

Формализация знаний и управление на основе позитивно-образованных языков

С.Н. Васильев

Аннотация. Рассматриваются возможности автоматизации процессов управления и расширения потенциала управления с помощью некоторых известных и новых методов представления и обработки знаний. Предлагаются оригинальные логические средства (позитивные исчисления). Отмечены их достоинства и ограничения в сравнении с известными, в том числе нелогичными, средствами искусственного интеллекта. Рассматривается применение этих средств для моделирования интеллектов управления динамическими системами.

Введение

Интеллектуальное управление и место логического подхода в разнообразии средств искусственного интеллекта

Наблюдаемое в технике вытеснение многих традиционных технических средств автоматизации цифровыми преобразователями придает системам управления характер *информационно-управляющих систем*, а встраивание в них процедур распознавания образов, аккумуляции экспериментального знания, планирования действий и других “интеллектуальных” функций подталкивает разработчиков называть их *системами интеллектуального управления*. Однако они, обладая все же недостаточным уровнем интеллектуальности, пока не оправдывают в полной мере своего названия [1].

В статье обсуждаются вопросы повышения потенциала управления этих систем с применением оригинальных результатов в области логического подхода к представлению знаний в системах управления и формированию целенаправленных управлений (команд, планов действия и т.д.). Приводимые нами результаты мы относим к так называемым системам *интел-*

лектного управления (СИУ) [2], т.е. системам автоматического управления на базе тех или иных средств искусственного интеллекта (ИИ), но пока без претензии на автоматизацию самого процесса целеполагания. При этом реализуется только автоматизация достижения цели управления, задаваемой человеком-создателем системы и объекта управления, хотя и эта задача сегодня - достаточно амбициозна. В отличие от принятого в русско-язычной литературе, термин «*интеллектуальное управление*» резервируется нами для будущих систем интеллектуального управления с *автоматическим целеполаганием*. Сравнение разных средств ИИ дано в таблице.

Вообще, использование средств ИИ расширяет потенциал проектирования и управления динамическими системами; охватываются задачи:

- с неизвестными или уже несправедливыми с некоторого момента эксплуатации уравнениями динамики (как в задачах выбора эффективных каналов управления в объектах с реконфигурацией),

- задачи, в которых модели в форме уравнений динамики могут либо уступать по эффективности своего использования моделям ИИ (как в задачах планирования действий в среде), либо полезны в комбинации с моделями ИИ.

Сравнение средств искусственного интеллекта

Средства ИИ	Типовые достоинства	Типовые недостатки
I. Нейросетевые	1. Применение в многофакторных проблемах с плохой формализуемостью закономерностей. 2. Высокая степень распараллеливаемости (быстродействие). 3. Способность к обучению.	1. Необходимость обучающей информации – представительного набора примеров «вход-выход». 2. Медленность обучения.
II. Эволюционные (генетические)	1. Высокая степень распараллеливаемости (быстродействие).	1. Априорная неизвестность степени эффективности в конкретном приложении. 2. Скорее самоорганизация природной стихии, чем интеллектуальный (творческий) процесс.
III. Продукционные	1. Возможность представления дескриптивно-конструктивных знаний и рефлексии. 2. Естественность правил («если -то»).	1. Сложность исполнения больших баз правил, недостаточная структурируемость. 2. Сложность обеспечения корректности выводов.
IV. Объектно-ориентированные (семантические сети, фреймы и т.д.)	1. Хорошая структурируемость. 2. Высокое быстродействие механизмов наследования свойств, умолчания и т.д.	1. Сложность программирования (отход от идеалов ИИ). 2. Недостаточная выразительность.
V. Логические	1. Высокая выразительная сила. 2. Корректность. 3. Высокая сложность решаемых офлайн-задач.	1. Недостаточное быстродействие, традиционные приложения – офлайн. 2. Традиционная плохая совместимость с эвристиками и опытом. 3. Неразрешимость богатых логик. 4. Недостаточность одной логики.
VI. Объектно-логические	1. Объединение достоинств объектно-ориентированных и логических.	1. Недостатки логических моделей. 2. Сложность программирования.

Традиционные системы управления, основываясь на уравнениях динамики, являются системами алгоритмического типа. Искусственные нейронные сети, получившие в 80-е гг. свое «второе дыхание» благодаря новым алгоритмам обучения (backpropagation), образуют первый уровень интеллектуального управления с высокой степенью распараллеливаемости обработки информации. Нейронные сети применимы для решения целого класса задач, где используются не уравнения динамики и даже не столько правила, как в традиционных экспертных системах, сколько опыт. Опытный врач поставит диагноз, даже если симптоматика заболевания искажена и ему не доводилось ранее ее наблюдать или абстрактно осмысливать. Механизм диагностики при этом иногда напоминает скорее распознавание на подсознательном уровне, чем цепочку рассуждений на логическом уровне. Построение экспертных систем, основанных на правилах, или использование логических исчислений для решения указанного класса задач упирается в сложность учета всех мыслимых сочетаний факторов и формализации закономерностей, связывающих условия задачи с результатом.

Алгоритм преобразования входных сигналов нейросети в выходные порождается процессом предварительной настройки (обучения) путем предъявления примеров пар «вход-выход», поэтому можно говорить о высокопроизводительном, но не более, чем рефлекторном, характере функционирования искусственных нейросетей.

Обладая высокой распараллеливаемостью, нейронные сети обеспечивают повышенное быстродействие, важное в задачах управления в реальном времени, т.е. когда идентификация или формирование закона управления осуществляются в темпе протекания процесса. Вместе с тем искусственные нейронные сети сегодняшнего дня моделируют скорее, например, глаз, чем мозг.

Аналогично, генетические алгоритмы моделируют самоорганизацию (дарвиновского типа) природной стихии, а не творческий процесс. Сказанное никак не умаляет значения нейроуправления и эволюционных алгоритмов самоорганизации. В частности, в силу своей реактивности и способности к обучению они полезны уже в современном их состоянии развития при создании многоуровневых и многофункциональных систем управления с элементами ИИ.

В ряде задач управления, в которых зависимости настолько сложны, что не допускают своего обычного аналитического представления, существенная роль принадлежит экспертным суждениям и знаниям человека, заставляя в дополнение к количественным методам или вместо них применять логический или логико-лингвистический подходы, в соответствии с которыми в качестве значений переменных допускаются не только числа, но и слова или предложения искусственного или естественного языка [3].

По своему замыслу, такие системы должны обладать способностью рассуждать и выводить заключения на основе знания некоторых общих закономерностей мира, сведений о текущем и целевом состоянии объекта управления и т.п. Такие системы не совсем удачно, но принято называть системами, основанными на знаниях (СОЗ). Так как интуитивные представления человека о логичности умозаключений представимы, например, в логических языках, то представление знаний в некотором логическом языке и их обработка с помощью специальных логических средств позволяют получать те или иные предпочтения на множестве допустимых управлений с целью выбора одного из них.

