемых в САПР, АСК и при математическом моделировании сложных процессов в основном сводятся к единой технологической схеме (рис.1).

Рис. 1. Единая технологическая схема разработки МПО.

Подавляющее большинство разработчиков до недавнего времени вообще объединяли 4, 6 и 7 этапы в один - программирование. По мере усложнения задач выявилась необходимость выделения построения блок-схемы алгоритма в самостоятельный этап работы.

Два последних этапа (6 и 7) разработки МПО являются самыми сложными и трудоемкими, поскольку приходится одновременно решать три группы вопросов:

- Упорядочение всех операторов алгоритма.
- Логическую и информационную увязку операторов алгоритма между собой.
- Информационное снабжение программы из внешних источников и базы данных.

Смешение трех групп вопросов делает работу специалистов весьма сложной и не позволяет применить автоматизацию, поскольку стандартизировать заранее все возможные варианты упорядочения операторов алгоритма и комбинации логических переходов между ними с их информационным обеспечением возможно лишь для наиболее простых случаев. В настоящее время можно выделить четыре принципиальных подхода к разработке МПО, схемы которых представлены на рис. 2.

Рис. 2. Принципиальные схемы разработки МПО.

Рассмотрение возможных вариантов разработки МПО в применении к системам автоматизированного проектирования и конструирования показывает, что в настоящее время наиболее реально используется один из самых малопроизводительных: прямое программирование. Применение ***R***-технологии все-таки не исключает построения схемы алгоритма вручную. Кроме того, представляется сомнительным, чтобы удалось написать в достаточном количестве правила преобразования исходной информации для таких трудно формализуемых процессов как проектирование и конструирование. Система ПРИЗ или аналогичные ей методы объектно-ориентированного программирования не всегда обеспечивают построение адекватных моделей сложных процессов и систем, не обеспечивает однозначности решения и отличается высокой сложностью.

Все это происходит потому, что до сих пор построение алгоритмов для решения задач различных типов осуществляется только с помощью средств дедуктивной математики, т.е. формальную систему или структурную схему алгоритма строят обычно от исходных данных (аксиом) к окончательному результату (теореме), применяя правила вывода типа: **"если *q*, то *P*".** При этом синтез или построение формальной системы осуществляется либо вручную, либо с помощью дедуктивных алгоритмов синтеза: полный перебор, метод ветвей и границ, некоторые полиномиальные алгоритмы синтеза и т.д. В последнее время, в связи с появлением других математических методов [1], появилась реальная возможность развития индуктивного подхода применительно к построению формальных систем, т.е. применение правил вывода типа: **"*P*, если *q*".**

Проблема выбора между индукцией и дедукцией имеет глубокие корни. В связи с этим необходим краткий исторический экскурс. Как было отмечено еще Аристотелем, к любому обобщению, т.е. познанию общего через частное, человек приходит индуктивным путем, применяя правила вывода типа: **"*P*, если *q*",** тогда как при использовании общих положений в познании единичного (частного) он всякий раз применяет метод дедукции, т.е. правила вывода типа: **"если *q*, то *P*".** С тех пор дедуктивные методы познания мира получили широкое распространение и подкрепление в виде математических методов. Возможно, что это произошло еще и потому, что тотальное распространение веры и религии привело к необходимости согласовывать свои утверждения с существующей догмой. Правда последние три столетия происходит смена теоретико-познавательного идеала - выдвигаются программы построения индуктивных логик как логик открытия и доказательства научных истин. Однако здесь мы сталкиваемся с некоторой неопределенностью индуктивных выводов, поскольку в силу временных ограничений вводится принцип рассмотрения не всего количества возможных частных случаев, а лишь части из них. Поэтому выводы по неполной индукции всегда носят вероятностный характер, который напрямую связан с незавершенностью человеческого опыта. Эту неопределенность в уже построенных индуктивных системах пытаются разрешить с помощью различных прагматических категорий: полезность, интерес, субъективная вероятность, а также вводя человеческий фактор в процедуры принятия решений. Возможно также, что недостаточное распространение индуктивной логики связано с отсутствием методов ее формального построения. Кроме того, хочется отметить, что в настоящее время индуктивные системы строятся с помощью существующих дедуктивных математических методов. В заключение остается только привести слова Энгельса: "Индукция и дедукция связаны между собой как синтез и анализ. Вместо того чтобы односторонне превозносить одну из них до небес за счет другой, надо стараться применять каждую на своем месте, а этого можно добиться лишь в том случае, если не упускать из виду их связь между собой, их взаимное дополнение друг друга".

