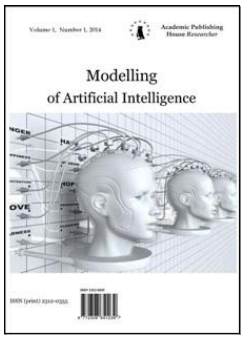


Published in the Russian Federation
Modeling of Artificial Intelligence
Has been issued since 2014.
ISSN: 2312-0355
Vol. 3, No. 3, pp. 138-148, 2014

DOI: 10.13187/mai.2014.3.138
www.ejournal11.com

UDC 004.8

# Conclusions of Intellectual Systems

Victor Ya. Tsvetkov

Moscow State Technical University of Radio Engineering, Electronics and Automation MSTU MIREA, Russian Federation
E-mail: cvj2@mail.ru

**Abstract**

This article describes the methods for collecting automated solutions in intellectual systems. This article features conclusive methods based on the forward and reverse chains, the difference between the method of straight chain and the method of the inverse chain. This article describes methods of finding solutions in the state space, the difference between in-depth and breadth-first research. A backtracking method is described. The reduction method is revealed. The article shows the limits of applicability of methods and provides a comparison of described methods with the method of precedents.

**Keywords**: artificial intellect; conclusive rules; graphs; decomposition; problem solving; formalization; intelligent systems; intellectual transport systems.

## Введение

В настоящее время актуальной проблемой в области искусственного интеллекта является проблема конструирования интеллектуальных систем реального времени [1], типичными представителями которых являются интеллектуальные транспортные системы (ИТС) [2, 3]. В более обобщенном понимании такие системы называют созданию интеллектуальные системы поддержки принятия решений (ИСППР) [4] для реального времени. Такие системы ориентированы на открытые и динамические предметные области. В основе таких систем лежит возможность вывода на основе имеющейся информации и информационных ресурсов. Они способны к обучению моделей представленных в виде знаний, способны уменьшать неопределенность и изменять свое состояние.

Современные исследования по созданию интеллектуальных систем поддержки принятия решений (ИСППР), в частности, интеллектуальных транспортных систем (ИТС), непосредственно связаны с проблемой вывода [5]. Она реализуется по-разному: логический вывод, метод прямой и обратной цепочек [6]; методы поиска решений в пространстве состояний [7], моделирования правдоподобных рассуждений [8] и другие. Второй и третий методы наиболее удобны с точки зрения их программной реализации. Рассмотрим проблему вывода в интеллектуальных система реального времени на примере ИТС.

**Методы вывода на основе прямой и обратной цепочек.** Как всякая интеллектуальная система ИТС включает базу знаний и базу данных. В базе знаний (БЗ), хранятся правила, факты сдержатся в базе данных (БД). Правила могут создаваться и

накапливаться в БЗ, данные (факты) могут создаваться и накапливаться в БД. Отмечаем принципиальное отличие. В обычной базе данных данные вводятся только пользователем. В ИТС, на основе совокупности правил, данные могут создаваться самой системой.

При продукционном представлении область знаний в ИТС представляется множеством продукционных правил «ЕСЛИ → ТОГДА», а данные представляются множеством фактов о текущей ситуации.

ИТС может использовать логическую цепочку вывода для принятия решения на основе имеющихся фактов; это является основной частью ее способностей. ИТС также может использовать логическую цепочку вывода для получения схемы решения и сведения проблемы к фактам. Механизм вывода должен быть построен так, чтобы правила сработали для получения правильного решения. Для этого применяют два способа:

Первый способ называется прямая, или условно-выводимая, цепочка. Он основан на декомпозиции проблемы [9] или задачи – для получения (вывода) графа решения или структуры вывода [10]. Этот способ используется для обучения ИТС. Он определяет совокупность логических правил для получения решений. Этот способ задает структуру вывода проблемы или алгоритм решения. Разбиение или декомпозиция осуществляется до уровня данных (фактов), которые делятся на определяемые (или измеряемые) и вычисляемые.