Логические исчисления не предписывают жесткой последовательности действий, а предполагают определенную свободу выбора в рамках соответствующего "исчисления возможностей". В общем случае СОЗ оперируют с более широкой информацией, в частности, объектно-ориентированными представлениями, тоже основанными на знаниях экспертов (семантические сети, фреймы и др.). Вместе с тем СОЗ могут использовать и традиционные алгоритмы, базирующиеся на уравнениях динамики. Поэтому *класс решаемых задач принципиально расширяется* по сравнению с традиционной проблематикой теории управления.

К СОЗ мы относим, прежде всего, следующие:

1) системы, основанные на инструктивных знаниях, т.е. на *правилах* "если-то" (Rule-Based Reasoning),

2) системы, основанные на *автоматическом доказательстве теорем* (АДТ, Automatic Theorem-Proving Techniques),

3) системы, основанные на *автоматическом гипотезировании* (Automatic Hypothesizing), т.е. на порождении гипотез,

4) системы, основанные на *рассуждениях по аналогии* (Analogical Reasoning),

5) *объектно-ориентированные* интеллектуальные системы (Object-Oriented Intelligent Systems),

6) *объектно-логические* системы, сочетающие некоторые достоинства последних систем с системами АДТ и использующие объектно-логические языки, фреймовые логики (F-logics), логики транзакций (Transaction Logics) и т.д. [4-6].

В СОЗ реализуется относительно высокоуровневый процесс умозаключений, а современные искусственные нейронные сети и генетические алгоритмы нами не включаются в понятие "СОЗ", хотя их роль (сама по себе и в составе комплексных многоуровневых систем управления) является весьма полезной.

Так, обрабатывая дополнительно информацию, полученную на выходе нейронных сетей, с помощью логических средств обработки знаний, можно обеспечить более высокоуровневую обработку информации. Характерные примеры – автоматическое двухуровневое распознавание образов (например, автомашин с борта летательного аппарата) или двухуровневое управление роботом в условиях других планет [7], когда экспертная система обеспечивает робастность функционирования, а нейронная сеть – тонкую настройку на специфические условия планеты, которые заранее с достаточной точностью не известны (если произойдут их резкие изменения, то система вновь передает управление экспертной системе и процесс обучения начинается заново).

Уровень обработки инструктивных знаний является, по степени своей интеллектуальности, промежуточным между нейроуправлением и высшим уровнем – уровнем автоматического осуществления умозаключений в мощных логиках (см. рисунок).

Современные СИУ, использующие интеллектуальные компоненты даже неалгоритмического типа (СОЗ), проявляют недостаток интеллектуальности в сложных задачах управления. Рассмотрим причины на примере экспертных систем (относимых нами к неалгоритмическим СИУ).



Обычно знания экспертной системы организуются в форме инструктивного знания (правил “если-то”, Rule-Based Expert Systems). Однако в ряде случаев количество правил оказывается настолько большим, что их совокупность становится трудно исполнимой, теряется целенаправленность процесса формирования управления, надежность (корректность). В случае, когда набор правил рефлексивен, т.е. в правых частях некоторых из них фигурируют действия по изменению самого набора правил, большая ответственность возлагается на разработчиков по верификации и обеспечению корректности функционирования экспертной системы. Нужны более мощные системы “обдумывания ситуации”, обрабатывающие знания более общего вида вместо простого использования правил, заранее заготовленных или модифицируемых в процессе функционирования мало надежным способом.

Использование объектно-ориентированных экспертных систем упрощает структурирование знаний, но заметно усложняет программирование экспертной системы.

Системы АДТ работают обычно в полной первопорядковой логике и существенно превосходят сегодня другие средства ИИ с точки зрения сложности теорем, доказываемых ими. Решены некоторые открытые задачи, поставленные известными математиками [8, 9], причем планка сложности неуклонно повышается.

С другой стороны, эти достижения в области систем АДТ принадлежат сфере офлайн-задач, т.е. задач без характерных ресурсных ограничений: это – сфера математических задач, программирования и вообще автоматизации строгих рассуждений, допускающих формализацию в некотором логическом языке, когда нет особой критичности в вопросе о ресурсах, отпущенных на проведение автоматических выкладок.

Последнее не означает, что методы АДТ принципиально не применимы к задачам реального времени. Например, так же, как и в логическом программировании [10], известны попытки применить АДТ в управлении движущимися объектами для построения полностью автономных систем. Первым примером

такого типа являлась система управления мобильным интегральным роботом STRIPS – самоходным аппаратом, совершающим передвижения в упрощенной среде по формируемым командам [11, 12]. В работе [13] полное исчисление предикатов было использовано для планирования с учетом новых для того времени продвижений в АДТ.

Следует заметить, что, по крайней мере, в отечественной литературе на возможность и перспективность использования автоматического доказательства теорем в теории и практике управления указал еще в 1960 г. А.А. Фельдбаум [14].

На наш взгляд, проникновение полных первопорядковых логик и методов автоматизации логического вывода (в частности, в форме АДТ) в сферу интересов и компетентности специалистов по теории управления является весьма существенным, позволяя развить теорию управления в направлении расширения потенциала решения сложных задач управления, увеличения степени автоматизации, создания СИУ, в большей степени оправдывающих свое название благодаря повышению уровня их интеллектуальности. При этом теория управления и искусственный интеллект становятся значительно более совместимыми, поскольку предикатные языки (1-го и более высоких порядков) существенно более выразительны и охватывают вещественные и другие переменные. Грубо говоря, выразительная сила предикатных языков так соотносится с выразительной силой булевских (пропозициональных) языков, как выразительная сила языка математической теории управления соотносится с выразительной силой языка двоичной арифметики. Предикатные языки позволяют формализовать более широкие знания для последующей обработки машинно-ориентированными правилами. Каждый шаг вывода формализует не столько специальный (локальный) переход от условий к действию (как при использовании инструктивных знаний), выражающий более или менее очевидным образом отдельный шаг приближения к цели управления, сколько интуитивное представление человека о правильности (логичности) умозаключений, т.е. имеет более универсальный характер.

Вместе с тем в АДТ, как в математически строгой части направления “General Reasoning” (“Рассуждения общего вида”, т.е. рассуждения, не стесненные, например, сильно ограниченным синтаксисом представления знаний в системах, основанных на правилах), остаются и дают знать о себе старые проблемы, породившие разные новые концепции организации компьютерных рассуждений: немонотонные логики (Nonmonotonic Logics), неформальные логики (Informal Logics), аргументирование (Argumentation), риторика (Rhetoric), пересмотр взглядов (Change of View), практические рассуждения (Practical Reasoning) и др. [15].

Неустранимо принципиальное противоречие между выразительностью формальных языков и разрешимостью проблемы дедукции (А. Тьюринг, А. Черч). Более того, даже теоретически разрешимые теории или их фрагменты могут быть практически неразрешимы по причине высокой сложности вычислений. Несмотря на вот уже 40 лет декларируемую важность эвристики для поиска вывода, а также опыта, в том числе ставшего алгоритмическим знанием, нет понимания механизмов их усвоения и использования. В частности, из-за эвристик может теряться полнота, но не должна теряться корректность. Необходимы новые идеи для придания “второго дыхания” системам автоматизации рассуждений.