Отметим, что метод, связывающий воедино индукцию и дедукцию, для регулярного построения формальных систем как структуры вычислительных алгоритмов и программ изложен в [2]. Любую формальную систему можно представить в виде ориентированного или неориентированного графа: вершинного ***G(Q,Г),*** где операторам алгоритма соответствуют вершины, а связям - дуги, или реберного ***H(V,Q),*** где операторам ставятся в соответствие дуги графа, а связям между ними - вершины. Первым соответствуют сопряженные сетевые модели (ССМ), вторым - обыкновенные сетевые модели (ОСМ).

В обычной практике разработки формальных систем в виде алгоритмов или программ применяют ССМ или блок-схемы, которые, как правило, непосредственно не обладают свойством эффективной рекурсивности, что является необходимым условием для реализации формальной системы на ЭВМ. Эта особенность блок-схем создает основные трудности при разработке алгоритмов и программ. Однако появилась возможность автоматизированного перехода от ССМ к ОСМ с помощью регулярных алгоритмов синтеза первого порядка [1,2]. Кроме того, с помощью тех же алгоритмов синтеза, стало возможным немедленное построение непосредственно ОСМ, обладающих свойством эффективной рекурсивности. Поэтому в настоящее время можно говорить о реальном сравнении ДСПВ и ИСПВ в применении к построению формальных систем.

Сравнивая систему аксиом и правил вывода при дедуктивном и индуктивном подходах к изучению решения задачи и последующему построению формальной системы, можно сказать, что ДСПВ при описании решения задачи требует формирования достаточно сложной исходной системы аксиом и правил вывода, сложность которых возрастает по мере роста сложности задач. В случае дедуктивного подхода при построении алгоритма или программы от данных к цели число правил вывода равно числу дуг графа. При индуктивном подходе число правил вывода равно числу вершин графа без одной. Оценим сложность ДСПВ по сравнению с ИСПВ. Обозначим:

m - число правил вывода в ДСПВ (число дуг графа);

$n-1$ - число правил вывода в ИСПВ (число вершин графа без одной).

Тогда относительная сложность ДСПВ по отношению к сложности ИСПВ будет:

$$\bar{S} = \frac{m}{n-1}$$

Одной из характеристик сложности структуры графа может служить его цикломатическое число:

$$v = m - n + 1 = m - (n-1)$$

Введем коэффициент плотности заполнения матрицы смежности:

$$r = \frac{m}{M_{\max}}$$

где: $$M_{\max} = \frac{n^2 - n}{2}$$

Тогда: $$m = r * M_{\max} = r * \frac{n^2 - n}{2}$$

$$v = \left(\frac{r}{2} * n - 1\right) * (n-1)$$

Обозначим:

$M_{ис}$ - число правил вывода в ИСПВ;

$M_{дс}$- число правил вывода в ДСПВ.

Легко видеть, что:

$$M_{дс} = r * \frac{n^2 - n}{2}$$

$$M_{ис} = n - 1$$

$$\bar{S} = \frac{M_{дс}}{n - 1} = r * \frac{n}{2}$$

Таким образом, число правил вывода в ДСПВ практически пропорционально квадрату числа последовательно выводимых теорем (применительно к разработке алгоритмов или программного обеспечения - квадрату операторов или модулей алгоритма). Показатель параболы может быть даже выше [3]. Правда, в ДСПВ многие правила вывода повторяются, и число повторяющихся равно числу правил вывода в ИСПВ. Тем не менее, чтобы исключить все повторяющиеся правила вывода, необходимо вначале переписать их все. Поэтому на концептуальном этапе проектирования для задач высокой и очень высокой размерности требуются огромные затраты времени и труда для построения формальной системы в виде набора аксиом и правил вывода. Это обстоятельство является главной причиной быстрого роста трудозатрат в случае разработки алгоритмов и программ с использованием ДСПВ (рис. 3).