Второй способ называется обратная, или целе-выводимая, цепочка. Он и служит основой принятия решений ИТС или вывода. Он использует совокупность существующих логических правил для получения решений при наличии исходных данных. При этом возможны два варианта: все данные заданы, часть данных вычисляется. Первый пример показан на рис. 1.

Механизм вывода решения сопоставляет каждое правило, хранящееся в базе знаний (БЗ) с фактами, содержащимися в базе данных (БД) Когда часть правила «ЕСЛИ» (условие) подходит факту, правило срабатывает и его часть ТОГДА (действие) *исполняется*. Срабатывающее правило может изменить множество фактов путем добавления нового факта.

Сопоставление частей «ЕСЛИ» правил с фактами создает цепочку вывода на основе фактов. Цепочка вывода показывает, как ИТС применяет правила для получения решения (заключения). Для иллюстрации метода вывода на основе цепочки, рассмотрим простой пример, когда все факты заданы.

Допустим, БД первоначально включает факты A,B,C,Y , а БЗ содержит следующие правила:

Правило 1.Y&X →Z

Правило 2. E&D&F→X

Правило 3. A→E

Правило 4. B→D

Правило 5. C→F

Цепочка вывода на рис. 1. показывает, как ИТС применяет правила для вывода факта Z.

Сначала срабатывает Правило 5 для вывода нового факта F из данного факта C. Затем срабатывает Правило 4 для вывода нового факта D из данного факта B. Затем срабатывает Правило 3 для вывода нового факта E из данного факта A.

Затем Правило 2 выполняется для вывода факта X из вторичных фактов F, D и E. И наконец, Правило 1 применяет первоначально известный факт Y и только что полученный факт X для прихода к заключению Z.

В данном случае осуществляется анализ левой части правила.

Продукционные системы, в которых сначала анализируется антецедентная часть (условия), имеют условно-выводимую архитектуру. Они основаны на интерпретации логической связки «A→E» правилом «ЕСЛИ → ТОГДА».

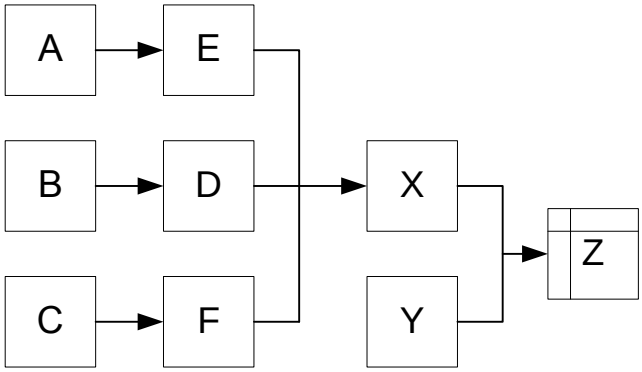


*Рис. 1.* Пример цепочки вывода решения, доказывающего истинность Z

Возможны другие виды интерпретации, что создает другие виды архитектур. Альтернативным типом архитектуры, которая достаточно часто используется в ИТС, являются действие-выводимые или консеквент-выводимые продукционные системы. Например, логическая связка вида

A&B&C→D

может быть интерпретировано в виде отмеченного выше продукционного правила, как «Логическая конъюнкция A, B и C влечет D» .

Но эта логическая связка может быть интерпретировано по другому «Чтобы доказать D, необходимо установить A, B, C».

В последнем случае цель должна быть достигнута дедуктивным выводом. Для этого исследуются консеквенты правил для нахождения такого правила, которое позволило бы достичь цели. Когда такое правило найдено, проверяются на истинность все его условия. Если условия истинны, продукция активируется. В противном случае продолжается поиск подходящей продукции.

Рассмотрим упрощенный пример системы с консеквент-выводимой архитектурой. Элементы БД считаются истинными, если содержатся в ней. БД содержит A и F. БЗ содержит следующие правила.