Средствами так называемых позитивных логик, обсуждаемых далее, удалось обеспечить “практическую разрешимость” (в смысле В.М. Глушкова), когда в дополнение к теоретической универсальности (полноте) или вместо нее система автоматизации дедукции обладает в конкретной, может быть, достаточно узкой, предметной области “творческими способностями”, соизмеримыми с интеллектом специалистов рассматриваемой области. При этом большая роль отводится эффективной совместности логики с эвристиками и адаптируемости логики под специфику предметной области. Свойство практической разрешимости обнаруживается здесь на конкретных задачах интеллектуального управления динамическими системами и, в частности, проявляется в преодолении проблем немонотонности, рекурсивности, обеспечения конструктивности выводов и др.

1. Представление и обработка знаний на основе позитивно-образованных языков

Рассмотрим суть так называемой позитивной логики. Она использует разработанный в [2] язык позитивно-образованных формул (по-формул). Он лучше приспособлен, чем известные языки, для усвоения как эвристик, так и опыта, поскольку достаточно однороден и в то же время хорошо структурирован. Это - полный язык первого порядка, формулы которого представляются как деревья: каждый узел суть позитивный квантор, т.е. выражение, содержащее последовательно знак квантора \forall или \exists , множество индивидуальных переменных и условие на эти переменные, именуемое конъюнктом. Конъюнкт - это множество атомов или *False* (рассматривается случай отсутствия функциональных символов; более общий случай см. в [16]). По определению, $A \subseteq False$ для любого конъюнкта A . По каждой ветви знаки кванторов чередуются. Пустое A обозначается *True*.

Семантика \forall -узлов такова: по-формула $\forall x_1 \dots x_m : A(F_1, \dots, F_n)$, $n \geq 1$, понимается в исчислении предикатов (ИП) как формула $\forall x_1 \dots \forall x_m (A^{\&} \rightarrow F_1^{III} \vee \dots \vee F_n^{III})$, где F_i^{III} - образ по-формулы F_i в ИП, $A^{\&}$ совпадает с $\&\{a \mid a \in A\}$, если $A \neq True, False$, иначе $A^{\&} = A$. При $n = 0$ получаем просто формулу $\forall x_1 \dots \forall x_m (A^{\&} \rightarrow False)$. Семантика \exists -узлов определяется аналогично (с заменой \forall на \exists , а обеих связок \rightarrow, \vee на $\&$); при $n \geq 2$ получаем конъюнктивное ветвление, а при $n = 0 : \exists x_1 \dots \exists x_m A^{\&}$.

Без ограничения общности можно считать, что всякая по-формула F начинается с корня $\forall True$ (дизъюнкция при $n \geq 2$), и рассматривать ее как непустое множество деревьев с экзистенциальными корнями и листьями.

Позитивное исчисление J суть пара $\langle \exists False, a \rangle$, где $\exists False$ - аксиома, a - правило вывода. В качестве аксиомы выбрано противоречие $((\exists False)^{III} = False)$, поскольку всякий J -вывод по-формулы F будет опровержением $(F)^{III}$.

Узлы $\{\forall \bar{y} : B_i\}$, следующие непосредственно за корнем (*базой*) $\exists \bar{x} : A$, называются *вопросами*, и, если $B_i \gamma \subseteq A$, где $\gamma : \bar{y}_i - \bar{x}$ - подстановка, то говорим, что вопрос B_i имеет *ответ* γ . Правило вывода a в J состоит в использовании ответа. *Выводом* называется последовательность по-формул, начинающаяся с F и заканчивающаяся аксиомой $\exists False$, а каждое применение правила вывода соответствует переходу от рассматриваемой формулы к непосредственно следующей по правилу a . Таким образом, поиск вывода можно представить как некоторую вопросно-ответную процедуру.

Пусть $F = \{\exists \bar{x} : A(\Phi), \Psi\} \in L$, $A \neq False$, Ψ может быть пустым. *Правило вывода* a , примененное к F , дает результат $a F$ и имеет вид

$$\omega F / F = \exists \bar{x} \cup \bar{z}_i^* : A \cup A_i \Theta^* \quad (\Phi, \Phi_i \Theta^*), \\ i = \overline{1, n}, \Psi / \exists \bar{x} : A(\Phi), \Psi$$

где вопрос

$\forall \bar{y} : B(\exists \bar{z}_1 : A_1(\Phi_1), \dots, \exists \bar{z}_n : A_n(\Phi_n)) \in \Phi$ и подстановка (ответ) $\epsilon : \bar{y} - \bar{x}$ таковы, что $B\Theta \subseteq A$, $\Theta^* = \Theta \cup (\bigcup_{p=1, n} \{\bar{z}_p \rightarrow \bar{z}_p^*\})$, а множе-

ства \bar{z}_p^* состоят из новых, различных переменных.

Используются упрощения $\exists False / \exists X : A \cup False(\Phi)$, $\Psi / (\exists False, \Psi)$, если Ψ - непусто.

Теорема 1 [2]. Исчисление J корректно и полно.

Иначе говоря, во-первых, для любой по-формулы F имеет место логическая эквивалентность $(F)^{III} \leftrightarrow (\omega F)^{III}$, а во-вторых, F выводимо в J тогда и только тогда, когда $\neg(F)^{III}$ общезначимо.

2. Специфика языка и исчисления по-формул

Хотя семантика по-формул очень прозрачная и определяется как обычная (классическая, теоретико-модельная) семантика соответствующих образов этих формул в ИП,

по-формулы довольно необычны по своей форме, а по-формализм в целом имеет ряд важных особенностей, расширяющих потенциал логического подхода к представлению и обработке знаний и определяющих его эффективность в приложениях (п.п. 7-10). Рассмотрим эти особенности.

1). По-формула имеет *крупно-блочную структуру*, все элементы которой суть *только позитивные кванторы* всеобщности и существования. Тем не менее, язык по-формул полон относительно выразительности первопорядковых языков.

2). По-представление для первопорядковых формул *более компактно*, чем в языке дизъюнктов (иначе говоря, в языке скулемовских стандартных форм, образуемых из предваренных нормальных форм специальным удалением кванторов существования, что и приводит к дизъюнктам), используемом методом резолюций [17-19]. Это же верно и для пропозициональных формул – в сравнении с представлениями в стандартных нормальных формах (дизъюнктивных и конъюнктивных).

3). По-формула имеет *простую и регулярную структуру*, т.е. при считывании текста формулы она обладает в определенном смысле предсказуемостью структуры ввиду поочередного появления в каждой ветви \exists - и \forall -узлов.

4). Отрицание по-формулы получается *просто инвертированием* символов \exists, \forall .

5). Нет необходимости предварительно в исходной первопорядковой формуле удалять кванторы существования процедурой скулемизации (как это требует применение метода резолюций), увеличивающей сложность термов и в целом формулы.