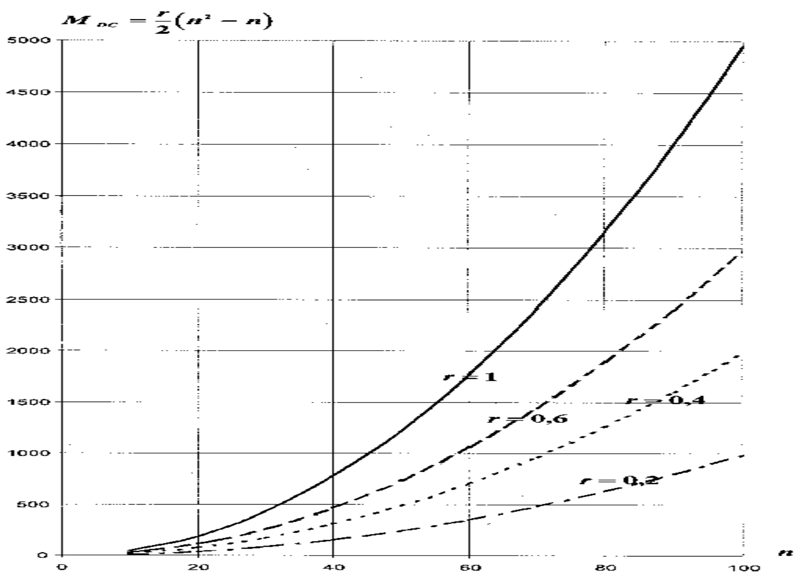

Рис. 3. Сравнительная оценка сложности дедуктивной и индуктивной систем правил вывода.

С помощью рис. 4 и 5 можно провести сравнение дедуктивного и индуктивного подходов к анализу решения задачи для построения формальной системы.

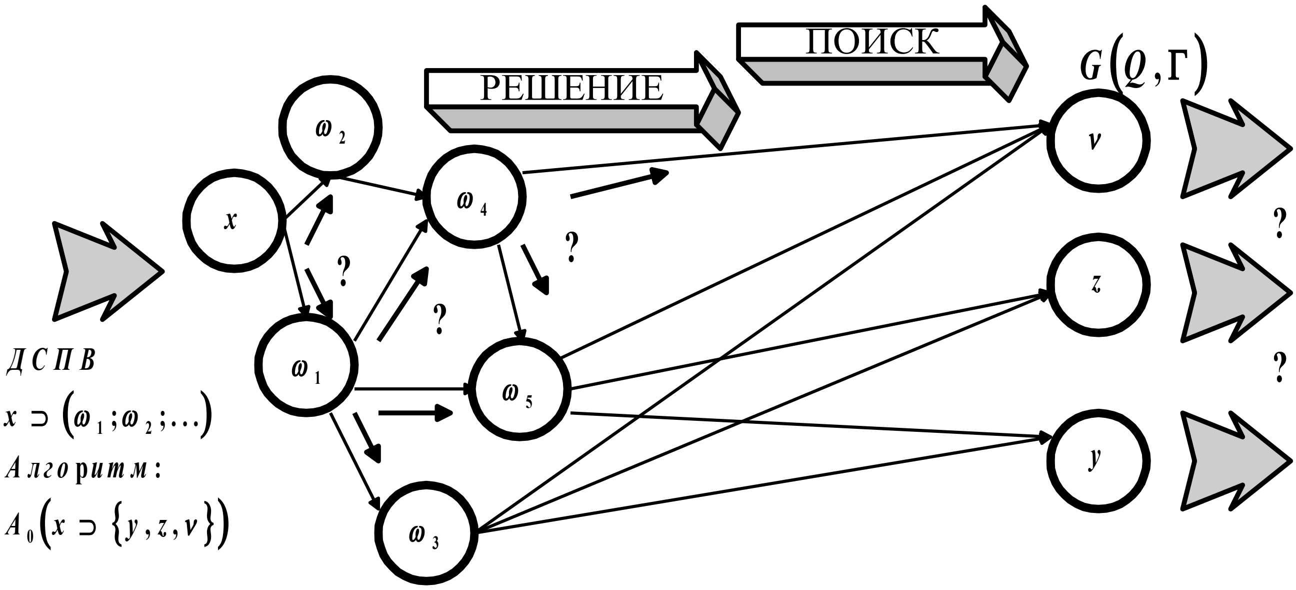

Рис. 4. ДСПВ – неопределенность поиска – неоднозначность решения.