Правило 1: A&B&C→D

Правило 2: D&F→G

Правило 3:A&J→*G*

Правило 4: B→C

Правило 5: F→B

Правило 6: L→J

Правило 7: G→H

Предположим, цель состоит в том, чтобы вывести истинность H. Построим цепочку шагов.

1. В первую очередь проверяется, находится ли H в БД?

2. Так как в данном случае это не так, то система пытается вывести истинность H, используя правила, имеющие H в правой части. Таким является правило 7.

3. Теперь система пытается вывести истинность G, так как истинность последнего влечет за собой истинность H. Снова проверяется БД: в БД нет G, следовательно, организуется поиск правила, содержащего G в правой части.

Таких правил два. При наличии неоднозначности (несколько альтернативных правил) возникает «конфликт». В качестве стратегии «разрешения конфликта» используют отношения порядка. То есть правила упорядочивают по приоритету, при котором правилу с наименьшим номером соответствует больший приоритет.

4 . В данном случае выбирается правило 2, поэтому целью теперь становится вывести истинность D и F.

5. A находится в базе данных, следовательно A – истинно

6. F находится в базе данных, следовательно F – истинно

7. D находится по правилу 1 через A, B, C

8. B находится по правилу 5 через F
9. C находится по правилу 4 через B

Таким образом, цель достигнута. Элементы, истинность которых доказана, добавляются в БД. В данном случае это – элементы H, G, D, C. B.

На рис. 2. приведена логическая схема декомпозиции H. Эта схема задает структуру правила выводя H из из A, F. Она показывает какие правила из имеющихся в БЗ надо использовать и в какой последовательности. Правила обозначены кружочками, выводимые (вычисляемые) факты квадратиками, истинные факты (исходные A, F) обозначены квадратиками с двойными сторонами.

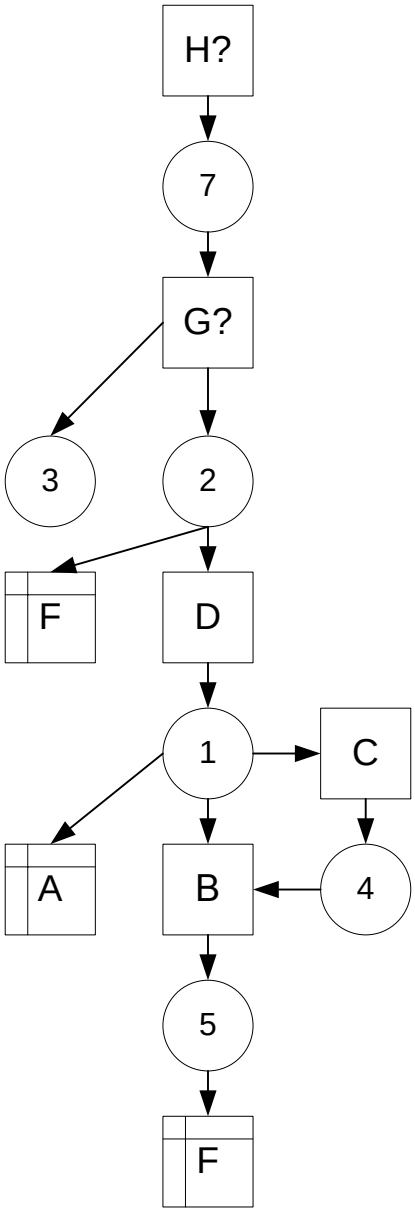


*Рис. 2.* Логическая схема декомпозиции, доказывающая истинность H из A, F

Таким образом, мы получили схему, согласно которой истинность A, F влечет истинность H. «(A, F) →H» Обращает на себя внимание возможная неоднозначность решения. Эта проблема, с которой всегда приходится сталкиваться.