6). В по-формализме исходная эвристическая структура знания *сохраняется лучше* благодаря тому, что естественно-языковое представление знаний никогда не использует “теоретических” кванторов $\forall x, \exists x$ и, наоборот, широко задействует типовые кванторы $\forall x(A _)$, $\exists x(A \& _)$. Использование в по-формулах позитивных кванторов как частного случая типовых кванторов означает, что выражения A имеют очень простой (и эффективный для обработки по-формул) вид (конъюнкции атомов). По-структура особенно близка к

исходной структуре знания, если оно записано как выражение, образованное из литералов (атомов и/или их отрицаний) с помощью позитивных кванторов и логических связок $\&, \vee$, т.к. в этом случае по-структура может отличаться от исходной просто появлением кванторов $\exists True, \exists False$ (вместо упомянутых связок $\&, \vee$) и/или заменой отрицаний атомов $\neg A$ подформулой $\forall A \exists False$.

7). Исчисление J имеет *единственное, унарное и крупно-блочное правило вывода*, благодаря чему размерность комбинаторного пространства поиска выводов уменьшается в сравнении с правилом резолюирования в методе резолюций, которое хотя тоже является единственным правилом, но бинарное и мелко-блочное.

8). Техника вывода (опровержения) анализирует ближайшую окрестность корня по-формулы, т.е. только корень (базу) и непосредственно следующие узлы структуры (вопросы к корню). Это оказывается возможным благодаря особенностям 1, 3, и позволяет *сфокусировать внимание* без потери свойства полноты вывода (в смысле теоремы 1).

9). Техника вывода может формулироваться в *содержательных терминах вопросно-ответной процедуры* вместо технических терминов формальной выводимости (т.е. в терминах логических связок, атомов и т.п.).

10). Благодаря особенностям 1, 3, 6, 8, 9 техника вывода *хорошо совместима с эвристиками* конкретных приложений, а также с общими эвристиками управления выводом. Благодаря особенности 7, процесс вывода состоит из *крупно-блочных шагов, хорошо наблюдаем и управляем*.

11). Техника вывода допускает *естественный ИЛИ-параллелизм*, т.к. опровержения по ветвям осуществляются независимо друг от друга.

12). Выводы получаются *легко интерпретируемыми человеком*, благодаря особенностям 9-11, 1-3, 6-8. Это важно в приложениях, допускающих человеко-машинный режим.

13). Базовое исчисление J обладает свойством *простой и сильной модифицируемости*: простым ограничением применения правила вывода может сильно варьироваться семантика

логики J (конструктивность, немонотонность, темпоральность и др.), что вместе с другими особенностями (и, в первую очередь, 10) весьма важно для приложений (п.п. 7-9).

14). Описанные язык L и исчисление J приведены в варианте без функциональных символов и допускают переформулировку на общий случай. Вместе с тем вопросы работы с равенствами, поддержки индукции, абдукции и некоторые другие логические вычисления пока остались неисследованными, хотя уже разработан метод гипотезирования [20], совмещающий возможности разработанных дедуктивных средств с решением первопорядковых логических уравнений и похожий на процедуры абдукции и генерализации первопорядковых формул (п. 10).

15). Организация вопросно-ответной процедуры зависит от класса решаемых задач. Примеры комбинаторно сложных задач, трудных для метода резолюций и просто решенных в по-формализме, реализованном в виде программной системы КВАНТ/1 (автор - Е.А. Черкашин), изложены в [2].

Заметим, что одними из первых отечественных дедуктивных систем были разработки, основанные на алгоритме Н.А. Шанина, алгоритме очевидности (ИК АН УССР) и обратном методе С.Ю. Маслова. В частности, дедуктивная компонента системы обработки математических текстов, разрабатывавшейся под руководством В.М. Глушкова [21], базировалась, помимо метода резолюций и других алгоритмов, на алгоритме очевидности.

Концептуально алгоритм очевидности ориентирован на проверку правильности математических текстов, написанных в языке TL , если доказательства *достаточно подробны*, и отыскание в них пробелов, требующих «расшифровки», в случае недостаточной детализации доказательств; поиск же доказательств сложных теорем должен выполняться в *диалоговом* режиме [22]. Язык TL [23, 24], по замыслу разработчиков, должен был быть близок к естественному математическому языку и удобным в человеко-машинных системах поиска выводов, формульных преобразований, построения примеров (моделей), проверки корректности разного рода конструкций (алгоритмов, определений и т.п.).

Формализация же используемых выше понятий очевидности и сложности доказательств производится с учетом особенностей реализации на ЭВМ [25]. Так, самая бедная реализация алгоритма очевидности может состоять в том, что в системе имеется явно выписанный список доказанных теорем, и выявление того, очевидна ли некоторая теорема или нет, состоит в ее нахождении в списке. Однако в силу технических ограничений на длину списка оказывается необходимым хранить доказательства многих теорем этого списка в потенции, с помощью специальных алгоритмов, которые в случае необходимости могут проверить их на выводимость. Таким образом, содержательно алгоритм очевидности должен представлять собой, по замыслу, *совокупность алгоритмов разрешения разрешимых фрагментов рассматриваемых математических теорий*. Алгоритм очевидности имитирует действия человека в поиске выводов *нижней* ступени [26], реализуя простейшие теоретико-множественные, алгебраические и логические преобразования. При этом, алгоритм очевидности работает в среде, основными элементами которой являются дерево целей и дерево посылок. Последнее несколько напоминает алгоритмы «естественного вывода» на базе секвенциальных исчислений (генценовского типа), в частности, алгоритм Н.А. Шанина [27], предложенный для автоматизации вывода теорем в исчислении высказываний.

В целом, алгоритм очевидности представляет собой специальным образом организованную систему алгоритмов, которая, по замыслу ее создателей, должна развиваться во времени как в сторону допустимого для машины увеличения емкости понятия «очевидности», так и с целью обеспечения удобства в обслуживании специалиста, пользующегося в диалоговом режиме алгоритмом очевидности как инструментом построения доказательств теорем и выполнения других работ по обработке математических текстов [28]. Это – *человеко-машинная* система, призванная в диалоге совмещать наиболее сильные стороны человека и машины.

Что касается обратного метода С.Ю. Маслова [29-34], то он (как и по-формализм, а также метод резолюций) относится к большому кругу работ по созданию *самого математического*

аппарата автоматического доказательства теорем на основе логических или логико-математических исчислений и процедур поиска вывода в этих исчислениях. Сегодня, так или иначе, большинство существующих программ поиска доказательств по существу основывается на идеях метода резолюций или обратного метода. Эти (т.н. локальные) методы предназначены для установления выводимости теорем в исчислении предикатов и являются основными методами локализации поиска (ограничения окрестностей поиска в дереве выводов), когда каждый шаг поиска достаточно осуществлять не во всем дереве выводов, а лишь в некоторой его части. Обратный метод был предложен одновременно с методом резолюций и независимо от него [34]. Будучи первоначально предназначенным для классического исчисления предикатов, он на скелемовских стандартных формах примерно эквивалентен обратному методу [35, 34]. Это подтверждается теоретическими оценками и экспериментом (ссылки в [34]). Однако, как отмечается в [34], обратный метод применим к секвенциям произвольной структуры и распространим на произвольные дедуктивные (не обязательно логические) секвенциальные исчисления без правил сечения. Он имеет более широкую область применения и не только доказывает теоремы, но в некоторых случаях (как и по-формализм) выявляет и невыводимость формулы. В по-формализме необходимым условием выводимости по-формулы является наличие среди висячих вершин хотя бы одного узла $\exists False$. В контексте исчисления предикатов (в силу примерной эквивалентности в этом случае метода резолюций и обратного метода), то, что говорилось ранее в плане сравнения по-формализма с методом резолюций, распространяемо и на обратный метод.