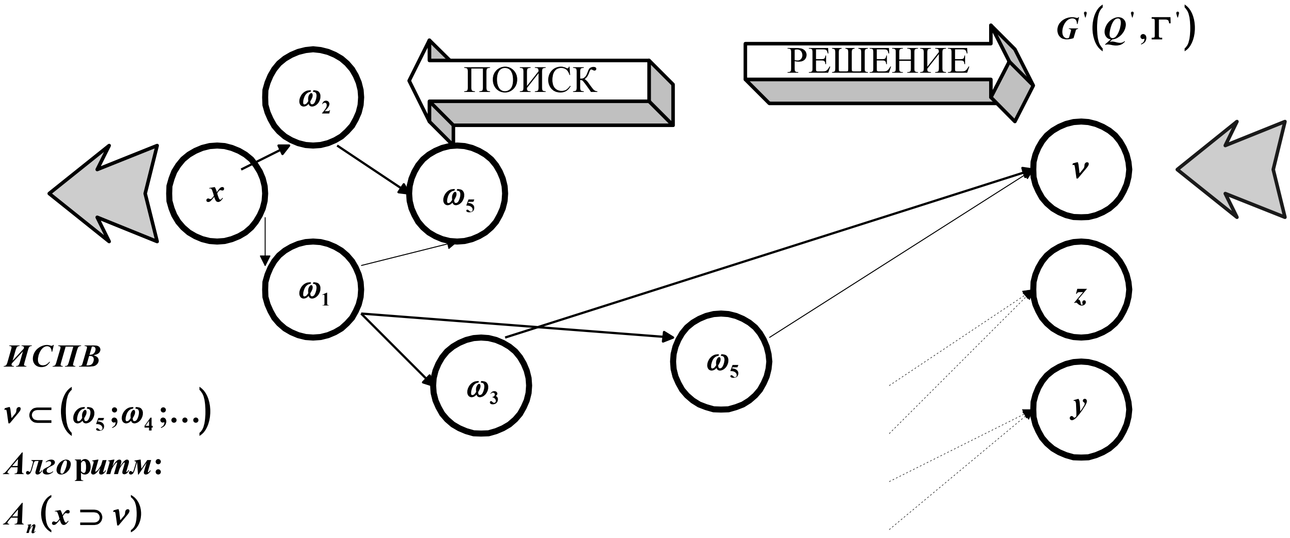

Рис. 5. ИСПВ – определенность поиска – однозначность решения.

Видно (рис. 4), что четвертое правило вывода не может быть реализовано раньше пятого. Следовательно, ДСПВ не обладает свойством эффективной рекурсивности, то есть: при составлении ДСПВ нет достаточной информации, чтобы относительно простым способом построить строгую формальную систему, обладающую свойством эффективной рекурсивности, хотя рассматривается простейший случай. В процессе разработки программного обеспечения в настоящее время повсеместно применяется ДСПВ, хотя бы потому, что на сегодняшний день в математике существуют в основном только дедуктивные формальные системы. Однако, применяя ДСПВ, мы неминуемо сталкиваемся с необходимостью включения в программу поиска путей решения задачи блоков проверки ранее не выполнявшихся правил вывода, для чего может потребоваться неоднократный просмотр и сопоставление многих путей в графе решения задачи на несколько шагов вперед, включая и случаи просмотра путей до конца.

Индуктивный подход, с одной стороны, требует, чтобы входы каждого модуля были точно описаны с указанием источников информации, что не очень усложняет задачу составления базы знаний. С другой стороны, индуктивный подход , в сравнении с дедуктивным, не требует указания адресов передачи выходной информации, что существенно упрощает задачу составления алгоритма. Кроме того, в ИСПВ отсутствуют все повторяющиеся правила вывода.

В простейшем случае ДСПВ может быть представлена в виде ориентированного графа без контуров ***G(Q,Г),*** где:

$\boldsymbol{Q}$ - множество вершин; $\boldsymbol{\Gamma = Q_i \rightarrow Q_j}$ - множество дуг.

В таком графе (рис. 4):

$\{x\}$ - множество минорант, $\{x\} \in \boldsymbol{Q}$;

$\{y, z, v, \ldots\}$ - множество мажорант, $\{y, z, v, \ldots\} \in \boldsymbol{Q}$.

Причем любая дуга $\boldsymbol{q_x \in Q}$ соединена по крайней мере одним путем с одной из $\boldsymbol{q_y \in}$ $\{y, z, v, \ldots\}$. Однако при выходе из миноранты $\boldsymbol{x}$ нельзя указать в точности дугу, которой начинается путь, ведущий в искомую мажоранту $\boldsymbol{v}$**.** Это утверждение справедливо для каждой из вершин, находящихся на любом из путей, соединяющих множество минорант $\{x\}$ с множеством мажорант $\{y, z, v, \ldots\}$. В то же время для получения однозначного решения задачи необходимо выявить в графе ***G(Q,Г)*** все те и только те пути, которые соединяют множество минорант $\{x\}$ с искомой мажорантой $\boldsymbol{v}$**.**

Для выявления таких путей необходимо:

1. Определить все пути, соединяющие множество минорант $\{x\}$ с множеством мажорант $\{y, z, v, \ldots\}$.