**Методы поиска решений в пространстве состояний.** *Методы перебора* [11]**.** Решение многих задач в интеллектуальных системах можно определить как проблему поиска,

где искомое решение - это цель поиска, а множество возможных путей достижения цели представляет собой пространство поиска (или пространство состояний). Поиск решений в пространстве состоит в определении последовательности операторов, которые преобразуют начальное состояние в целевое.

Задачу поиска в пространстве состояний можно сформулировать в общем виде так: Пусть исходная задача описывается тройкой (*S, F, T),* где *S* – множество начальных состояний; *F* – множество операторов, отображающих одни состояния в другие; *T* – множество целевых состояний.

Решение задачи состоит в нахождении последовательности операторов $f_1, f_2, f_3$ — ($f_i \in$ F), которые преобразуют начальные состояния в конечные. Задача поиска в пространстве состояний описывается с помощью понятий теории графов. Задается начальная вершина Pr (исходное состояние рис. 2) и множество целевых вершин *T*. Для построения пространства состояний используются операторы *F*.

Последовательно от корня к вершинам дерева применяются операторы, относящиеся к этому уровню, для построения вершин-преемников следующего уровня (раскрытие вершин). Дуги идентифицируются как операторы преобразования. Получаемые вершины - преемники проверяются: не являются ли они целевыми. Далее переходят к следующему уровню. Терминальные вершины – это те, к которым нельзя применить никаких операторов. Раскрытие продолжается до тех пор, пока не получена целевая или терминальная вершина. Поиск на графе состояний – это процесс построения графа Pr, содержащего целевую вершину. Этот граф называется графом декомпозиции проблемы.

Как было отмечено выше, возможна неоднозначность решения и наличие альтернатив. Это обуславливает наличие нескольких вариантов реализации метода перебора, которые отличаются порядком, в котором будут перебираться вершины.

*Поиск в глубину*. При поиске в глубину прежде всего раскрывается та вершина, которая имеет наибольшую глубину (рис. 3). Из вершин, расположенных на одинаковой глубине, выбор вершины для раскрытия определяется произвольно. Для сдерживания возможности следования по бесперспективному пути вводится ограничение на глубину. Вершины, находящиеся на граничной глубине, не раскрываются.

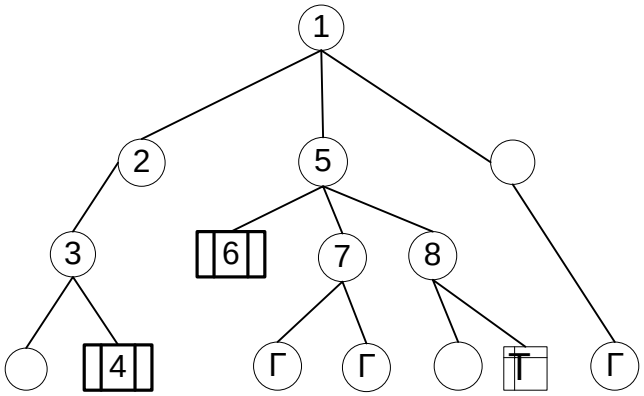


*Рис. 3.* Граф состояния, по которому осуществляется поиск в глубину

На рис. 3 терминальные вершины обозначены прямоугольником с двойным подчеркиванием сторон, целевая вершина с символом «Т» – квадратная. Граничные вершины имеют символ «Г». Вершины с цифрами показывают последовательность поиска. Вершины без цифр и символов, те на которых поиск не проводится. Как только целевая вершина достигнута, поиск прекращается.

*Поиск в ширину*. Вершины раскрываются в последовательности их порождения. Поиск идет по ширине дерева, т. к. раскрытие вершины происходит вдоль одного уровня. Целевая вершина выбирается сразу же после порождения. При поиске в ширину возможно нахождение наиболее короткого пути к целевой вершине, если такой путь есть. На рис. 4 представлен граф поиска в ширину. Обозначения те же, что на рис. 3. Вершины с цифрами показывают последовательность поиска. Понятие граничной вершины в этом случае отсутствует.