3. Конструктивный вариант позитивной логики

Известно, что логики с классической семантикой полезны в решении задач интеллектуального управления, но чаще всего – в сочетании с конструктивными логиками. В некоторых логиках эти семантики (классическая и конструктивная) просто совпадают. Примером является логика ПРОЛОГ-систем. В терминах языка L исчисления

логики J можно указать более широкий конструктивный фрагмент первопорядковых логик, определяемый следующим утверждением [2] о конструктивности некоторых J -выводов – в смысле их преобразуемости в интуиционистский вывод, например, в секвенциальном исчислении LJ [36].

Теорема 2 (о конструктивном фрагменте исчисления J). Для любой задачи, специфицируемой первопорядковой формулой $G_1 - G_2$, где G_1 – произвольная формула (описывает условия и конструктивные средства решения задачи), а G_2 – цель, допускающая в по-формализме представление вида $(G_2)^{PO} = \forall x : A (\exists \bar{y}_1 : B_1, \dots, \exists \bar{y}_n : B_n)$, J -вывод формулы $\exists True ((G_1)^{PO}, (\neg G_2)^{PO})$ конструктивен.

Таким образом, ограничение применения a только формулами, указанными в теореме 2, обеспечивает свойство конструктивности получаемых выводов.

Этот факт использован при построении информационно-управляющей системы наведения телескопа на центр планеты в неполной фазе [37] в условиях длительного слежения, когда изменения фазы, положения оси симметрии изображения планеты в фокальной плоскости телескопа и углового размера диска планеты существенны, что приводит к потере некоторыми датчиками свойства информативности и необходимости логической диагностики их информативности, структурной реконфигурации измерительной системы с последующим конструктивным логическим выводом управляющих воздействий на шаговые двигатели исполнительной системы.

Соответствующий режим многошагового наведения и слежения имеет характер скользящего режима (подробнее см. в [37]), а соответствующая система интеллектуального управления с точки зрения динамичности знания принадлежит классу систем, управляемых временем (Time-Driven Reasoning), так как все знания периодически и полностью обновляются. Конкретные значения полученных в результате измерений и вычислений данных не изменяют стратегии логического вывода. Поэтому нет

необходимости в использовании каких-либо специальных временных связей или дополнительной переменной для времени (такие подходы развиваются в так называемых временных (темпоральных) логиках [38, 39]).

Рассмотрим следующий простой пример использования по-формализма в пропозициональном фрагменте.

Пример. Рассмотрим задачу структурной реконфигурации системы управления ориентацией космического аппарата после обнаружения и локализации отказа, например, датчика угла рысканья (f_3). Пусть два других угловых датчика (f_1, f_2) и три датчика скорости (g_1, g_2, g_3) исправны, а также имеются цифровой пропорциональный регулятор (h) и бортовой алгоритм F вычисления угла рысканья по известным двум показаниям f_1 и f_2 и трем угловым скоростям. Соответствующая задача специфицируется формулой $G_1 - G_2$, где

$$\begin{aligned} (G_2)^{по} &= \forall \varphi, \phi, \theta, \omega_x, \omega_y, \omega_z \exists u, \\ (\neg G_2)^{по} &= \exists \varphi, \phi, \theta, \omega_x, \omega_y, \omega_z \forall u \exists False, \\ (G_1 \& \neg G_2)^{по} &= \exists True (\forall \tilde{\varphi}, \tilde{\psi}, \tilde{\theta} \exists u, (h) \\ &\quad \forall \varphi \exists \tilde{\varphi}, \forall \psi \exists \tilde{\psi}, (f_1, f_2) \\ &\quad \forall \omega_x \exists \tilde{\omega}_x, \forall \omega_y \exists \tilde{\omega}_y, \forall \omega_z \exists \tilde{\omega}_z, (g_1, g_2, g_3) \\ &\quad \forall \tilde{\varphi}, \tilde{\psi}, \tilde{\omega}_x, \tilde{\omega}_y, \tilde{\omega}_z \exists \tilde{\theta}, (F) \\ &\quad \forall True \exists \varphi, \phi, \theta, \omega_x, \omega_y, \omega_z \forall u \exists False). ((\neg G_2)^{по} \end{aligned}$$

Так как G_2 принадлежит классу формул, указанному в теореме 2, любой J -вывод формулы $(G_1 \& \neg G_2)^{по}$ будет конструктивен. Один из них имеет 9 шагов и сопровождается следующими изменениями в базе:

$$\begin{aligned} A_0 &= True, \quad A_1 = \{\varphi, \psi, \theta, \omega_x, \omega_y, \omega_z\}, \\ A_2 &= A_1 \cup \{\tilde{\varphi}\}, \quad A_3 = A_2 \cup \{\tilde{\psi}\}, \quad A_4 = A_3 \cup \{\tilde{\omega}_x\}, \\ A_5 &= A_4 \cup \{\tilde{\omega}_y\}, \quad A_6 = A_5 \cup \{\tilde{\omega}_z\}, \quad A_7 = A_6 \cup \{\tilde{\theta}\}, \\ A_8 &= A_7 \cup \{u\}, \quad A_9 = False. \end{aligned}$$

Желаемая структура алгоритмически извлекается из этого J -вывода в форме композиции $h(f_1, f_2, F(f_1, f_2, g_1, g_2, g_3))$ (вместо прежней структуры $h(f_1, f_2, f_3)$ нормального

режима работы) и устраняет незнание угла рысканья θ .

В данном примере не понадобилось использовать немонотонную логику и/или вводить в рассмотрение фактор времени, что рассматривается ниже, в п.п. 8 и 9.

4. Немонотонный вариант исчисления J

Рассмотрим возможную организацию *временных* выводов, учитывающих устаревание фактов во времени. Это относится к классу задач с *абсолютно предсказуемым будущим*, когда для решения задачи достаточно учесть только те изменения в мире, которые вызываются действиями управляемого объекта, в данном случае мобильного робота.

В [12] дан анализ результатов, где для планирования действий робота был использован метод резолюций совместно с механизмом забывания фактов. Но комбинация метода резолюций и процедуры забывания в общем случае оказалась внелогической, в отличие от чисто логической техники по-формализма.

В основе лежит описание действий, как по-формулам вида $\forall \bar{x} : A(\bar{x}), B^*(\bar{x}) \exists C(\bar{x})$, где символ $*$ принадлежит не объектному, а метаязыку и служит для ограничения применения α из-за того, что факты из B^* устаревают. А именно, если мы, отвечая на вопрос, имеем $A(\bar{x}\gamma), B^*(\bar{x}\gamma) \subseteq D$, где γ – подстановка, а база D содержит текущее описание мира, то факты $B^*(\bar{x}\gamma)$, содержащиеся в D , помечаются знаком $*$ и не могут быть использованы для дальнейшего вывода. Таким образом, ограничив применение правила α указанным образом, мы можем теперь рассуждать немонотонно, оставаясь при этом снова в рамках классической логики.