2. Выделить из этого множества путей, подмножество тех путей, которые заканчиваются искомой мажорантой $\boldsymbol{v}$**.**

3. Исключить из графа ***G(Q,Г)*** все дуги и вершины, не принадлежащие выделенному подмножеству путей. Оставшийся подграф ***G'(Q',Г')*** будет отражать искомый алгоритм.

Таким образом, при применении ДСПВ, мы сталкиваемся с неопределенностью при поиске подграфа, являющегося отображением искомого алгоритма, и с неоднозначностью решения. Разрешить эту неопределенность при использовании ДСПВ можно либо введением дополнительной информации, сужающей область поиска, либо перебором всех путей в графе ***G(Q,Г).*** Индуктивная система правил вывода, в отличие от дедуктивной, позволяет исключить фактор неопределенности <u>на данный момент времени</u> для исходной системы аксиом. Действительно, при применении ИСПВ, поиск пути, соединяющего $\{x\}$ с заданной мажорантой $\boldsymbol{v}$, начинается именно с искомой мажоранты. В этом случае при движении в графе ***G(Q,Г)*** от искомой мажоранты $\boldsymbol{v}$ к миноранте $\{x\}$ при прохождении любой из вершин выход из вершины

ω_i осуществляется только в направлении тех вершин ω_{i-1}, которые необходимы, как исходные данные для работы алгоритмического оператора в вершине ω_i. Поэтому любой путь от мажоранты $\boldsymbol{v}$ к миноранте $\{x\}$ строится вполне определенно. При этом, если $\{x\}$ - суть множество вершин, то после построения всех путей, ведущих в обратном направлении от искомой мажоранты $\boldsymbol{v}$ к миноранте $\boldsymbol{x}$ из множества $\{x\}$ исключаются те вершины, в которых не начинается ни один путь, ведущий в мажоранту $\boldsymbol{v}$.

В рассмотренном примере на рис. 3 при построении подграфа ***G'(Q',Г')*** с использованием ДСПВ необходимо, прежде всего, построить граф ***G(Q,Г),*** содержащий 14 путей, соединяющих $\{x\}$ с $\{y, z, v, \ldots\}$. В то же время граф ***G(Q,Г)*** (рис. 4), построенный с применением ИСПВ, содержит всего 6 путей, соединяющих $\{x\}$ с $\boldsymbol{v}$.

Проведенное обсуждение особенностей ДСПВ и ИСПВ позволяет сделать следующие выводы:

1. ДСПВ отличается более высокой сложностью по сравнению с ИСПВ.
2. ДСПВ не обеспечивает однозначного линейного упорядочения блоков алгоритма или алгоритмических операторов при построении формальной системы.
3. ДСПВ не обеспечивает определенности поиска при построении структурной схемы алгоритма и однозначности решения задачи.
4. Применение ИСПВ позволяет формализовать процесс построения алгоритма линейным образом и исключить фактор неопределенности на данный момент времени. В случае изменения системы исходных аксиом формальную систему легко вновь построить или перестроить с учетом новых исходных данных.
5. Применение ИСПВ для построения формальной системы позволяет существенно сократить трудозатраты на разработку алгоритма и программы уже на концептуальном этапе проектирования.

ЛИТЕРАТУРА:

1. Малинин Л.И. Восемь дополнительных теорем из теории графов. В кн. Нечипоренко В.И. Структурный анализ сложных систем. М.: Сов. Радио, 1977.-С.198-212.
2. Малинина Н.Л. Автоматизированный синтез вычислительных алгоритмов для проектирования сложных систем. Дис. к. ф.-м. н.// МАИ.- М.: 1993. - 200с.
3. Тыугу Э.Х. Концептуальное программирование. М.: Наука. 1984.-255с.
4. Нагих Н.Я. К вопросу о происхождении индукции и дедукции. Автореферат на соискание ученой степени кандидата философских наук. М.: МГУ, 1968.-15с.
5. Формальная логика. Л.: ЛГУ, 1977.-357с.

6. Иванова И.И. Сравнительный анализ методов античной, средневековой и классической логики. Автореферат на соискание ученой степени кандидата философских наук. М.: МГУ,1984.-25с.

7. Индуктивная логика и формирование научного знания. Сб. статей. М.: Наука, 1987.-185с.

8. Кайберг Г. Вероятность и индуктивная логика. М.: Прогресс, 1978.-373с.

СВЕДЕНИЯ ОБ АВТОРЕ

Малинина Наталия Леонидовна, доцент кафедры вычислительной математики и программирования Московского авиационного института (государственного технического университета), к.ф.-м.н.