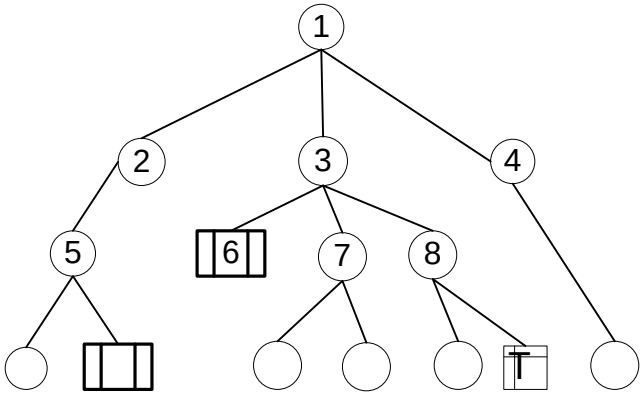


*Рис. 4.* Граф состояния, по которому осуществляется поиск в ширину

*Поиск на основе стоимости дуг.* Этот поиск применяют в сетевой модели, если имеется дополнительная информация, на основе которой дугам ставят в соответствие некоторый показатель (стоимость), чтобы оценить использование соответствующего правила. При поиске целевой вершины стремятся найти путь минимальной стоимости. Раскрытие вершин производится в порядке возрастания их стоимости. Наличие неоднозначности приводит к тому, что для одной целевой вершины может быть несколько путей. Для каждой целевой вершины определяют минимальную стоимость пути, построенного от начальной вершины до нее.

*Поиск с возвратом (бэктрекинг)* [12]. При реализации такого поиска при выборе правила определяется точка возврата, т. е. если дальнейший поиск в выбранном направлении приведет к терминальным вершинам или будет бесперспективным, то осуществляется переход к точке возврата, пройденной на ранних этапах поиска.

Далее применяется другое правило, и процесс поиска продолжается. Здесь все неудачные итерации, приведшие к тупиковой ситуации, забываются, как только применяется новое правило, и переходят к другому направлению поиска. Такой метод поиска с возвратом называется *хронологическим возвратом*. Он часто малоэффективен, т.к. не запоминает неудачные состояния и шаги поиска, встретившиеся на некотором пути. Многие из них впоследствии окажут свое отрицательное влияние при реализации других путей.

При таком поиске много полезной информации отбрасывается и не используется при анализе дальнейших направлений поиска. Необходимо анализировать шаги вывода, приведшие к тупиковым ситуациям и ошибкам. Для этого надо запоминать шаги вывода. Кроме того, возвращаться надо не к точке возврата, предшествующей данному состоянию, а к состоянию, которое вызвало неудачный ход поиска.

Для методов поиска решения при представлении пространства состояний в виде графа характерно то, что в них в основном предусматривается запоминание результатов применения нескольких последовательностей правил.

Другая особенность - они работают в пробном режиме, то есть при выборе и использовании применимого правила для какой-либо ситуации предусматривается возможность возврата к этой ситуации для применения другого правила. Достоинствами методов перебора является достаточно простая их реализация и возможность в принципе находить решение, если оно существует. Однако на практике всегда есть ограничения по времени и объему памяти на процесс реализации поиска. Для больших пространств и сложных массивов задач методы перебора неприемлемы из-за информационного «комбинаторного взрыва». Для сокращения поиска необходима дополнительная информация.

*Эвристические методы поиска.* Эти методы поиска можно использовать при наличии дополнительной информации – неких эмпирических правил, которые позволяют сокращать объем просматриваемых вариантов решений. Эвристическая информация, в частности, использует опыт, экспертные методы, аксиоматический подход. При использовании

эвристических методов поиска вершины стремятся упорядочить таким образом, чтобы процесс поиска распространился в направлениях, перспективных с точки зрения эксперта.