Сравним особенности такого стиля формализации с подходом, когда явно вводится дискретное время своими тактами t_0, t_1, t_2, \dots для отслеживания последовательности действий. Используется формула, “порождающая время”:

$$\forall t : T(t) \exists t' : T(t'), N(t, t'), \quad (4.1)$$

т.е. “если t есть момент времени, то существует момент t' , непосредственно следующий за t ”. В логике резолюционного типа использование в выводе такого «небольшого» фрагмента знания о пространственно-временной среде может лишить надежды на успешность выводов, в отличие от выводов в исчислении J над хорошо структурированными по-формулами, легко совместимых с эвристиками.

Каждое действие описывается формулой вида

$$\forall \bar{x}, t, t': A(\bar{x}, t), N(t, t') \quad \exists: A'(\bar{x}, t'),$$

где $A(\bar{x}, t)$ и $A'(\bar{x}, t')$ – полные описания соответственно исходного (предусловие) и конечного состояния мира. Вывод “порождает” промежуток времени t_i, t_{i+1} и планирует на этот промежуток одно из действий. Если некоторое предусловие является общим для нескольких разных действий, то могут быть спланированы несколько одновременных действий. Если они несовместимы, то вывод необходимо дополнить стратегией, запрещающей вывод двух действий в один промежуток времени, т.е. чтобы в каждый нечетный шаг вывода срабатывала формула (4.1).

Из двух упомянутых стилей представления знаний способ явного введения времени является более универсальным и его громоздкость есть плата за эту универсальность.

5. Использование развертывания следствий вместо доказательства теоремы

Потенциал применения логического подхода шире, чем только в форме АДТ в подсистеме диагностики и/или в контуре управления. Так, задача диагностики представима как задача поиска минимального непротиворечивого множества формул [40]. В данном пункте рассмотрим формирование управления в по-формализме, но не в форме автоматического доказательства априори заданной теоремы, а в форме вывода следствий (что отчасти согласуется с давними постановками из [41, 42]). Такое формирование управления основывается на рекурсивном развертывании функционирования системы во времени и пространстве, например, в задачах управления воздушным боем или управления

группой пассажирских лифтов. В этих приложениях, типа описанных в [2], неопределенность и многокритериальность являются существенными элементами постановки задачи. Здесь мы рассмотрим только задачу о лифтах [38].

Используется упреждающее моделирование динамики поведения лифтов во времени при разных управлениях с целью вывода характеристик (свойств) траекторий в пространстве состояний динамической системы, позволяющих судить о качестве соответствующих альтернативных вариантов управлений (плохие траектории, а точнее отвечающие им управления, далее в выводе не рассматриваются). При этом задействуется формула (4.1) и, как следствие такого развертывания, возникает рекурсивность. Мы называем некоторое множество вопросов *рекурсивным*, если возможен бесконечно длинный вывод при использовании только этих вопросов.

Основная по-формула имеет вид $\exists x: A(\Phi, \Psi, \Sigma)$. Здесь A – описание текущего состояния; Φ – формула (5.1), порождающая при применении α новый момент времени. Ψ описывает функционирование лифтов и используется для построения дерева возможных вариантов динамики в качественных терминах. Некоторый вариант может оказаться желаемым поведением, т.е. решением задачи выбора управления. В Σ формализованы некоторые эвристические принципы бракования управлений с целью сокращения и выбора окончательного. Множество (Ψ, Σ) не является рекурсивным.

Вывод состоит из циклов, каждый из которых начинается с порождения нового момента времени и выявления состояний лифтов при разных значениях управлений. Затем осуществляется оценка, в результате которой множество рассматриваемых вариантов развертывания динамики сокращается. Если остается более одного варианта, реализуется следующий цикл упреждающего моделирования и так далее до окончательного выбора единственного управления. Ограничение применения правила α осуществляется следующим образом: Φ используется только тогда, когда Ψ и Σ исчерпали свои ответы.

6. Гипотезирование в задачах с неполной информацией

Рассмотрим кратко результат, состоящий в развитии разработанных дедуктивных средств в направлении абдуктивного типа: проблематики генерализации (обобщения) первопорядковых формул, решения логических уравнений, индуктивного логического программирования и т.п.

Универсальные (полные) методы обнаружения выводимости (например, метод резолюций [19]) при использовании в задачах с ограниченными ресурсами недостаточно эффективны и нуждаются в тех или иных дополнительных средствах независимо от разрешимости теории. Кроме того, даже при отсутствии ограничений на ресурсы в неразрешимых (полуразрешимых) теориях возникает проблема принятия решения в случае, если доказываемая формула невыводима, а признаков этого не обнаруживается ни сразу, ни в процессе доказательства. При этом учет ресурсных ограничений оказывается полезным сам по себе для построения решающих правил, в частности, прерывающих бесконечный процесс поиска вывода.

Предлагается подход, который при возникновении трудностей с доказательством A преодолевает их путем конструктивного формирования и принятия некоторых дополнительных предположений X . Этот подход представляется характерным для содержательных рассуждений.

Под трудностями доказательства понимаются не только собственно исчерпание ресурсов, но и, например, появление тех или иных признаков бесперспективности или невозможности дальнейшего доказательства (зацикливание, неприменимость правил вывода и т. п.). Поясним это на простейшем примере.

По своей постановке задача разработки алгоритмического метода синтеза нетривиальных гипотез X для выводимости формул $X \vdash A$ напоминает работы по автоматическому синтезу теорем [21, 41-44], решению первопорядковых логических уравнений [44-46], обобщению формул исчисления предикатов [47, 48], индуктивному логическому программированию [49, 50], автоматическому построению теорий [51]. Однако предлагаемый метод решения по своему содержанию и сфере возможных приложе-

ний достаточно сильно отличается от указанных работ.

Гипотезы X синтезируются в сигнатуре A (или в ее подмножестве) по мере появления в процессе доказательства трудностей с его продолжением (завершением).

В случае, когда формула A представима как отрицание хорновской или как конъюнкция таких отрицаний, выделен случай, когда синтезируемое условие X является не только достаточным, но и необходимым [20]. Это повышает нетривиальность X , так как оно становится логически неулучшаемым (т.е. его логическое ослабление невозможно). Однако естественно, что свойство нетривиальности не является до конца формализуемым и, помимо желательности ослабления X , предполагает, например, обеспечение как можно более простой проверяемости X на выполнимость по сравнению с A на моделях рассматриваемой теории. Учет критерия проверяемости предлагается, например, в [44, 45]. Получение X в форме необходимых и достаточных условий позволяет дополнительными преобразованиями из [44, 45] получать, вообще говоря, более слабые окончательные условия.

Для невыполнимых (противоречивых) формул A естественно получаются также невыполнимые условия X . При этом интересна и другая постановка задачи обеспечения выводимости [52], когда допускается некоторая “минимальная” корректировка самой формулы A . Эта постановка предложена в [52] только для исчисления высказываний и нам не известно какое-либо ее развитие.

В качестве применения этого метода гипотезирования в “офлайн”-задачах назовем синтез теорем типа теорем сравнения и, в частности, теорем о свойстве достижимости процессов абстрактной динамической системы и свойстве типа управляемости процессов, описываемых конечно-автоматными уравнениями при возмущениях (например, в анализе динамики поведения пороговых сетей).

Из задач управления в реальном времени (“онлайн”-задач) назовем задачу автоматического порождения условий разрешимости логического распознавания типа летательного аппарата в условиях неполноты информации

(например, в контексте дальнего воздушного боя [2]). При этом формируемые условия имеют смысл рекомендаций ``нашему" пилоту, какие действия (маневры и др.) достаточно предпринять для того, чтобы получить бортовыми средствами новую информацию (например, с нового ракурса наблюдения) и успешно завершить распознавание.

Заключение

Для информационно-управляющих систем рассмотрены возможности автоматизации процессов управления и расширения потенциала управления путем использования некоторых известных и новых методов представления и обработки знаний.

Описаны логические средства (позитивные исчисления), их преимущества и ограничения в сравнении с известными, в том числе и нелогическими, средствами ИИ.

Рассмотрены вопросы применения этих логических средств как для моделирования, так и для формирования и улучшения управления динамическими системами. Рассматриваемые применения относятся к интеллектуальному управлению движущимися объектами. Делается вывод о предпочтительности использования разработанных логических средств в сравнении с известными.

Эти и другие современные достижения в области интеллектуального управления относятся к проблематике автоматизации поиска способов достижения поставленных извне целей. Достижения в автоматизации целеполагания и перемотра критериев качества управления совсем незначительны. Это по-прежнему остается суперзадачей в области создания систем следующего поколения – систем *интеллектуального* управления. Впрочем, и в проблематике интеллектуального управления (без автоматического целеполагания) остается ряд интересных проблем. Отметим их.

Несмотря на сильные стороны разработанных позитивных исчислений, представляется актуальным: разработать их "нечеткий" вариант, нарастить дедуктивные средства новыми возможностями пополнения знаний, в том числе абдуктивного типа (см. некоторый вариант этого в [20]), и развить некоторые современные

идеи приближения дедукции к практическим рассуждениям [15]. Кроме того, актуальной проблемой является организация взаимодействия уровня рассуждений общего вида со средним уровнем - обработки инструктивных знаний (типа правил "если – то") и с нижним уровнем (искусственные нейронные сети, генетические алгоритмы, традиционные алгоритмы управления). Примеры такой «ad hoc»-организации упомянуты (см. п. 1). Однако остаются, например, неразработанными принципы и методы автоматического преобразования знаний верхнего уровня в более быстро обрабатываемые знания нижних уровней (уровни инструктивных знаний и алгоритмов). Важной задачей остается автоматическая оценка иррелевантности знаний, поскольку не только дефицит, но и переизбыток информации, ведет к деградации СИУ. Интересны результаты об оценке иррелевантности знаний в задачах дедукции над большими базами знаний хорновского типа, полученные в [53].

Литература

1. Åström K.J., McAvoy T.J. Intelligent Control: An Overview and Evaluation. - In: Handbook of Intelligent Control. Neural, Fuzzy, and Adaptive Approaches / D.A. White, D.A. Sofge (eds.). - Van Nostrand Reinhold, New York, 1992, pp. 3-34.
2. Васильев С.Н., Жерлов А.К., Федосов Е.А., Федунцов Б.Е. Интеллектуальное управление динамическими системами. М., ФИЗМАТЛИТ, МАИК "Наука / Интерпериодика", 2000.
3. Заде Л. Понятие лингвистической переменной и его применение к понятию приближенных решений. - В серии: Математика. Новое в зарубежной науке / Под ред. А.Н. Колмогорова, С.П. Новикова. М., Мир, 1976.
4. Kifer M., Wu J. A Logic for Object-Oriented Logic Programming. - Proc. 8th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, 1989, pp. 379-393.
5. Kifer M., Lausen G. A Higher Order Language for Reasoning about Objects, Inheritance and Scheme. - Proc. of 1989 ACM SIGMOD Intern. Conference on Management of Data, 1990, v. 18, N 2, pp. 134-146.
6. Kifer M., Lausen G., Wu J. Logical Foundations of Object-Oriented and Frame-Based Languages. Tech. Report 90/14, SUNY at Stony Brook, 1990.
7. Handelman D.A., Stengel R.F. An Architecture for Real-Time Rule-Based. - Proc. American Control Conference, 1987, pp. 1636-1642.
8. Wos L. Solving Open Questions with an Automated Theorem Proving Program. - Lecture Notes in Computer Science, 1982, v. 138, pp. 1-31; русский перевод: Л. Воз.