Для определения направления поиска используется прагматическая мера, характеризующая перспективность вершины или пути, где эта вершина находится. Такую меру называют *оценочной функцией f(n).* Эта функция является оценкой стоимости кратчайшего пути из начальной вершины в целевую при условии, что он проходит через вершину *n*. Таким образом, этот метод является вариантом метода стоимости дуг, но в этом случае оценка дуг осуществляется эмпирически, на основе накопленного опыта, а не на основе правила. При определении пути выбирается вершина с минимальным значением оценочной функции.

Недостатком метода является то, что оценочная функция должна адекватно характеризовать пространство поиска и требует большого объем знаний о проблемной области и тщательный анализ пространства состояний. На практике это снижает эффективность поиска, так как при больших графах требуются большие объемы вычислений.

Кроме того, эвристический поиск с использование оценочной функции предполагает достоверное знание пространства состояний. Однако в реальной практике информация и факты обновляются. Старые факты устаревают. Поэтому даже исходные достоверные данные становятся недостоверными. Кроме того, они являются экспертными, то есть содержат элемент субъективности. Отсюда при таком подходе приходится иметь дело с оценками недостаточно полными и определенными. В современных условиях на процесс поиска влияет дефицит времени.

Для преодоления дефицита времени используют методы, отличные от цепочек рассуждений называемых монотонными. Монотонными рассуждениями называют такие, для которых каждое последующее рассуждение е следует из предыдущего. Кроме того монотонность связана с тем, что направление результата операций зависит только от направления рассуждения.

Специалисты используют немонотонные рассуждения, или *методы качественных оценок*, основанные на известных правилах вывода, закономерностях, связях и отношениях – для уменьшения элементарных и детальных рассуждений. Одним из таких методов является метод редукции

*Метод редукции* [13]. Этот метод основан на упрощении поиска решения и сведении решения общей задачи к решению составляющих ее подзадач. Для представления исходной задачи в виде совокупности подзадач используется оператор перехода к новому описанию. Этот оператор преобразует исходную задачу таким образом, что при решении всех подзадач – преемников обеспечивается решение исходной задачи.

Для представления исходной задачи может существовать множество операторов, каждый из которых порождает свою совокупность подзадач. Часть этих подзадач может оказаться неразрешимой. Для других подзадач процесс повторяется, т.е. их в свою очередь, также сводят к подзадачам. Это продолжается до тех пор, пока каждая подзадача не будет иметь очевидное решение.

Исходная задача описывается графом состояний. Он имеет сложную структуру. *Подзадачи* рассматриваются как составные части сложного графа, которые отображают его части. Подзадачи, как элементы структуры, можно заменить дугами в общей схеме, вследствие чего структура исходного графа существенно упрощается.

Процесс преобразования также удобно описывать с помощью графовых структур. Процесс поиска решения исходной задачи при таком описании представляет собой направленный *граф редукции задач.*

На рис. 5 представлен упрощенный фрагмент (только три уровня) графа задачи S, которая сводится к решению задач 1, 2, 3, 4, 5.

Решение 1 неизвестно – знак тупиковой вершины. Решение 4 существует – знак истинной вершины. Решения 2, 3, 5 не определены (кружочки) сводятся к решению задач 6, 7, 8, 9, 10, 11.

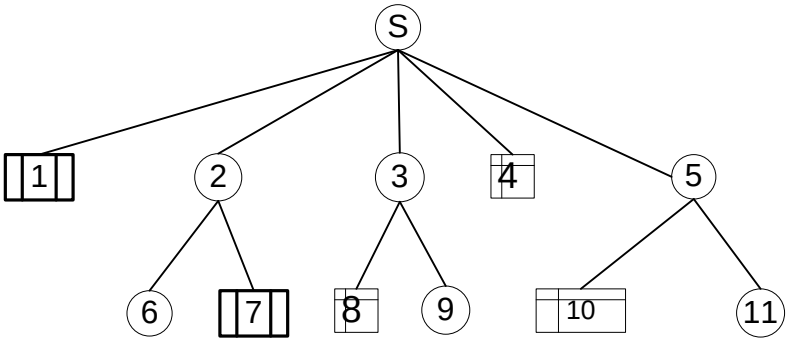


*Рис. 5.* Фрагмент исходного графа состояний

Решения 6, 9, 11 не определены (кружочки). Решения 8, 10 истинны, они определяют решение 3 и 5. Решение 7 тупиковое – неизвестно.

На рис 6 представлен граф редукции этой задачи.

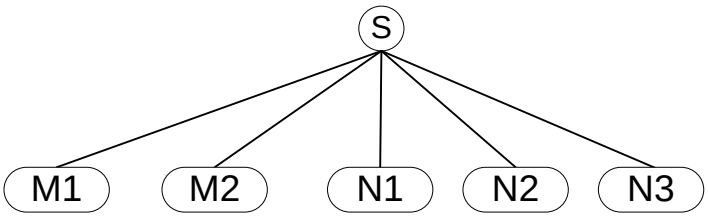


*Рис. 6.* Граф редукции задач

В результате редукции исходный граф упрощается и сводится к двум новым задачам M1, M2 и трем задачам с известными решениями N1, N2, N3. Существующее пространство состояний обеспечивает частичное решение задачи S. Это решение может либо удовлетворить, либо не удовлетворить. В последнем случае требуется найти решение задач M1, M2, что приведет к полному решению задачи S.

## Заключение

В работе затронуты актуальные вопросы, связанные с реализацией методов моделирования выводов для интеллектуальных систем типа ИСППР. Рассмотрены основные способы моделирования выводов в плане их применения в ИСППР. Основное внимание уделено проблеме топологического построения моделей выводов. Вывод по правилам является привлекательным, потому что он подразумевает наличие формализованной задачи, для которой существуют логические методы поиска области истинности решения. Однако это достоинство и содержит недостаток – требование формализации области вывода. Метод применим, если возможно формализация. Поэтому для систем поддержки принятия решений различают два направления в развитии логического вывода знаний [14]: развитие систем логического вывода, основанного на правилах; развитие систем логического вывода, основанного на прецедентах. Второе направление является альтернативой, рассмотренному в данной статье, и позволяет решать задачи со слабо формализуемой областью.

## Примечания:

1. Cheng A.Y., Cheng Y.L. Intelligent real-time graphic-object to database linking-actuator for enabling intuitive on-screen changes and control of system configuration: пат. 5706453 США. 1998.

2. Маркелов В.М., Соловьёв И.В., Цветков В.Я. Интеллектуальные транспортные системы как инструмент управления // Государственный советник. 2014. №3. С. 42-49.

3. Цветков В.Я., Розенберг И.Н. Интеллектуальные транспортные системы – LAP LAMBERT Academic Publishing GmbH & Co. KG, Saarbrücken, Germany 2012. 297 с.

4. Guerlain S., Brown D. E., Mastrangelo C. Intelligent decision support systems // Systems, Man, and Cybernetics, 2000 IEEE International Conference on. IEEE, 2000. T. 3. p. 1934-1938.
5. Pearl J. Causality: models, reasoning and inference. Cambridge : MIT press, 2000.
6. Suzko A. A., Zakhariev B. N. Direct and inverse problems. Springer, 1990.
7. Doyle J.C. et al. State-space solutions to standard H 2 and H∞ control problems //Automatic Control, IEEE Transactions on. 1989. T. 34. №. 8. p. 831-847.
8. Collins A., Michalski R. The logic of plausible reasoning: A core theory // Cognitive science. – 1989. T. 13. №. 1. p. 1-49.
9. Tsvetkov V.Ya. Dichotomous Systemic Analysis**.** Life Science Journal 2014; 11(6). Pp. 586-590.
10. Elsukov P.Y. Formation model of the structural model for the management of energy saving // Herald MSTU MIREA HERALD. 2014. № 3 (4). p. 135-145.
11. Slaney J., Fujita M., Stickel M. Automated reasoning and exhaustive search: Quasigroup existence problems //Computers & mathematics with applications. 1995. T. 29. №. 2. p. 115-132.
12. Ginsberg M.L. Dynamic backtracking //arXiv preprint cs/9308101. 1993.
13. Kutcher G.J. et al. Histogram reduction method for calculating complication probabilities for three-dimensional treatment planning evaluations // International Journal of Radiation Oncology* Biology* Physics. 1991. T. 21. №. 1. C. 137-146.
14. Каменнова М.С. Корпоративные информационные системы: технологии и решения. // Системы управления базами данных. 1995. № 3. с. 88-99.