- Решение некоторых открытых проблем с помощью программы для автоматического доказательства теорем. - Киберн. сб., Нов.сер., вып. 21, М., Мир, 1984, с. 235-263.
9. Wos L., Veroff R. Logical Basis for the Automation of Reasoning: Case Studies. - In: Handbook of Logic in Artificial Intelligence and Logic Programming / D.M. Gabbay, C.J. Hogger, J.A. Robinson (eds.), Clarendon Press, Oxford, 1994, v. 2, pp. 1-40.
 10. Kohn W. Declarative Control Architecture. - Communications of the ACM, 1991, 34, N 8, pp. 65-79.
 11. Fikes R., Nilsson N. STRIPS: A New Approach to the Application of Theorem Proving in Problem Solving. - Artificial Intelligence, 1971, 2, pp. 189-208; русский перевод в сб. "Интегральные роботы", вып.1, М., Мир, 1973.
 12. Fikes R.E., Nilsson N.J. STRIPS, a Retrospective. - Artificial Intelligence, 1993, 59, NN 1-2, pp. 227-232.
 13. Bibel W. A Deductive Solution for Plan Generation. - New Generation Computing, 1986, 4, N 2, pp. 115-132.
 14. Фельдбаум А.А. Новые принципы автоматического управления. - Изв. ВУЗов СССР, Радиотехника, 1960, N 3, с. 299-308, N 4, с. 419-430.
 15. Gabbay D.M. Preface to Handbook of Logic in Artificial Intelligence and Logic Programming / Gabbay D.M., Hogger C.J., Robinson J.A., Nute D. (eds). Clarendon Press, Oxford, 1994, v. 3, pp. v-xxii.
 16. Давыдов А.В. Об исчислении позитивно-образованных формул с функциональными символами // Материалы 2-ой Всероссийской конференции "Информационные и вычислительные технологии и системы", т.2, с. 84-88. 2006.
 17. Robinson J.A. The Generalised Resolution Principle. - In: Machine Intelligence / D.Michie (ed.), 1968, 3, NY, American Elsevier, pp. 77-94.
 18. Robinson J.A. On Automatic Deduction // Rice University Studies, 50, 1964, p. 69-89.
 19. Robinson J.A. A Machine-oriented Logic Based on the Resolution Principle // J. ACM, 12, 1965, p. 23-41.
 20. Васильев С.Н. Метод синтеза условий выводимости хорновских и некоторых других формул. - Сибирский математический журнал, 1997, т. 38, N 5, с. 1034-1046.
 21. Глушков В.М. Машина доказывает. - М., Знание, 1981.
 22. Капитонова Ю.В., Летичевский А.А. О некоторых идеях формирования математического аппарата кибернетики в работах В.М. Глушкова // Кибернетика, 6, 1982, с. 1-5, 9.
 23. Глушков В.М., Капитонова Ю.В., Летичевский А.А., Вершинин К.П., Малеваный Н.П. К построению практического формального языка для записи математических теорий // Кибернетика, 5, 1972, с. 19-28.
 24. Глушков В.М., Вершинин К.П., Капитонова Ю.В. О формальном языке для записи математических текстов // Автоматизация поиска доказательств теорем в математике. Киев: ИК АН УССР, 1974, с. 3-26.
 25. Глушков В.М., Капитонова Ю.В. Автоматизация поиска доказательств теорем математических теорий и интеллектуальные машины // Кибернетика, 5, 1972, с. 2-6.
 26. Ануфриев Ф.В., Асельдеров З.М. Алгоритм очевидности // Кибернетика, 5, 1972, с. 29-60.
 27. Шанин Н.А. и др. Алгоритм машинного поиска естественного логического вывода в исчислении высказываний. М.-Л., Наука, 1965.
 28. Асельдеров З.М., Ануфриев Ф.В., Капитонова Ю.В. Организация данных для алгоритма очевидности в поиске доказательств теорем в формализованных теориях // Кибернетика, 5, 1972, с. 61-67.
 29. Маслов С.Ю. Применение обратного метода к теории разрешимых фрагментов классического исчисления // ДАН СССР, 171, №6, 1964, с. 17-20.
 30. Маслов С.Ю. Обратный метод установления выводимости в классическом исчислении // ДАН СССР, 159, №1, 1966, с. 1282-1285.
 31. Маслов С.Ю. Обратный метод установления выводимости для логических исчислений // Труды Математического института им. В.А. Стеклова, 98, М.-Л., Наука, 1968, с. 26-87.
 32. Маслов С.Ю. Связь между тактиками обратного метода и метода резолюций // Записки научных семинаров ЛОМИ АН СССР, 16, 1969, с. 137-146.
 33. Maslov S.Yu. Proof-search Strategies for Methods of the Resolution // Machine Intelligence, 6, N.Y., American Elsevier, 1971, p. 77-90.
 34. Маслов С.Ю., Минц Г.Е. Теория поиска вывода и обратный метод // Поиск вывода и автоматизация доказательств. Дополнение А к книге: Ч. Чень, Р. Ли. Математическая логика и автоматическое доказательство теорем, перевод с английского Г.В. Давыдова, Г.Е. Минца, А.В. Соколиной, под ред. С.Ю. Маслова, М., Наука, 1983, с. 291-314.
 35. Давыдов Г.В. Синтез метода резолюций с обратным методом // Записки научных семинаров ЛОМИ АН СССР, 20, М.-Л., 1971.
 36. Такеути Г. Теория доказательств. М., Мир, 1978.
 37. Васильев С.Н., Черкашин Е.А. Интеллектуальное управление телескопом. - Сибирский журнал индустриальной математики, 1998, т. 1, N 2, с. 81-98.
 38. Васильев С.Н., Жерлов А.К. Логическое моделирование и управление в реальном времени. - Труды Всесоюзной конференции "Интеллектуальные системы в машиностроении" / Под ред. В.А. Виттиха, Самара, 1991, вып. 2, с. 33-38.
 39. Gabbay D.M., Reynolds M. Towards a Computational Treatment of Time. - Handbook of Logic in Artificial Intelligence and Logic Programming. Epistemic and Temporal Reasoning / D.M.Gabbay, C.J.Hogger, J.A.Robinson (eds.), Clarendon Press, Oxford, 1995, v. 4, pp. 351-437.
 40. Reiter R. A Theory of Diagnosis from First Principles. - Artificial Intelligence, 1987, 32, N 1, pp. 57-95.
 41. Wang H. Towards Mechanical Mathematics. - IBM J. Res. Develop., 1960, v. 4, N 1, pp. 2-22.
 42. Lee R. A Completeness Theorem and a Computer Program for Finding Theorems Derivable from Given Axioms. - Doctoral Diss., Dept. of Electrical Engineering and Computer Science, Univ. of California, Berkeley, 1967.
 43. Матросов В.М., Васильев С.Н., Анапольский Л.Ю. Метод сравнения в математической теории систем. - омосибирск, Наука, 1981.

44. Матросов В.М., Васильев С.Н., Каратуев В.Г., Козлов Р.И., Суменков Е.А., Ядыкин С.А. Алгоритмы вывода теорем метода векторных функций Ляпунова. - Новосибирск, Наука, 1981.
45. Vassilyev S.N. Machine Synthesis of Mathematical Theorems. - J. of Logic Programming, 1990, v. 9, N 2 & 3, pp. 35-266.
46. Таутс А.И. Решение логических уравнений итерационным методом в исчислении предикатов. - Известия Института физической астрономии Академии наук ЭССР, 1964, т. 24, с. 17-24.
47. Muggleton S. Inductive Acquisition of Expert Knowledge. - Turing Institute Press, Addison-Wesley Publ. Company, Wokingham, 1990.
48. Moore J.S., Boyer R.S. A Computational Logic Handbook. - Perspectives in Computing. Academic Press, 1988, vol. 23.
49. Muggleton S., Buntine W. Machine Invention of First-Order Predicates by Inverting Resolution. - Proc. 5 Intern. Conference on Machine Learning / J.Laird (ed.), 1988, pp. 339-352.
50. Flach P.A. Towards the Inductive Logic Programming. - Proc. BENELEARN-91, Depart. of Social Science Informatics, Univ. of Amsterdam, 1991, pp. 88-96.
51. Lenat D.B. On Automated Scientific Theory Formation: A Case Study Using the AM Program. - Machine Intelligence, 1981, N 9, pp. 251-283.
52. Давыдов Г.В. О корректировании недоказуемых формул. - Зап. Научн. Сем. ЛОМИ АН СССР, 1967, т. 4, с. 18-29.
53. Levy A.Y., Fikes R.E., Sagiv Y. Speeding up Inferences Using Relevance Reasoning: A Formalism and Algorithms. - Artificial Intelligence, 1997, 97, pp. 83-136.

Васильев Станислав Николаевич. Директор Института проблем управления РАН им. В.А. Трапезникова. Окончил Казанский авиационный институт (ныне КГТУ) им. А.Н. Туполева в 1970 году. Доктор физико-математических наук, профессор, чл.-к. РАН (с 1997 г.), академик (с 2006 г.). Имеет 315 публикаций, в том числе 8 монографий. Награжден орденом Дружбы (2006г.), орденом Почета (1999 г.). Лауреат премии СО РАН в области фундаментальных исследований (1986 г.). Лауреат государственной премии СССР в области науки и техники (1984 г.). Область научных интересов - процессы управления, логико-динамические системы, интеллектуальное управление, динамика и управление гетерогенными системами, логические модели динамических систем.