**References:**

1. Cheng A. Y., Cheng Y. L. Intelligent real-time graphic-object to database linking-actuator for enabling intuitive on-screen changes and control of system configuration: пат. 5706453 США. 1998.
2. Markelov V.M., Soloviev B.V., Tsvetkov V. Ya. Intelligent transport systems as a management tool // State Advisor. 2014. №3. p.42-49.
3. Tsvetkov V. Ya., Rosenberg I,N. Intelligent Transport Systems – LAP LAMBERT Academic Publishing GmbH & Co. KG, Saarbrücken, Germany 2012. 297 p.
4. Guerlain S., Brown D. E., Mastrangelo C. Intelligent decision support systems //Systems, Man, and Cybernetics, 2000 IEEE International Conference on. IEEE, 2000. T. 3. p. 1934-1938.
5. Pearl J. Causality: models, reasoning and inference. Cambridge: MIT press, 2000.
6. Suzko A. A., Zakhariev B. N. Direct and inverse problems. Springer, 1990.
7. Doyle J. C. et al. State-space solutions to standard H 2 and H∞ control problems //Automatic Control, IEEE Transactions on. 1989. T. 34. № 8. p. 831-847
8. Collins A., Michalski R. The logic of plausible reasoning: A core theory //cognitive science. 1989. T. 13. №. 1. p. 1-49.
9. Tsvetkov V.Ya. Dichotomous Systemic Analysis**.** Life Science Journal 2014; 11(6). Pp. 586-590.
10. Elsukov P. Y. Formation model of the structural model for the management of energy saving // Herald MSTU MIREA HERALD. 2014. № 3 (4). c. 135-145.
11. Slaney J., Fujita M., Stickel M. Automated reasoning and exhaustive search: Quasigroup existence problems //Computers & mathematics with applications. 1995. T. 29. №. 2. p.115-132.
12. Ginsberg M. L. Dynamic backtracking //arXiv preprint cs/9308101. 1993.
13. Kutcher G. J. et al. Histogram reduction method for calculating complication probabilities for three-dimensional treatment planning evaluations //International Journal of Radiation Oncology* Biology* Physics. 1991. T. 21. №. 1. C. 137-146.
14. Kamennova MS Corporate information systems: technologies and solutions. // Database management system. 1995. № 3. p. 88-99.

УДК 004.8

# Вывод в интеллектуальных системах

Виктор Яковлевич Цветков

Московский государственный технический университет радиотехники, электроники и автоматики МГТУ МИРЭА, Российская Федерация
Доктор технических наук, профессор
E-mail: cvj2@mail.ru

**Аннотация.** Статья описывает методы автоматизированного получения решений в интеллектуальных системах. Статья описывает методы вывода на основе прямой и обратной цепочек. Показано различие между методом прямой цепочки и методом обратной цепочки. Статья описывает методы поиска решений в пространстве состояний. Дается различие между поиском в глубину и поиском в ширину. Описан метод бэктрекинга. Раскрывается содержание метода редукции. Показаны границы применимости методов и дается сравнение описанных методов с методом прецедентов.

**Ключевые слова**: искусственный интеллект; правила вывода; графы; декомпозиция; решение проблем; формализация; интеллектуальные системы; интеллектуальные транспортные системы.